# TMS320DM8127 and TMS320DM814x DaVinci™ Digital Media Processor

# Technical Reference Manual

Literature Number: SPRUGZ8G
14 November 2011–Revised June 2016

# Contents


**Preface** ..... **153**

**1 Chip Level Resources** ..... **155**

1.1 ARM® Cortex™-A8 Microprocessor Unit (MPU) Subsystem Overview ..... 156
1.1.1 ARM Cortex-A8 RISC Processor ..... 156
1.1.2 Embedded Trace Module (ETM) ..... 157
1.1.3 ARM Cortex-A8 Interrupt Controller (AINTC) ..... 157
1.1.4 ARM Cortex-A8 PLL (PLL_ARM) ..... 157
1.1.5 ARM MPU Interconnect ..... 157
1.2 Media Controller Subsystem ..... 158
1.2.1 Dual Cortex-M3 MPU Subsystem Overview ..... 158
1.2.2 Dual Cortex-M3 MPU Subsystem Integration ..... 159
1.2.3 Dual Cortex-M3 MPU Subsystem Functional Description ..... 162
1.2.4 Shared Cache ..... 164
1.2.5 Shared Cache MMU ..... 164
1.2.6 L2 MMU ..... 165
1.2.7 Registers ..... 166
1.3 C674x DSP ..... 180
1.3.1 Introduction ..... 180
1.3.2 TMS320C674x Megamodule ..... 181
1.3.3 Memory Map ..... 183
1.3.4 Advanced Event Triggering (AET) ..... 183
1.4 HD Video Coprocessor Subsystem ..... 183
1.4.1 HDVICP2 Overview ..... 183
1.4.2 ICONTs ..... 185
1.4.3 vDMA ..... 185
1.4.4 iME3 ..... 185
1.4.5 iPE3 ..... 185
1.4.6 MC3 ..... 185
1.4.7 CALC3 ..... 186
1.4.8 iLF3 ..... 186
1.4.9 ECD3 ..... 186
1.4.10 SL2 Interface ..... 186
1.4.11 Message Bus ..... 186
1.4.12 HDVICP2 Local Interconnect ..... 186
1.4.13 MailBox ..... 186
1.4.14 HDVICP2 System Control ..... 186
1.5 System MMU ..... 188
1.5.1 MMU Overview ..... 188
1.5.2 MMU Integration ..... 188
1.5.3 MMU Functional Description ..... 189
1.5.4 MMU Low-level Programming Models ..... 202
1.5.5 MMU Registers ..... 205
1.6 SGX530 Graphics Subsystem ..... 218
1.6.1 SGX Overview ..... 218
1.6.2 SGX Integration ..... 221





Copyright © 2011–2016, Texas Instruments Incorporated

1.6.3 SGX Functional Description ..... 223
1.6.4 SGX Register Manual ..... 225
1.7 Device Interrupts and EDMA Events ..... 240
1.7.1 Interrupt Controllers ..... 240
1.7.2 Interrupt Requests to Cortex-A8 MPU INTC ..... 241
1.7.3 Interrupt Requests to C674x INTC ..... 244
1.7.4 Interrupt Requests to Media Controller INTC ..... 248
1.7.5 EDMA Channel Synchronization Events ..... 252
1.8 Inter-Processor Communication ..... 255
1.8.1 Reset Requirements ..... 255
1.8.2 Features ..... 255
1.8.3 Overview and Strategy ..... 256
1.8.4 IPC Component Configuration ..... 259
1.9 Mailbox ..... 260
1.9.1 Overview ..... 260
1.9.2 System Mailbox Integration ..... 260
1.9.3 Functional Description ..... 262
1.9.4 Programming Guide ..... 266
1.9.5 Mailbox Registers ..... 270
1.10 Spinlock ..... 290
1.10.1 Overview ..... 290
1.10.2 Integration ..... 291
1.10.3 Functional Description ..... 292
1.10.4 Programming Guide ..... 294
1.10.5 Spinlock Registers ..... 296
1.11 Error Location Module ..... 300
1.11.1 Error Location Module Overview ..... 300
1.11.2 ELM Integration ..... 301
1.11.3 ELM Functional Description ..... 302
1.11.4 ELM Basic Programming Model ..... 305
1.11.5 ELM Registers ..... 312
1.12 Bus Interconnect ..... 324
1.12.1 Interconnect Overview ..... 324
1.12.2 L3 Interconnect ..... 328
1.12.3 L4 Interconnects ..... 408
**2 Power, Reset, and Clock Management (PRCM) Module ..... 462**
2.1 Introduction ..... 463
2.1.1 Device Power-Management Architecture Building Blocks ..... 463
2.1.2 Module-Level Clock Management ..... 464
2.1.3 Clock Domain ..... 467
2.1.4 Power Management ..... 469
2.2 Power Reset Clock Management Structure ..... 472
2.2.1 Introduction ..... 472
2.3 Clock Generation and Management ..... 474
2.3.1 Types of Clocks ..... 474
2.3.2 Clock Structure ..... 474
2.3.3 Main PLL Clock Structure ..... 477
2.3.4 USB PLL Clock Structure ..... 479
2.3.5 DDR PLL Clock Structure ..... 480
2.3.6 SERDES and Ethernet Clock Structure ..... 480
2.3.7 Video PLL Clock Structure ..... 481
2.3.8 Audio PLL Clock Structure ..... 483
2.3.9 Timer Clock Structure ..... 486


Copyright © 2011–2016, Texas Instruments Incorporated

2.3.10 Device Clocking from External Sources ..... 488
2.3.11 Clock Output Pin ..... 489
2.4 Oscillator ..... 490
2.5 DPLLS ..... 491
2.5.1 DPLLS Frequency Range ..... 492
2.5.2 DPLLS Fractional M-divider Programming ..... 492
2.5.3 DPLLS Clock Functions ..... 492
2.5.4 DPLLS Frequency Factors ..... 493
2.6 DPLLLJ ..... 493
2.6.1 DPLLLJ Fractional M-divider Programming ..... 495
2.6.2 DPLLLJ Clock Functions ..... 495
2.6.3 DPLLLJ Frequency Range (MHz) ..... 496
2.6.4 DPLLLJ Frequency Factors ..... 496
2.6.5 Mode of Operations ..... 496
2.6.6 BYPASS Mode ..... 497
2.6.7 M2 & N2 Change On-the-Fly ..... 498
2.6.8 Gating and Power Down of Output Clocks ..... 498
2.6.9 M2PWDNZ / CLKDCOLDOPWDNZ bits ..... 498
2.6.10 Connected Outputs of DPLLS ..... 499
2.6.11 Connected Outputs of DPLLLJ ..... 499
2.7 Reset Management ..... 499
2.7.1 Overview ..... 499
2.7.2 Reset Concepts and Definitions ..... 499
2.7.3 Power On Reset (PORz) ..... 501
2.7.4 External Warm Reset (RESETz) ..... 501
2.7.5 Emulation Cold Reset (ICEPICK POR) ..... 501
2.7.6 Emulation Warm Reset ..... 501
2.7.7 Reset Requestor ..... 501
2.7.8 Software Controlled Reset ..... 501
2.7.9 Test Reset (TRSTz) ..... 501
2.7.10 Reset Priority ..... 502
2.7.11 Reset Characteristics ..... 502
2.7.12 Trace Functionality Across Reset ..... 502
2.7.13 Reset Isolation ..... 503
2.7.14 Reset Output ($\overline{\text{RSTOUT_WD_OUT}}$) ..... 503
2.7.15 Power-on-Reset (PORz) Sequence ..... 504
2.7.16 RESETz Sequence ..... 504
2.7.17 Power-Up/Down Sequence ..... 505
2.7.18 IO State ..... 505
2.7.19 MPU Subsystem POR Sequence ..... 505
2.7.20 MPU Subsystem Warm Sequence ..... 506
2.7.21 HDVICP Power-On Reset Sequence ..... 507
2.7.22 HDVICP Software Warm Reset Sequence ..... 509
2.7.23 C674x DSP Power-On Reset Sequence ..... 509
2.7.24 C674x DSP Warm Reset Sequence ..... 510
2.7.25 Media Controller Power-On Reset Sequence ..... 510
2.7.26 Media Controller Warm Reset Sequence ..... 512
2.8 Voltage and Power Domains ..... 512
2.8.1 Voltage Domains ..... 513
2.8.2 Power Domains ..... 514
2.8.3 Device Modules and Power Management Attribute List ..... 515
2.8.4 Power Domain Power-Down Sequence ..... 515
2.8.5 Power Domain Power-Up Sequence ..... 515


Copyright © 2011–2016, Texas Instruments Incorporated

2.9 Voltage Management Functional Description ..... 517
2.9.1 Overview ..... 517
2.9.2 Internal LDOs Control ..... 518
2.10 PRCM Registers ..... 518
2.10.1 PLLSS Registers ..... 520
2.10.2 PRM_DEVICE Registers ..... 707
2.10.3 CM_DEVICE Registers ..... 710
2.10.4 OCP_SOCKET_PRM Registers ..... 712
2.10.5 CM_DPLL Registers ..... 713
2.10.6 CM_ACTIVE Registers ..... 733
2.10.7 CM_DEFAULT Registers ..... 735
2.10.8 CM_HDVICP Registers ..... 743
2.10.9 CM_ISP Registers ..... 747
2.10.10 CM_DSS Registers ..... 751
2.10.11 CM_SGX Registers ..... 754
2.10.12 PRM_ACTIVE Registers ..... 756
2.10.13 PRM_DEFAULT Registers ..... 761
2.10.14 PRM_HDVICP Registers ..... 766
2.10.15 PRM_ISP Registers ..... 771
2.10.16 PRM_DSS Registers ..... 775
2.10.17 PRM_SGX Registers ..... 780
2.10.18 CM_ALWON Registers ..... 785
2.10.19 PRM_ALWON Registers ..... 848

**3 Control Module ..... 851**

3.1 Control Module ..... 852
3.1.1 MMR_LOCK Register (Locked/Unlocked States) ..... 852
3.1.2 Pin Multiplexing (Pin Mux) ..... 853
3.1.3 BOOT Status ..... 854
3.1.4 DSP / EDMA Memory Management Unit Configuration ..... 854
3.1.5 Interrupt and DMA Controller Crossbars ..... 854
3.1.6 Device Level Peripheral Configuration ..... 855
3.1.7 Reset Isolation ..... 855
3.2 CONTROL_MODULE Registers ..... 856
3.2.1 CONTROL_REVISION Register (offset = 0h) [reset = 40000000h] ..... 862
3.2.2 CONTROL_HWINFO Register (offset = 4h) [reset = 0h] ..... 863
3.2.3 CONTROL_SYSCONFIG Register (offset = 10h) [reset = 0h] ..... 864
3.2.4 CONTROL_STATUS Register (offset = 40h) [reset = 0h] ..... 865
3.2.5 BOOTSTAT Register (offset = 44h) [reset = 0h] ..... 867
3.2.6 DSPBOOTADDR Register (offset = 48h) [reset = 800000h] ..... 868
3.2.7 MMR_LOCK0 Register (offset = 60h) [reset = 1A1C8144h] ..... 869
3.2.8 MMR_LOCK1 Register (offset = 64h) [reset = FDF45530h] ..... 870
3.2.9 MMR_LOCK2 Register (offset = 68h) [reset = 1AE6E320h] ..... 871
3.2.10 MMR_LOCK3 Register (offset = 6Ch) [reset = 2FFA927Ch] ..... 872
3.2.11 MMR_LOCK4 Register (offset = 70h) [reset = 143F832Ch] ..... 873
3.2.12 CONTROL_SEC_CTRL Register (offset = 100h) [reset = 1Bh] ..... 874
3.2.13 DEVOSC Register (offset = 468h) [reset = 0h] ..... 875
3.2.14 AUXOSC Register (offset = 46Ch) [reset = 2h] ..... 876
3.2.15 PCIE_CFG Register (offset = 480h) [reset = 0h] ..... 877
3.2.16 DEVICE_ID Register (offset = 600h) [reset = B8F202Eh] ..... 878
3.2.17 DEV_FEATURE Register (offset = 604h) [reset = 0h] ..... 879
3.2.18 INIT_PRIORITY_0 Register (offset = 608h) [reset = 0h] ..... 881
3.2.19 INIT_PRIORITY_1 Register (offset = 60Ch) [reset = 0h] ..... 882
3.2.20 MMU_CFG Register (offset = 610h) [reset = 80h] ..... 883


Copyright © 2011–2016, Texas Instruments Incorporated

3.2.21 TPTC_CFG Register (offset = 614h) [reset = AAh] ..... 884
3.2.22 DSP_IDLE_CFG Register (offset = 61Ch) [reset = 28h] ..... 885
3.2.23 USB_CTRL0 Register (offset = 620h) [reset = 3C006007h] ..... 886
3.2.24 USB_STS0 Register (offset = 624h) [reset = 0h] ..... 888
3.2.25 USB_CTRL1 Register (offset = 628h) [reset = 3C006007h] ..... 889
3.2.26 USB_STS1 Register (offset = 62Ch) [reset = 0h] ..... 891
3.2.27 MAC_ID0_LO Register (offset = 630h) [reset = 0h] ..... 892
3.2.28 MAC_ID0_HI Register (offset = 634h) [reset = 0h] ..... 893
3.2.29 MAC_ID1_LO Register (offset = 638h) [reset = 0h] ..... 894
3.2.30 MAC_ID1_HI Register (offset = 63Ch) [reset = 0h] ..... 895
3.2.31 SW_REVISION Register (offset = 640h) [reset = 0h] ..... 896
3.2.32 DCAN_RAMINIT Register (offset = 644h) [reset = 0h] ..... 897
3.2.33 AUD_CTRL Register (offset = 64Ch) [reset = 0h] ..... 898
3.2.34 GMII_SEL Register (offset = 650h) [reset = 0h] ..... 899
3.2.35 OCMEM_PWRDN Register (offset = 654h) [reset = 0h] ..... 900
3.2.36 MEDIA_CONTROLLER_MEM_PWRDN Register (offset = 65Ch) [reset = 0h] ..... 901
3.2.37 SD_DAC_CTRL Register (offset = 670h) [reset = 18800C6h] ..... 902
3.2.38 SD_DAC0_CAL Register (offset = 674h) [reset = 0h] ..... 904
3.2.39 SD_DAC1_CAL Register (offset = 678h) [reset = 0h] ..... 905
3.2.40 SD_DAC0_REGCTRL Register (offset = 67Ch) [reset = 0h] ..... 906
3.2.41 SD_DAC0_REGSTATUS Register (offset = 680h) [reset = 0h] ..... 907
3.2.42 SD_DAC1_REGCTRL Register (offset = 684h) [reset = 0h] ..... 908
3.2.43 SD_DAC1_REGSTATUS Register (offset = 688h) [reset = 0h] ..... 909
3.2.44 EMIF_CLK_GATE Register (offset = 694h) [reset = 0h] ..... 910
3.2.45 SMRT_CTRL Register (offset = 6A0h) [reset = 0h] ..... 911
3.2.46 ARM_CORTEX_A8_HW_DBG_SEL Register (offset = 6A4h) [reset = 0h] ..... 912
3.2.47 ARM_CORTEX_A8_HW_DBG_INFO Register (offset = 6A8h) [reset = 0h] ..... 913
3.2.48 PRCM_DEBUG_ALWON_DEFAULT Register (offset = 6B0h) [reset = 3F7B0010h] ..... 914
3.2.49 PRCM_DEBUG_PD_DOMAIN_STATUS Register (offset = 6B4h) [reset = 871C7180h] ..... 916
3.2.50 PCIE_PLLCFG0 Register (offset = 6D8h) [reset = 40007000h] ..... 919
3.2.51 PCIE_PLLCFG1 Register (offset = 6DCh) [reset = 0h] ..... 921
3.2.52 PCIE_PLLCFG2 Register (offset = 6E0h) [reset = 0h] ..... 922
3.2.53 PCIE_PLLCFG3 Register (offset = 6E4h) [reset = 4000000h] ..... 923
3.2.54 PCIE_PLLCFG4 Register (offset = 6E8h) [reset = 0h] ..... 925
3.2.55 PCIE_PLLSTATUS Register (offset = 6ECh) [reset = 0h] ..... 927
3.2.56 PCIE_RXSTATUS Register (offset = 6F0h) [reset = 0h] ..... 928
3.2.57 PCIE_TXSTATUS Register (offset = 6F4h) [reset = 0h] ..... 929
3.2.58 SATA_PLLCFG0 Register (offset = 720h) [reset = C0007160h] ..... 930
3.2.59 SATA_PLLCFG1 Register (offset = 724h) [reset = 812C0000h] ..... 931
3.2.60 SATA_PLLCFG2 Register (offset = 728h) [reset = 0h] ..... 932
3.2.61 SATA_PLLCFG3 Register (offset = 72Ch) [reset = 4000000h] ..... 933
3.2.62 SATA_PLLCFG4 Register (offset = 730h) [reset = 0h] ..... 936
3.2.63 SATA_TXSTATUS Register (offset = 73Ch) [reset = 0h] ..... 938
3.2.64 SATA_TESTCFG Register (offset = 740h) [reset = 0h] ..... 939
3.2.65 VDD_MPU_OPP_050 Register (offset = 770h) [reset = 0h] ..... 940
3.2.66 VDD_MPU_OPP_100 Register (offset = 774h) [reset = 0h] ..... 941
3.2.67 VDD_MPU_OPP_120 Register (offset = 778h) [reset = 0h] ..... 942
3.2.68 VDD_MPU_OPP_166 Register (offset = 77Ch) [reset = 0h] ..... 943
3.2.69 VDD_C674x_DSP_OPP_050 Register (offset = 788h) [reset = 0h] ..... 944
3.2.70 VDD_C674x_DSP_OPP_100 Register (offset = 78Ch) [reset = 0h] ..... 945
3.2.71 VDD_C674x_DSP_OPP_120 Register (offset = 790h) [reset = 0h] ..... 946
3.2.72 VDD_C674x_DSP_OPP_166 Register (offset = 794h) [reset = 0h] ..... 947
3.2.73 VDD_HDVICP_OPP_050 Register (offset = 7A0h) [reset = 0h] ..... 948


Copyright © 2011–2016, Texas Instruments Incorporated

3.2.74 VDD_HDVICP_OPP_100 Register (offset = 7A4h) [reset = 0h] ... 949
3.2.75 VDD_HDVICP_OPP_120 Register (offset = 7A8h) [reset = 0h] ... 950
3.2.76 VDD_HDVICP_OPP_166 Register (offset = 7ACh) [reset = 0h] ... 951
3.2.77 VDD_CORE_OPP_050 Register (offset = 7B8h) [reset = 0h] ... 952
3.2.78 VDD_CORE_OPP_100 Register (offset = 7BCh) [reset = 0h] ... 953
3.2.79 VDD_CORE_OPP_120 Register (offset = 7C0h) [reset = 0h] ... 954
3.2.80 VDD_CORE_OPP_166 Register (offset = 7C4h) [reset = 0h] ... 955
3.2.81 BB_SCALE Register (offset = 7D0h) [reset = 0h] ... 956
3.2.82 USB_VID_PID Register (offset = 7F4h) [reset = 0h] ... 957
3.2.83 PCIE_VID_PID Register (offset = 7F8h) [reset = 0h] ... 958
3.2.84 PINCNTL[270:1] Register (offset = 800h) [reset = 40000h] ... 959
3.2.85 MLBP_SIG_IO_CTRL Register (offset = E18h) [reset = 38h] ... 960
3.2.86 MLBP_DAT_IO_CTRL Register (offset = E1Ch) [reset = 38h] ... 961
3.2.87 SERDES_REFCLK_CTL Register (offset = E24h) [reset = 3h] ... 962
3.2.88 DSP_INTMUX_15_18 Register (offset = F00h) [reset = 0h] ... 963
3.2.89 DSP_INTMUX_19_22 Register (offset = F04h) [reset = 0h] ... 964
3.2.90 DSP_INTMUX_23_26 Register (offset = F08h) [reset = 0h] ... 965
3.2.91 DSP_INTMUX_27_30 Register (offset = F0Ch) [reset = 0h] ... 966
3.2.92 DSP_INTMUX_31_34 Register (offset = F10h) [reset = 0h] ... 967
3.2.93 DSP_INTMUX_35_38 Register (offset = F14h) [reset = 0h] ... 968
3.2.94 DSP_INTMUX_39_42 Register (offset = F18h) [reset = 0h] ... 969
3.2.95 DSP_INTMUX_43_46 Register (offset = F1Ch) [reset = 0h] ... 970
3.2.96 DSP_INTMUX_47_50 Register (offset = F20h) [reset = 0h] ... 971
3.2.97 DSP_INTMUX_51_54 Register (offset = F24h) [reset = 0h] ... 972
3.2.98 DSP_INTMUX_55_58 Register (offset = F28h) [reset = 0h] ... 973
3.2.99 DSP_INTMUX_59_62 Register (offset = F2Ch) [reset = 0h] ... 974
3.2.100 DSP_INTMUX_63_66 Register (offset = F30h) [reset = 0h] ... 975
3.2.101 DSP_INTMUX_67_70 Register (offset = F34h) [reset = 0h] ... 976
3.2.102 DSP_INTMUX_71_74 Register (offset = F38h) [reset = 0h] ... 977
3.2.103 DSP_INTMUX_75_78 Register (offset = F3Ch) [reset = 0h] ... 978
3.2.104 DSP_INTMUX_79_82 Register (offset = F40h) [reset = 0h] ... 979
3.2.105 DSP_INTMUX_83_86 Register (offset = F44h) [reset = 0h] ... 980
3.2.106 DSP_INTMUX_87_90 Register (offset = F48h) [reset = 0h] ... 981
3.2.107 DSP_INTMUX_91_94 Register (offset = F4Ch) [reset = 0h] ... 982
3.2.108 DSP_INTMUX_95 Register (offset = F50h) [reset = 0h] ... 983
3.2.109 Media_Controller_INTMUX_0_3 Register (offset = F54h) [reset = 0h] ... 984
3.2.110 Media_Controller_INTMUX_4_7 Register (offset = F58h) [reset = 0h] ... 985
3.2.111 Media_Controller_INTMUX_8_11 Register (offset = F5Ch) [reset = 0h] ... 986
3.2.112 Media_Controller_INTMUX_12_15 Register (offset = F60h) [reset = 0h] ... 987
3.2.113 Media_Controller_INTMUX_16_19 Register (offset = F64h) [reset = 0h] ... 988
3.2.114 Media_Controller_INTMUX_20_23 Register (offset = F68h) [reset = 0h] ... 989
3.2.115 Media_Controller_INTMUX_24_27 Register (offset = F6Ch) [reset = 0h] ... 990
3.2.116 Media_Controller_INTMUX_28_31 Register (offset = F70h) [reset = 0h] ... 991
3.2.117 Media_Controller_INTMUX_32_35 Register (offset = F74h) [reset = 0h] ... 992
3.2.118 Media_Controller_INTMUX_36_39 Register (offset = F78h) [reset = 0h] ... 993
3.2.119 Media_Controller_INTMUX_40_43 Register (offset = F7Ch) [reset = 0h] ... 994
3.2.120 Media_Controller_INTMUX_44_47 Register (offset = F80h) [reset = 0h] ... 995
3.2.121 Media_Controller_INTMUX_48_51 Register (offset = F84h) [reset = 0h] ... 996
3.2.122 Media_Controller_INTMUX_52_55 Register (offset = F88h) [reset = 0h] ... 997
3.2.123 Media_Controller_INTMUX_56 Register (offset = F8Ch) [reset = 0h] ... 998
3.2.124 EDMA3CC_EVTMUX_0_3 Register (offset = F90h) [reset = 0h] ... 999
3.2.125 EDMA3CC_EVTMUX_4_7 Register (offset = F94h) [reset = 0h] ... 1000
3.2.126 EDMA3CC_EVTMUX_8_11 Register (offset = F98h) [reset = 0h] ... 1001


Copyright © 2011–2016, Texas Instruments Incorporated

3.2.127 EDMA3CC_EVTMUX_12_15 Register (offset = F9Ch) [reset = 0h] ..... 1002
3.2.128 EDMA3CC_EVTMUX_16_19 Register (offset = FA0h) [reset = 0h] ..... 1003
3.2.129 EDMA3CC_EVTMUX_20_23 Register (offset = FA4h) [reset = 0h] ..... 1004
3.2.130 EDMA3CC_EVTMUX_24_27 Register (offset = FA8h) [reset = 0h] ..... 1005
3.2.131 EDMA3CC_EVTMUX_28_31 Register (offset = FACh) [reset = 0h] ..... 1006
3.2.132 EDMA3CC_EVTMUX_32_35 Register (offset = FB0h) [reset = 0h] ..... 1007
3.2.133 EDMA3CC_EVTMUX_36_39 Register (offset = FB4h) [reset = 0h] ..... 1008
3.2.134 EDMA3CC_EVTMUX_40_43 Register (offset = FB8h) [reset = 0h] ..... 1009
3.2.135 EDMA3CC_EVTMUX_44_47 Register (offset = FBCh) [reset = 0h] ..... 1010
3.2.136 EDMA3CC_EVTMUX_48_51 Register (offset = FC0h) [reset = 0h] ..... 1011
3.2.137 EDMA3CC_EVTMUX_52_55 Register (offset = FC4h) [reset = 0h] ..... 1012
3.2.138 EDMA3CC_EVTMUX_56_59 Register (offset = FC8h) [reset = 0h] ..... 1013
3.2.139 EDMA3CC_EVTMUX_60_63 Register (offset = FCCh) [reset = 0h] ..... 1014
3.2.140 GPIO_MUX Register (offset = FD4h) [reset = 0h] ..... 1015
3.2.141 RESET_ISO Register (offset = 1000h) [reset = 0h] ..... 1016

**4 ROM Code Memory and Peripheral Booting ..... 1017**

4.1 Introduction ..... 1018
4.1.1 Acronyms and Naming Conventions ..... 1018
4.2 Overview ..... 1019
4.2.1 Architecture ..... 1019
4.2.2 Functionality ..... 1020
4.3 Memory Map ..... 1021
4.3.1 ROM Memory Map ..... 1021
4.3.2 RAM Memory Map ..... 1024
4.4 Startup and Configuration ..... 1026
4.4.1 ROM Code Start-up ..... 1026
4.4.2 CPU State at Public Startup ..... 1027
4.4.3 Clocking Configuration ..... 1027
4.5 Booting ..... 1027
4.5.1 Overview ..... 1027
4.5.2 Device List ..... 1028
4.6 Fast External Booting ..... 1030
4.6.1 Overview ..... 1030
4.6.2 External Booting ..... 1030
4.7 Memory Booting ..... 1031
4.7.1 Overview ..... 1031
4.7.2 XIP Memory ..... 1032
4.7.3 NAND ..... 1036
4.7.4 MMC/SD Cards ..... 1046
4.7.5 SPI ..... 1058
4.7.6 Blocks and Sectors Search Summary ..... 1059
4.8 Peripheral Booting ..... 1060
4.8.1 Overview ..... 1060
4.8.2 Boot Image Location and Size ..... 1060
4.8.3 Peripheral Boot Procedure Overview ..... 1060
4.8.4 EMAC Boot Procedure ..... 1060
4.8.5 PCIe Boot Procedure ..... 1062
4.8.6 UART Boot Procedure ..... 1064
4.9 Image Format ..... 1065
4.9.1 Overview ..... 1065
4.9.2 Table of Contents ..... 1066
4.10 Execution ..... 1067
4.10.1 Overview ..... 1067



Copyright © 2011–2016, Texas Instruments Incorporated

4.10.2 Execution ... 1067
4.11 Services for HLOS Support ... 1068
4.12 Tracing ... 1069
**5 DCAN Controller Area Network ... 1072**
5.1 Overview ... 1073
5.1.1 Features ... 1073
5.1.2 Functional Description ... 1073
5.2 CAN Operation ... 1075
5.2.1 CAN Module Initialization ... 1075
5.2.2 CAN Message Transfer (Normal Operation) ... 1077
5.2.3 Test Modes ... 1078
5.3 Dual Clock Source ... 1081
5.4 Interrupt Functionality ... 1082
5.4.1 Message Object Interrupts ... 1082
5.4.2 Status Change Interrupts ... 1082
5.4.3 Error Interrupts ... 1083
5.5 Local Power-Down Mode ... 1084
5.5.1 Entering Local Power-Down Mode ... 1084
5.5.2 Wakeup From Local Power Down ... 1084
5.6 Parity Check Mechanism ... 1086
5.6.1 Behavior on Parity Error ... 1086
5.6.2 Parity Testing ... 1086
5.7 Debug/Suspend Mode ... 1087
5.8 Configuration of Message Objects ... 1087
5.8.1 Configuration of a Transmit Object for Data Frames ... 1088
5.8.2 Configuration of a Transmit Object for Remote Frames ... 1088
5.8.3 Configuration of a Single Receive Object for Data Frames ... 1088
5.8.4 Configuration of a Single Receive Object for Remote Frames ... 1089
5.8.5 Configuration of a FIFO Buffer ... 1089
5.9 Message Handling ... 1090
5.9.1 Message Handler Overview ... 1090
5.9.2 Receive/Transmit Priority ... 1090
5.9.3 Transmission of Messages in Event Driven CAN Communication ... 1090
5.9.4 Updating a Transmit Object ... 1091
5.9.5 Changing a Transmit Object ... 1091
5.9.6 Acceptance Filtering of Received Messages ... 1092
5.9.7 Reception of Data Frames ... 1092
5.9.8 Reception of Remote Frames ... 1092
5.9.9 Reading Received Messages ... 1092
5.9.10 Requesting New Data for a Receive Object ... 1093
5.9.11 Storing Received Messages in FIFO Buffers ... 1093
5.9.12 Reading From a FIFO Buffer ... 1093
5.10 CAN Bit Timing ... 1095
5.10.1 Bit Time and Bit Rate ... 1095
5.10.2 DCAN Bit Timing Registers ... 1100
5.11 Message Interface Register Sets ... 1103
5.11.1 Message Interface Register Sets 1 and 2 ... 1103
5.11.2 IF3 Register Set ... 1104
5.12 Message RAM ... 1105
5.12.1 Structure of Message Objects ... 1106
5.12.2 Addressing Message Objects in RAM ... 1108
5.12.3 Message RAM Representation in Debug/Suspend Mode ... 1109
5.12.4 Message RAM Representation in Direct Access Mode ... 1110


Copyright © 2011–2016, Texas Instruments Incorporated

5.13 GIO Support ..... 1110
5.14 Registers ..... 1110
5.14.1 DCAN Control Registers ..... 1111
**6 DMM/TILER ..... 1152**
6.1 Introduction ..... 1153
6.1.1 Overview ..... 1153
6.1.2 Features ..... 1154
6.1.3 Functional Block Diagram ..... 1154
6.1.4 Terminologies and Acronyms Used in this Document ..... 1155
6.2 Architecture ..... 1156
6.2.1 DMM Functional Description ..... 1156
6.2.2 TILER Functional Description ..... 1167
6.3 Use Case ..... 1191
6.3.1 DMM Basic Register Setup ..... 1191
6.3.2 Simple LUT Bypass Use Case: Arrangement of Video Buffers ..... 1192
6.3.3 LUT Refill Using the PAT Refill Engines ..... 1194
6.3.4 Address Management Using LISA Sections ..... 1201
6.4 DMM/TILER Registers ..... 1206
6.4.1 DMM Revision Register: DMM_REVISION ..... 1206
6.4.2 DMM Clock Management Configuration: DMM_SYSCONFIG ..... 1207
6.4.3 LISA Configuration Locking Register: DMM_LISA_LOCK ..... 1207
6.4.4 DMM LISA MAP Registers: DMM_LISA_MAP_0-DMM_LISA_MAP_3 ..... 1208
6.4.5 DMM TILER Orientation Registers: DMM_TILER_OR0-DMM_TILER_OR1 ..... 1209
6.4.6 DMM PAT Configuration Register: DMM_PAT_CONFIG ..... 1209
6.4.7 DMM PAT View Registers: DMM_PAT_VIEW_0-DMM_PAT_VIEW_2 ..... 1210
6.4.8 DMM View Map Registers: DMM_PAT_VIEW_MAP_0-DMM_PAT_VIEW_MAP_4 ..... 1211
6.4.9 DMM PAT View Mapping Base Address Register: DMM_PAT_VIEW_MAP_BASE ..... 1212
6.4.10 DMM PAT End of Interrupt Register: DMM_PAT_IRQ_EOI ..... 1212
6.4.11 DMM PAT Raw Interrupt Status Register: DMM_PAT_IRQSTATUS_RAW ..... 1214
6.4.12 DMM PAT Interrupt Status Register: DMM_PAT_IRQSTATUS ..... 1216
6.4.13 DMM PAT Interrupt Enable Register: DMM_PAT_IRQENABLE_SET ..... 1218
6.4.14 DMM PAT Interrupt Disable Register: DMM_PAT_IRQENABLE_CLR ..... 1220
6.4.15 DMM PAT Status Registers: DMM_PAT_STATUS_0-DMM_PAT_STATUS_3 ..... 1222
6.4.16 DMM PAT Descriptor Registers: DMM_PAT_DESCR_0-DMM_PAT_DESCR_3 ..... 1223
6.4.17 DMM PAT Area Geometry Registers: DMM_PAT_AREA_0-DMM_PAT_AREA_3 ..... 1223
6.4.18 DMM PAT Control Registers: DMM_PAT_CTRL_0-DMM_PAT_CTRL_3 ..... 1224
6.4.19 DMM PAT Area Entry Data Registers: DMM_PAT_DATA_0-DMM_PAT_DATA_3 ..... 1224
6.4.20 DMM PEG Priority Registers: DMM_PEG_PRIO_0-DMM_PEG_PRIO_7 ..... 1225
6.4.21 DMM PEG Priority Registers for PAT: DMM_PEG_PRIO_PAT ..... 1225
**7 DDR2 and DDR3 Memory Controller ..... 1226**
7.1 Introduction ..... 1227
7.1.1 Purpose of the Peripheral ..... 1227
7.1.2 Features Supported ..... 1227
7.1.3 Features Not Supported ..... 1227
7.2 Architecture ..... 1227
7.2.1 Signal Descriptions ..... 1229
7.2.2 Memory Map ..... 1230
7.2.3 Clock Control ..... 1230
7.2.4 DDR2/DDR3 Memory Controller Subsytem Overview ..... 1230
7.2.5 Address Mapping ..... 1234
7.2.6 Performance Management ..... 1238
7.2.7 Performance Counter Usage ..... 1241
7.2.8 DDR3 Software Write/Read Leveling ..... 1242



Copyright © 2011–2016, Texas Instruments Incorporated

7.2.9 PRCM Sequence for DDR2/DDR3 Memory Controller ... 1242
7.2.10 Interrupt Support ... 1242
7.2.11 EDMA Event Support ... 1242
7.2.12 Emulation Considerations ... 1243
7.2.13 Power Management ... 1243
7.3 DDR PHY ... 1244
7.3.1 Functional Overview ... 1244
7.3.2 Data Slice Block ... 1245
7.3.3 Master DLL ... 1245
7.3.4 Ratio Logic Block ... 1245
7.3.5 Command Write, Write, and Read Operation ... 1245
7.4 Electrical Characteristics Control ... 1246
7.4.1 Output Impedance Calibration ... 1246
7.4.2 IO Configuration and Slew Rate Control ... 1246
7.4.3 ODT Control ... 1247
7.5 DDR2/DDR3 SDRAM Memory Initialization ... 1248
7.5.1 DDR2 SDRAM Memory Initialization ... 1248
7.5.2 DDR3 SDRAM Memory Initialization ... 1249
7.6 Interfacing the DDR2/DDR3 Memory Controller to DDR Memory Devices ... 1250
7.6.1 Configuring DDR2/DDR3 Memory Controller Registers to Meet DDR2 SDRAM Specifications ... 1250
7.6.2 Configuring DDR2/DDR3 Memory Controller Registers to Meet DDR3 SDRAM Specifications ... 1253
7.7 DDR2/DDR3 SDRAM Device Configuration Steps ... 1260
7.8 DDR2/DDR3 Configuration Registers ... 1261
7.8.1 DDR2/DDR3 Memory Controller Registers ... 1261
7.8.2 DDR2/DDR3 PHY Registers ... 1290
7.8.3 DDR Related Control Module Registers ... 1297

**8 Enhanced Direct Memory Access Controller ... 1303**

8.1 Overview ... 1304
8.2 Features ... 1304
8.3 Terminology Used in This Document ... 1305
8.3.1 Functional Overview ... 1307
8.3.2 Types of EDMA3 Transfers ... 1310
8.3.3 Parameter RAM (PaRAM) ... 1312
8.3.4 Initiating a DMA Transfer ... 1323
8.3.5 Completion of a DMA Transfer ... 1326
8.3.6 Event, Channel, and PaRAM Mapping ... 1327
8.3.7 EDMA3 Channel Controller Regions ... 1329
8.3.8 Chaining EDMA3 Channels ... 1331
8.3.9 EDMA3 Interrupts ... 1332
8.3.10 Memory Protection ... 1338
8.3.11 Event Queue(s) ... 1342
8.3.12 EDMA3 Transfer Controller (EDMA3TC) ... 1344
8.3.13 Event Dataflow ... 1347
8.3.14 EDMA3 Prioritization ... 1347
8.3.15 EDMA3 Operating Frequency (Clock Control) ... 1349
8.3.16 Reset Considerations ... 1349
8.3.17 Power Management ... 1349
8.3.18 Emulation Considerations ... 1349
8.4 EDMA Transfer Examples ... 1350
8.4.1 Block Move Example ... 1350
8.4.2 Subframe Extraction Example ... 1352
8.4.3 Data Sorting Example ... 1353
8.4.4 Peripheral Servicing Example ... 1355


Copyright © 2011–2016, Texas Instruments Incorporated

8.5 EDMA3 Registers ..... 1369
8.5.1 EDMA3 Channel Controller Registers ..... 1369
8.5.2 EDMA3 Transfer Controller Registers ..... 1421
8.6 Appendix A ..... 1446
8.6.1 Debug Checklist ..... 1446
8.6.2 Miscellaneous Programming/Debug Tips ..... 1447
8.7 Setting Up a Transfer ..... 1448
**9 3PSW Ethernet Subsystem (EMAC) ..... 1450**
9.1 Introduction ..... 1451
9.1.1 Features ..... 1451
9.1.2 Functional Block Diagram ..... 1452
9.1.3 Interface Mode Selection ..... 1452
9.1.4 Subsystem Signal Summary ..... 1452
9.1.5 Clocking ..... 1454
9.1.6 Interrupt Support ..... 1455
9.1.7 Memory Map ..... 1455
9.1.8 3PSW Ports ..... 1456
9.2 CPSW Architecture ..... 1457
9.2.1 CPSW_3G ..... 1457
9.2.2 Common Platform Time Sync (CPTS) ..... 1483
9.2.3 CPPI Buffer Descriptors ..... 1488
9.2.4 MDIO ..... 1495
9.2.5 Interrupts ..... 1497
9.2.6 Interrupt Pacing ..... 1501
9.2.7 Reset Isolation ..... 1502
9.2.8 CPSW_3G Network Statistics ..... 1502
9.3 Software Operation ..... 1503
9.3.1 Transmit Operation ..... 1503
9.3.2 Receive Operation ..... 1505
9.3.3 MDIO Software Interface ..... 1507
9.3.4 Initialization and Configuration of CPSW ..... 1508
9.4 Registers ..... 1509
9.4.1 CPSW_3G Module ..... 1509
9.4.2 MDIO Module ..... 1590
9.4.3 CPSW Subsytem Module ..... 1598
**10 General-Purpose I/O (GPIO) Interface ..... 1605**
10.1 Introduction ..... 1606
10.1.1 Purpose of the Peripheral ..... 1606
10.1.2 Features ..... 1606
10.1.3 Block Diagram ..... 1607
10.2 Architecture ..... 1608
10.2.1 Operating Modes ..... 1608
10.2.2 Clocking and Reset Strategy ..... 1608
10.2.3 Interrupt Features ..... 1609
10.2.4 General-Purpose Interface Basic Programming Model ..... 1611
10.3 GPIO Registers ..... 1615
10.3.1 GPIO_REVISION Register ..... 1616
10.3.2 GPIO_SYSCONFIG Register ..... 1617
10.3.3 GPIO_EOI Register ..... 1618
10.3.4 GPIO_IRQSTATUS_RAW_0 Register ..... 1619
10.3.5 GPIO_IRQSTATUS_RAW_1 Register ..... 1619
10.3.6 GPIO_IRQSTATUS_0 Register ..... 1620
10.3.7 GPIO_IRQSTATUS_1 Register ..... 1620


Copyright © 2011–2016, Texas Instruments Incorporated

10.3.8 GPIO_IRQENABLE_SET_0 Register ..... 1621
10.3.9 GPIO_IRQENABLE_SET_1 Register ..... 1621
10.3.10 GPIO_IRQENABLE_CLR_0 Register ..... 1622
10.3.11 GPIO_IRQENABLE_CLR_1 Register ..... 1622
10.3.12 GPIO_SYSSTATUS Register ..... 1623
10.3.13 GPIO_CTRL Register ..... 1624
10.3.14 GPIO_OE Register ..... 1624
10.3.15 GPIO_DATAIN Register ..... 1625
10.3.16 GPIO_DATAOUT Register ..... 1625
10.3.17 GPIO_LEVELDETECT0 Register ..... 1626
10.3.18 GPIO_LEVELDETECT1 Register ..... 1626
10.3.19 GPIO_RISINGDETECT Register ..... 1627
10.3.20 GPIO_FALLINGDETECT Register ..... 1627
10.3.21 GPIO_DEBOUNCENABLE Register ..... 1628
10.3.22 GPIO_DEBOUNCINGTIME Register ..... 1628
10.3.23 GPIO_CLEARDATAOUT Register ..... 1629
10.3.24 GPIO_SETDATAOUT Register ..... 1629

**11 General-Purpose Memory Controller (GPMC) ..... 1630**

11.1 Introduction ..... 1631
11.1.1 Overview ..... 1631
11.1.2 Block Diagram ..... 1631
11.2 Architecture ..... 1633
11.2.1 GPMC Signals ..... 1633
11.2.2 GPMC Modes ..... 1635
11.2.3 GPMC Integration ..... 1637
11.2.4 GPMC Functional Description ..... 1638
11.3 Basic Programming Model ..... 1712
11.3.1 GPMC High-Level Programming Model Overview ..... 1712
11.3.2 GPMC Initialization ..... 1714
11.3.3 GPMC Configuration in NOR Mode ..... 1714
11.3.4 GPMC Configuration in NAND Mode ..... 1715
11.3.5 Set Memory Access ..... 1717
11.3.6 GPMC Timing Parameters ..... 1718
11.4 Use Cases And Tips ..... 1731
11.4.1 How to Set GPMC Timing Parameters for Typical Accesses ..... 1731
11.4.2 How to Choose a Suitable Memory to Use With the GPMC ..... 1739
11.5 Registers ..... 1743
11.5.1 GPMC_REVISION ..... 1744
11.5.2 GPMC_SYSCONFIG ..... 1744
11.5.3 GPMC_SYSSTATUS ..... 1745
11.5.4 GPMC_IRQSTATUS ..... 1746
11.5.5 GPMC_IRQENABLE ..... 1747
11.5.6 GPMC_TIMEOUT_CONTROL ..... 1748
11.5.7 GPMC_ERR_ADDRESS ..... 1748
11.5.8 GPMC_ERR_TYPE ..... 1749
11.5.9 GPMC_CONFIG ..... 1750
11.5.10 GPMC_STATUS ..... 1751
11.5.11 GPMC_CONFIG1_i ..... 1752
11.5.12 GPMC_CONFIG2_i ..... 1755
11.5.13 GPMC_CONFIG3_i ..... 1756
11.5.14 GPMC_CONFIG4_i ..... 1758
11.5.15 GPMC_CONFIG5_i ..... 1760
11.5.16 GPMC_CONFIG6_i ..... 1761


Copyright © 2011–2016, Texas Instruments Incorporated

11.5.17 GPMC_CONFIG7_i ... 1762
11.5.18 GPMC_NAND_COMMAND_i ... 1763
11.5.19 GPMC_NAND_ADDRESS_i ... 1763
11.5.20 GPMC_NAND_DATA_i ... 1763
11.5.21 GPMC_PREFETCH_CONFIG1 ... 1764
11.5.22 GPMC_PREFETCH_CONFIG2 ... 1766
11.5.23 GPMC_PREFETCH_CONTROL ... 1766
11.5.24 GPMC_PREFETCH_STATUS ... 1767
11.5.25 GPMC_ECC_CONFIG ... 1768
11.5.26 GPMC_ECC_CONTROL ... 1769
11.5.27 GPMC_ECC_SIZE_CONFIG ... 1770
11.5.28 GPMC_ECCj_RESULT ... 1772
11.5.29 GPMC_BCH_RESULT0_i ... 1773
11.5.30 GPMC_BCH_RESULT1_i ... 1773
11.5.31 GPMC_BCH_RESULT2_i ... 1773
11.5.32 GPMC_BCH_RESULT3_i ... 1774
11.5.33 GPMC_BCH_RESULT4_i ... 1774
11.5.34 GPMC_BCH_RESULT5_i ... 1775
11.5.35 GPMC_BCH_RESULT6_i ... 1775
11.5.36 GPMC_BCH_SWDATA ... 1775

**12 High-Definition Video Processing Subsystem (HDVPSS) ... 1776**

12.1 Description of the Subsystem ... 1777
12.1.1 Overview ... 1777
12.1.2 Acronyms and Definitions ... 1777
12.1.3 Features ... 1778
12.1.4 Functional Operation ... 1782
12.1.5 HDVPSS Data Flow ... 1787
12.1.6 Example Data Flows ... 1790
12.1.7 Interrupt Mapping ... 1802
12.1.8 Clocking ... 1806
12.1.9 Reset ... 1807
12.2 Internal Modules ... 1808
12.2.1 Chroma Up-Sampler (CHR_US) ... 1808
12.2.2 Color Space Converter (CSC) ... 1813
12.2.3 Compositor Module ... 1819
12.2.4 Constrained Image Generator (CIG) ... 1826
12.2.5 Graphics Module (GRPX) ... 1829
12.2.6 Standard-Definition Video Encoder (SD_VENC) ... 1841
12.2.7 High-Definition Video Encoder (HD_VENC) ... 1875
12.2.8 VIP Parser ... 1887
12.2.9 De-Interlacer (DEI) Module ... 1926
12.2.10 Noise Filter (NF) ... 1932
12.2.11 Scaler (SC) ... 1938
12.2.12 Video Compositer (VCOMP) ... 1974
12.2.13 Video Port Direct Memory Access (VPDMA) ... 1980
12.3 Registers ... 2061
12.3.1 CHR_US Registers ... 2063
12.3.2 CIG Registers ... 2072
12.3.3 COMP Registers ... 2084
12.3.4 CSC Registers ... 2090
12.3.5 DEI Registers ... 2097
12.3.6 GRPX Registers ... 2113
12.3.7 INTC_CLKC_CONTROL Registers ... 2114


Copyright © 2011–2016, Texas Instruments Incorporated

12.3.8 VPDMA Registers ..... 2196
12.3.9 HD_VENC_D Registers ..... 2496
12.3.10 NOISE_FILTER Registers ..... 2525
12.3.11 SC_M Registers ..... 2536
12.3.12 SD_VENC Registers ..... 2560
12.3.13 VCOMP Registers ..... 2616
12.3.14 VIP_PARSER Registers ..... 2630

**13 High-Definition Multimedia Interface (HDMI) ..... 2699**
13.1 Introduction ..... 2700
13.1.1 Overview ..... 2700
13.1.2 Main Features ..... 2700
13.1.3 Functional Block Diagram ..... 2701
13.1.4 Video Formats and Timings ..... 2701
13.1.5 Environment ..... 2703
13.2 Architecture ..... 2704
13.2.1 Clock Configuration ..... 2704
13.2.2 Signals ..... 2705
13.2.3 Integration ..... 2705
13.2.4 Software Reset ..... 2705
13.2.5 Power Management ..... 2707
13.2.6 Interrupt Requests ..... 2708
13.2.7 DMA Requests ..... 2708
13.2.8 Wrapper Functions ..... 2708
13.2.9 TXPHY Functions ..... 2718
13.2.10 Programming Video Input Mode and Video Output Mode ..... 2719
13.3 HDMI Registers ..... 2720
13.3.1 HDMI Wrapper Registers ..... 2720
13.3.2 HDMI Core System Registers ..... 2733
13.3.3 HDMI IP Core Gamut Registers ..... 2805
13.3.4 HDMI IP Core Audio Video Registers ..... 2808
13.3.5 HDMI IP Core CEC Registers ..... 2844
13.3.6 HDMI PHY Registers ..... 2860

**14 ARM Interrupt Controller (AINTC) ..... 2863**
14.1 Introduction ..... 2864
14.1.1 Overview ..... 2864
14.1.2 Limitations ..... 2864
14.1.3 Functional Block Diagram ..... 2864
14.1.4 Environment ..... 2864
14.1.5 ARM A8—Interrupt Controller Integration ..... 2866
14.2 Architecture ..... 2867
14.2.1 Clocking and Reset ..... 2867
14.2.2 Interrupt Request Lines ..... 2867
14.2.3 Interrupt Processing ..... 2867
14.2.4 Module Power Saving ..... 2869
14.2.5 Error Handling ..... 2869
14.2.6 Interrupt Latency ..... 2869
14.3 Basic Programming Model ..... 2870
14.3.1 Initialization Sequence ..... 2870
14.3.2 INTC Processing Sequence ..... 2870
14.3.3 INTC Preemptive Processing Sequence ..... 2874
14.3.4 Interrupt Preemption ..... 2878
14.3.5 ARM A8 INTC Spurious Interrupt Handling ..... 2878
14.4 Registers ..... 2879


Copyright © 2011–2016, Texas Instruments Incorporated

14.4.1 INTCPS_REVISION Register ... 2880
14.4.2 INTCPS_SYSCONFIG Register ... 2880
14.4.3 INTCPS_SYSSTATUS Register ... 2881
14.4.4 INTCPS_SIR_IRQ Register ... 2881
14.4.5 INTCPS_SIR_FIQ Register ... 2882
14.4.6 INTCPS_CONTROL Register ... 2882
14.4.7 INTCPS_PROTECTION Register ... 2883
14.4.8 INTCPS_IDLE Register ... 2883
14.4.9 INTCPS_IRQ_PRIORITY Register ... 2884
14.4.10 INTCPS_FIQ_PRIORITY Register ... 2884
14.4.11 INTCPS_THRESHOLD Register ... 2885
14.4.12 INTCPS_ITR0-3 Registers ... 2885
14.4.13 INTCPS_MIR0-3 Registers ... 2886
14.4.14 INTCPS_MIR_CLEAR0-3 Registers ... 2886
14.4.15 INTCPS_MIR_SET0-3 Registers ... 2886
14.4.16 INTCPS_ISR_SET0-3 Registers ... 2887
14.4.17 INTCPS_ISR_CLEAR0-3 Registers ... 2887
14.4.18 INTCPS_PENDING_IRQ0-3 Registers ... 2887
14.4.19 INTCPS_PENDING_FIQ0-3 Registers ... 2888
14.4.20 INTCPS_ILR0-127 Registers ... 2888

**15 Inter-Integrated Circuit (I2C) Controller Module ... 2889**

15.1 Introduction ... 2890
15.1.1 Overview ... 2890
15.1.2 Functional Block Diagram ... 2890
15.1.3 Features ... 2890
15.2 Architecture ... 2891
15.2.1 I2C Master/Slave Contoller Signals ... 2891
15.2.2 I2C Reset ... 2892
15.2.3 Data Validity ... 2892
15.2.4 START & STOP Conditions ... 2893
15.2.5 I2C Operation ... 2893
15.2.6 Arbitration ... 2895
15.2.7 I2C Clock Generation and I2C Clock Synchronization ... 2895
15.2.8 Prescaler (SCLK/ICLK) ... 2896
15.2.9 Noise Filter ... 2896
15.2.10 I2C Interrupts ... 2896
15.2.11 DMA Events ... 2897
15.2.12 Interrupt and DMA Events ... 2897
15.2.13 FIFO Management ... 2897
15.2.14 How to Program I2C ... 2901
15.3 I2C Registers ... 2903
15.3.1 I2C_REVNB_LO Register (Module Revision LOW BYTES) ... 2904
15.3.2 I2C_REVNB_HI Register (HIGH BYTES) (Module Revision) ... 2905
15.3.3 I2C_SYSC Register (System Configuration) ... 2906
15.3.4 I2C_EOI Register (I2C End of Interrupt) ... 2907
15.3.5 I2C_IRQSTATUS_RAW Register (I2C Status Raw) ... 2908
15.3.6 I2C_IRQSTATUS Register (I2C Status) ... 2912
15.3.7 I2C_IRQENABLE_SET Register (I2C Interrupt Enable Set) ... 2914
15.3.8 I2C_IRQENABLE_CLR Register (I2C Interrupt Enable Clear) ... 2916
15.3.9 I2C_WE Register (I2C Wakeup Enable) ... 2918
15.3.10 I2C_DMARXENABLE_SET Register (Receive DMA Enable Set) ... 2921
15.3.11 I2C_DMATXENABLE_SET Register (Transmit DMA Enable Set) ... 2921
15.3.12 I2C_DMARXENABLE_CLR Register (Receive DMA Enable Clear) ... 2922


Copyright © 2011–2016, Texas Instruments Incorporated

15.3.13 I2C_DMATXENABLE_CLR Register (Transmit DMA Enable Clear) ..... 2922
15.3.14 I2C_DMARXWAKE_EN Register (Receive DMA Wakeup) ..... 2923
15.3.15 I2C_DMATXWAKE_EN Register (Transmit DMA Wakeup) ..... 2925
15.3.16 I2C_SYSS Register (System Status) ..... 2927
15.3.17 I2C_BUF Register (Buffer Configuration) ..... 2928
15.3.18 I2C_CNT Register (Data Counter) ..... 2930
15.3.19 I2C_DATA Register (Data Access) ..... 2931
15.3.20 I2C_CON Register (I2C Configuration) ..... 2932
15.3.21 I2C_OA Register (I2C Own Address) ..... 2934
15.3.22 I2C_SA Register (I2C Slave Address) ..... 2935
15.3.23 I2C_PSC Register (I2C Clock Prescaler) ..... 2936
15.3.24 I2C_SCLL Register (I2C SCL Low Time) ..... 2937
15.3.25 I2C_SCLH Register (I2C SCL High Time) ..... 2937
15.3.26 I2C_SYSTEST Register (System Test) ..... 2938
15.3.27 I2C_BUFSTAT Register (I2C Buffer Status) ..... 2941
15.3.28 I2C_OA1 Register (OA1) (Own Address 1) ..... 2942
15.3.29 I2C_OA2 Register (I2C Own Address 2) ..... 2943
15.3.30 I2C_OA3 Register (I2C Own Address 3) ..... 2944
15.3.31 I2C_ACTOA Register (Active Own Address) ..... 2945
15.3.32 I2C_SBLOCK Register (I2C Clock Blocking Enable) ..... 2946

**16 Multichannel Audio Serial Port (McASP) ..... 2947**

16.1 Introduction ..... 2948
16.1.1 Purpose of the Peripheral ..... 2948
16.1.2 Features ..... 2948
16.1.3 Protocols Supported ..... 2949
16.1.4 Functional Block Diagram ..... 2950
16.1.5 Supported Use Case Statement ..... 2953
16.1.6 Industry Standard Compliance Statement ..... 2953
16.1.7 Definition of Terms ..... 2957
16.2 Architecture ..... 2960
16.2.1 Overview ..... 2960
16.2.2 Clock and Frame Sync Generators ..... 2960
16.2.3 Memory Map ..... 2964
16.2.4 Signal Descriptions ..... 2964
16.2.5 Pin Multiplexing ..... 2964
16.2.6 Transfer Modes ..... 2965
16.2.7 General Architecture ..... 2972
16.2.8 Operation ..... 2977
16.2.9 Reset Considerations ..... 2994
16.2.10 Setup and Initialization ..... 2994
16.2.11 Interrupts ..... 2998
16.2.12 EDMA Event Support ..... 3000
16.2.13 Power Management ..... 3002
16.2.14 Emulation Considerations ..... 3002
16.3 McASP Registers ..... 3003
16.3.1 Revision Identification Register (REV) ..... 3006
16.3.2 Pin Function Register (PFUNC) ..... 3006
16.3.3 Pin Direction Register (PDIR) ..... 3008
16.3.4 Pin Data Output Register (PDOUT) ..... 3010
16.3.5 Pin Data Input Register (PDIN) ..... 3012
16.3.6 Pin Data Set Register (PDSET) ..... 3014
16.3.7 Pin Data Clear Register (PDCLR) ..... 3016
16.3.8 Global Control Register (GBLCTL) ..... 3018


Copyright © 2011–2016, Texas Instruments Incorporated

16.3.9 Audio Mute Control Register (AMUTE) ... 3020
16.3.10 Digital Loopback Control Register (DLBCTL) ... 3022
16.3.11 Digital Mode Control Register (DITCTL) ... 3023
16.3.12 Receiver Global Control Register (RGBLCTL) ... 3024
16.3.13 Receive Format Unit Bit Mask Register (RMASK) ... 3025
16.3.14 Receive Bit Stream Format Register (RFMT) ... 3026
16.3.15 Receive Frame Sync Control Register (AFSRCTL) ... 3028
16.3.16 Receive Clock Control Register (ACLKRCTL) ... 3029
16.3.17 Receive High-Frequency Clock Control Register (AHCLKRCTL) ... 3030
16.3.18 Receive TDM Time Slot Register (RTDM) ... 3031
16.3.19 Receiver Interrupt Control Register (RINTCTL) ... 3032
16.3.20 Receiver Status Register (RSTAT) ... 3033
16.3.21 Current Receive TDM Time Slot Registers (RSLOT) ... 3034
16.3.22 Receive Clock Check Control Register (RCLKCHK) ... 3035
16.3.23 Receiver DMA Event Control Register (REVTCTL) ... 3036
16.3.24 Transmitter Global Control Register (XGBLCTL) ... 3037
16.3.25 Transmit Format Unit Bit Mask Register (XMASK) ... 3038
16.3.26 Transmit Bit Stream Format Register (XFMT) ... 3039
16.3.27 Transmit Frame Sync Control Register (AFSXCTL) ... 3041
16.3.28 Transmit Clock Control Register (ACLKXCTL) ... 3042
16.3.29 Transmit High-Frequency Clock Control Register (AHCLKXCTL) ... 3043
16.3.30 Transmit TDM Time Slot Register (XTDM) ... 3044
16.3.31 Transmitter Interrupt Control Register (XINTCTL) ... 3045
16.3.32 Transmitter Status Register (XSTAT) ... 3046
16.3.33 Current Transmit TDM Time Slot Register (XSLOT) ... 3047
16.3.34 Transmit Clock Check Control Register (XCLKCHK) ... 3048
16.3.35 Transmitter DMA Event Control Register (XEVTCTL) ... 3049
16.3.36 Serializer Control Registers (SRCTL*n*) ... 3050
16.3.37 DIT Left Channel Status Registers (DITCSRA0-DITCSRA5) ... 3051
16.3.38 DIT Right Channel Status Registers (DITCSRB0-DITCSRB5) ... 3051
16.3.39 DIT Left Channel User Data Registers (DITUDRA0-DITUDRA5) ... 3051
16.3.40 DIT Right Channel User Data Registers (DITUDRB0-DITUDRB5) ... 3051
16.3.41 Transmit Buffer Registers (XBUF*n*) ... 3052
16.3.42 Receive Buffer Registers (RBUF*n*) ... 3052
16.3.43 Write FIFO Control Register (WFIFOCTL) ... 3053
16.3.44 Write FIFO Status Register (WFIFOSTS) ... 3054
16.3.45 Read FIFO Control Register (RFIFOCTL) ... 3055
16.3.46 Read FIFO Status Register (RFIFOSTS) ... 3056

**17 Multichannel Buffered Serial Port (McBSP) ... 3057**

17.1 Introduction ... 3058
17.1.1 Overview ... 3058
17.1.2 Features ... 3058
17.1.3 Block Diagram ... 3059
17.2 Architecture ... 3060
17.2.1 McBSP Data Transfer Process ... 3060
17.2.2 McBSP Sample Rate Generator ... 3067
17.2.3 McBSP Exception/Error Conditions ... 3072
17.2.4 McBSP DMA Configuration ... 3077
17.2.5 Multichannel Selection Modes ... 3078
17.2.6 McBSP Full/Half Cycle Modes ... 3085
17.2.7 Power Management ... 3087
17.2.8 Programming Model ... 3089
17.3 McBSP Registers ... 3110



Copyright © 2011–2016, Texas Instruments Incorporated

17.3.1 Revision Number Register (REVNB) ..... 3112
17.3.2 System Configuration Register (SYSCONFIG_REG) ..... 3113
17.3.3 End of Interrupt Register (EOI) ..... 3114
17.3.4 Interrupt Status Raw Register (IRQSTATUS_RAW) ..... 3114
17.3.5 Interrupt Status Register (IRQSTATUS) ..... 3117
17.3.6 Interrupt Enable Set Register (IRQENABLE_SET) ..... 3119
17.3.7 Interrupt Enable Clear Register (IRQENABLE_CLR) ..... 3121
17.3.8 DMA Rx Enable Set Register (DMARXENABLE_SET) ..... 3123
17.3.9 DMA Tx Enable Set Register (DMATXENABLE_SET) ..... 3123
17.3.10 DMA Rx Enable Clear Register (DMARXENABLE_CLR) ..... 3124
17.3.11 DMA Tx Enable Clear Register (DMATXENABLE_CLR) ..... 3124
17.3.12 DMA Rx Wake Enable Register (DMARXWAKE_EN) ..... 3125
17.3.13 DMA Tx Wake Enable Register (DMATXWAKE_EN) ..... 3127
17.3.14 McBSP Data Receive Register (DRR_REG) ..... 3129
17.3.15 McBSP Data Transmit Register (DXR_REG) ..... 3129
17.3.16 McBSP Serial Port Control Register 2 (SPCR2_REG) ..... 3129
17.3.17 McBSP Serial Port Control Register 1 (SPCR1_REG) ..... 3131
17.3.18 McBSP Receive Control Register 2 (RCR2_REG) ..... 3132
17.3.19 McBSP Receive Control Register 1 (RCR1_REG) ..... 3133
17.3.20 McBSP Transmit Control Register 2 (XCR2_REG) ..... 3134
17.3.21 McBSP Transmit Control Register 1 (XCR1_REG) ..... 3135
17.3.22 McBSP Sample Rate Generator Register 2 (SRGR2_REG) ..... 3136
17.3.23 McBSP Sample Rate Generator Register 1 (SRGR1_REG) ..... 3137
17.3.24 McBSP Multichannel Register 2 (MCR2_REG) ..... 3138
17.3.25 McBSP Multichannel Register 1 (MCR1_REG) ..... 3140
17.3.26 McBSP Receive Channel Enable Register Partition A (RCERA_REG) ..... 3142
17.3.27 McBSP Receive Channel Enable Register Partition B (RCERB_REG) ..... 3142
17.3.28 McBSP Transmit Channel Enable Register Partition A (XCERA_REG) ..... 3143
17.3.29 McBSP Transmit Channel Enable Register Partition B (XCERB_REG) ..... 3143
17.3.30 McBSP Pin Control Register (PCR_REG) ..... 3144
17.3.31 McBSP Receive Channel Enable Register Partition C (RCERC_REG) ..... 3146
17.3.32 McBSP Receive Channel Enable Register Partition D (RCERD_REG) ..... 3146
17.3.33 McBSP Transmit Channel Enable Register Partition C (XCERC_REG) ..... 3147
17.3.34 McBSP Transmit Channel Enable Register Partition D (XCERD_REG) ..... 3147
17.3.35 McBSP Receive Channel Enable Register Partition E (RCERE_REG) ..... 3148
17.3.36 McBSP Receive Channel Enable Register Partition F (RCERF_REG) ..... 3148
17.3.37 McBSP Transmit Channel Enable Register Partition E (XCERE_REG) ..... 3149
17.3.38 McBSP Transmit Channel Enable Register Partition F (XCERF_REG) ..... 3149
17.3.39 McBSP Receive Channel Enable Register Partition G (RCERG_REG) ..... 3150
17.3.40 McBSP Receive Channel Enable Register Partition H (RCERH_REG) ..... 3150
17.3.41 McBSP Transmit Channel Enable Register Partition G (XCERG_REG) ..... 3151
17.3.42 McBSP Transmit Channel Enable Register Partition H (XCERH_REG) ..... 3151
17.3.43 McBSP Transmit Buffer Threshold Register (DMA or IRQ Trigger) (THRSH2_REG) ..... 3152
17.3.44 McBSP Receive Buffer Threshold Register (DMA or IRQ trigger) (THRSH1_REG) ..... 3152
17.3.45 McBSP Interrupt Status Register (OCP compliant IRQ line) (IRQSTATATUS) ..... 3153
17.3.46 McBSP Interrupt Enable Register (OCP compliant IRQ line) (IRQENABLE) ..... 3155
17.3.47 McBSP Wakeup Enable Register (WAKEUPEN) ..... 3157
17.3.48 McBSP Transmit Configuration Control Register (XCCR_REG) ..... 3158
17.3.49 McBSP Receive Configuration Control Register (RCCR_REG) ..... 3160
17.3.50 McBSP Transmit Buffer Status Registers (XBUFFSTAT_REG) ..... 3161
17.3.51 McBSP Receive Buffer Status Registers (RBUFFSTAT_REG) ..... 3161
17.3.52 McBSP_STATUS_REG ..... 3162

**18 Multimedia Card (MMC)/Secure Digital (SD)/ Secure Digital I/O (SDIO) Card Interface ..... 3163**


Copyright © 2011–2016, Texas Instruments Incorporated

18.1 Introduction ... 3164
18.1.1 Overview ... 3164
18.1.2 Features ... 3165
18.2 Architecture ... 3166
18.2.1 MMC/SD/SDIO Functional Modes ... 3166
18.2.2 Resets ... 3173
18.2.3 Clocks ... 3174
18.2.4 Power Management ... 3174
18.2.5 Interrupt Requests ... 3177
18.2.6 DMA Modes ... 3180
18.2.7 Mode Selection ... 3183
18.2.8 Buffer Management ... 3183
18.2.9 Transfer Process ... 3186
18.2.10 Transfer or Command Status and Error Reporting ... 3187
18.2.11 Auto Command 12 Timings ... 3191
18.2.12 Transfer Stop ... 3193
18.2.13 Output Signals Generation ... 3194
18.2.14 Card Boot Mode Management ... 3196
18.2.15 CE-ATA Command Completion Disable Management ... 3197
18.2.16 Test Registers ... 3197
18.2.17 MMC/SD/SDIO Hardware Status Features ... 3198
18.3 Low-Level Programming Models ... 3199
18.3.1 Surrounding Modules Global Initialization ... 3199
18.3.2 MMC/SD/SDIO Controller Initialization Flow ... 3199
18.3.3 Operational Modes Configuration ... 3202
18.4 MMC/SD/SDIO Registers ... 3204
18.4.1 IP Revision Identifier Register (MMCHS_HL_REV) ... 3205
18.4.2 IP Module Hardware Configuration Register (MMCHS_HL_HWINFO) ... 3206
18.4.3 Clock Management Configuration (MMCHS_HL_SYSCONFIG) ... 3207
18.4.4 System Configuration Register (MMCHS_SYSCONFIG) ... 3208
18.4.5 System Status Register (MMCHS_SYSSTATUS) ... 3210
18.4.6 Card Status Response Error (MMCHS_CSRE) ... 3210
18.4.7 System Test Register (MMCHS_SYSTEST) ... 3211
18.4.8 Configuration Register (MMCHS_CON) ... 3215
18.4.9 Power Counter Register (MMCHS_PWCNT) ... 3219
18.4.10 SDMA System Address (MMCHS_SDMASA) ... 3219
18.4.11 Transfer Length Configuration Register (MMCHS_BLK) ... 3220
18.4.12 Command Argument Register (MMCHS_ARG) ... 3221
18.4.13 Command and Transfer Mode Register (MMCHS_CMD) ... 3221
18.4.14 Command Response[31:0] Register (MMCHS_RSP10) ... 3225
18.4.15 Command Response[63:32] Register (MMCHS_RSP32) ... 3225
18.4.16 Command Response[95:64] Register (MMCHS_RSP54) ... 3226
18.4.17 Command Response[127:96] Register (MMCHS_RSP76) ... 3226
18.4.18 Data Register (MMCHS_DATA) ... 3227
18.4.19 Present State Register (MMCHS_PSTATE) ... 3228
18.4.20 Control Register (MMCHS_HCTL) ... 3230
18.4.21 SD System Control Register (MMCHS_SYSCTL) ... 3233
18.4.22 Interrupt Status Register (MMCHS_STAT) ... 3235
18.4.23 Interrupt SD Enable Register (MMCHS_IE) ... 3240
18.4.24 Interrupt Signal Enable Register (MMCHS_ISE) ... 3243
18.4.25 Auto CMD12 Error Status Register (MMCHS_AC12) ... 3246
18.4.26 Capabilities Register (MMCHS_CAPA) ... 3247
18.4.27 Force Event Register for Error Interrupt Status (MMCHS_FE) ... 3249


Copyright © 2011–2016, Texas Instruments Incorporated

18.4.28 ADMA Error Status Register (MMCHS_ADMAES) ... 3251
18.4.29 ADMA System Address Low Bits Register (MMCHS_ADMASAL) ... 3252
18.4.30 ADMA System Address High Bits Register (MMCHS_ADMASAH) ... 3252
18.4.31 Versions Register (MMCHS_REV) ... 3253

**19 Peripheral Component Interconnect Express (PCIe) Module ... 3254**
19.1 Introduction / Feature Overview ... 3255
19.1.1 Purpose of the Peripheral ... 3255
19.1.2 Terminology Used in This Document ... 3255
19.1.3 Features Supported ... 3256
19.1.4 Functional Block Diagram ... 3256
19.1.5 Supported Use Case Statement ... 3258
19.1.6 Industry Standard(s) Compliance Statement ... 3258
19.1.7 Features Not Supported ... 3258
19.2 Peripheral Architecture ... 3259
19.2.1 Clock Control ... 3259
19.2.2 Supported PCIe Transactions ... 3259
19.2.3 Address Translations ... 3259
19.2.4 Address Spaces ... 3265
19.2.5 Bus Mastering ... 3267
19.2.6 PCIe Loopback ... 3267
19.2.7 L3 Memory Map ... 3268
19.2.8 Signal Descriptions ... 3269
19.2.9 Reset Considerations ... 3269
19.2.10 Interrupt Support ... 3269
19.2.11 DMA Support ... 3272
19.2.12 Power Management ... 3273
19.2.13 OCP Power Management ... 3276
19.3 Supported Use Case ... 3277
19.3.1 Software Configuration ... 3277
19.3.2 Accessing Read-only Registers in Configuration Space ... 3280
19.3.3 Accessing EP Application Registers from PCIe RC ... 3280
19.4 Encoding of LTSSM State in DEBUG registers ... 3281
19.5 PCIESS Memory Map ... 3281
19.5.1 Application Registers ... 3281
19.5.2 Configuration Registers Common to Type 0 and Type 1 Headers ... 3323
19.5.3 Configuration Type 0 Registers ... 3325
19.5.4 Configuration Type 1 Registers ... 3333
19.5.5 Power Management Capabilities Registers ... 3340
19.5.6 Message Signaled Interrupts Registers ... 3341
19.5.7 PCI Express Capabilities Registers ... 3343
19.5.8 PCI Express Extended Capabilities Registers ... 3351
19.5.9 Port Logic Registers ... 3359

**20 Real-Time Clock (RTC) ... 3367**
20.1 Introduction ... 3368
20.1.1 Overview ... 3368
20.1.2 Features ... 3368
20.1.3 Functional Block Diagram ... 3368
20.2 Architecture ... 3369
20.2.1 Clock Source ... 3369
20.2.2 Signal Descriptions ... 3369
20.2.3 Interrupt Support ... 3370
20.2.4 Programming/Usage Guide ... 3371
20.2.5 Scratch Registers ... 3375


Copyright © 2011–2016, Texas Instruments Incorporated

20.2.6 Power Management ..... 3375
20.2.7 Reset Considerations ..... 3375
20.3 Registers ..... 3376
20.3.1 Seconds Register (SECONDS_REG) ..... 3377
20.3.2 Minutes Register (MINUTES_REG) ..... 3377
20.3.3 Hours Register (HOURS_REG) ..... 3378
20.3.4 Days of the Month Register (DAYS_REG) ..... 3378
20.3.5 Month Register (MONTHS_REG) ..... 3379
20.3.6 Year Register (YEARS_REG) ..... 3379
20.3.7 Day of the Week Register (WEEKS_REG) ..... 3380
20.3.8 Alarm Second Register (ALARM_SECONDS_REG) ..... 3380
20.3.9 Alarm Minute Register (ALARM_MINUTES_REG) ..... 3381
20.3.10 Alarm Hour Register (ALARM_HOURS_REG) ..... 3381
20.3.11 Alarm Day of the Month Register (ALARM_DAYS_REG) ..... 3382
20.3.12 Alarm Month Register (ALARM_MONTHS_REG) ..... 3382
20.3.13 Alarm Year Register (ALARM_YEARS_REG) ..... 3383
20.3.14 Control Register (CTRL_REG) ..... 3384
20.3.15 Status Register (STATUS_REG) ..... 3386
20.3.16 Interrupt Register (INTERRUPTS_REG) ..... 3387
20.3.17 Compensation (LSB) Register (COMP_LSB_REG) ..... 3388
20.3.18 Compensation (MSB) Register (COMP_MSB_REG) ..... 3389
20.3.19 Oscillator Register (OSC_REG) ..... 3390
20.3.20 Scratch Registers (SCRATCHx_REG) ..... 3390
20.3.21 Kick Registers (KICK0R, KICK1R) ..... 3391
20.3.22 RTC Revision Register (RTC_REVISION) ..... 3392
20.3.23 System Configuration Register (RTC_SYSCONFIG) ..... 3392
20.3.24 Wakeup Enable Register (RTC_IRQWAKEEN) ..... 3393

**21 Serial ATA (SATA) Controller ..... 3394**

21.1 Introduction ..... 3395
21.1.1 Purpose of the Peripheral ..... 3395
21.1.2 Features Supported ..... 3395
21.1.3 Features Not Supported ..... 3396
21.1.4 Functional Block Diagram ..... 3397
21.1.5 Industry Standard(s) Compliance ..... 3397
21.1.6 Non-Industry Standard(s) Compliance ..... 3397
21.1.7 Terminology Used in this Document ..... 3398
21.2 Architecture ..... 3399
21.2.1 Clock Control ..... 3399
21.2.2 Signal Description ..... 3400
21.2.3 DMA ..... 3400
21.2.4 Transport Layer ..... 3400
21.2.5 FIFOs ..... 3401
21.2.6 Link Layer ..... 3401
21.2.7 PHY ..... 3401
21.2.8 Pin Multiplexing ..... 3401
21.2.9 Power Management ..... 3401
21.2.10 Reset ..... 3402
21.2.11 Interfacing to Single and Multiple Devices ..... 3402
21.2.12 Initialization ..... 3402
21.2.13 Interrupt Support ..... 3404
21.2.14 EDMA Event Support ..... 3405
21.3 Use Cases ..... 3405
21.3.1 Controller Clock, PLL and SERDES Initialization ..... 3405



Copyright © 2011–2016, Texas Instruments Incorporated

21.3.2 General Utilities: Structures and Subroutines Sample Program Uses ..... 3406
21.3.3 Example on Initialization and Spinning Up Device ..... 3420
21.3.4 Example of DMA Write Transfer ..... 3422
21.3.5 Example of DMA Read Transfer ..... 3423
21.4 Registers (Controller and PHY) ..... 3424
21.4.1 HBA Capabilities Register (CAP) ..... 3426
21.4.2 Global HBA Control Register (GHC) ..... 3427
21.4.3 Interrupt Status Register (IS) ..... 3428
21.4.4 Ports Implemented Register (PI) ..... 3429
21.4.5 AHCI Version Register (VS) ..... 3429
21.4.6 Command Completion Coalescing Control Register (CCC_CTL) ..... 3430
21.4.7 Command Completion Coalescing Ports Register (CCC_PORTS) ..... 3431
21.4.8 BIST Active FIS Register (BISTAFR) ..... 3432
21.4.9 BIST Control Register (BISTCR) ..... 3432
21.4.10 BIST FIS Count Register (BISTFCTR) ..... 3435
21.4.11 BIST Status Register (BISTSR) ..... 3435
21.4.12 BIST DWORD Error Count Register (BISTDECR) ..... 3436
21.4.13 BIST DWORD Error Count Register (TIMER1MS) ..... 3436
21.4.14 Global Parameter 1 Register (GPARAM1R) ..... 3437
21.4.15 Global Parameter 2 Register (GPARAM2R) ..... 3438
21.4.16 Port Parameter Register (PPARAMR) ..... 3439
21.4.17 Version Register (VERSIONR) ..... 3440
21.4.18 ID Register (IDR) ..... 3440
21.4.19 Port Command List Base Address Register (P0CLB) ..... 3441
21.4.20 Port FIS Base Address Register (P0FB) ..... 3441
21.4.21 Port Interrupt Status Register (P0IS) ..... 3442
21.4.22 Port Interrupt Enable Register (P0IE) ..... 3444
21.4.23 Port Command Register (P0CMD) ..... 3445
21.4.24 Port Task File Data Register (P0TFD) ..... 3447
21.4.25 Port Signature Register (P0SIG) ..... 3448
21.4.26 Port Serial ATA Status Register (P0SSTS) ..... 3448
21.4.27 Port Serial ATA Control Register (P0SCTL) ..... 3450
21.4.28 Port Serial ATA Error Register (P0SERR) ..... 3451
21.4.29 Port Serial ATA Active Register (P0SACT) ..... 3453
21.4.30 Port Command Issue Register (P0CI) ..... 3453
21.4.31 Port Serial ATA Notification Register (P0SNTF) ..... 3454
21.4.32 Port DMA Control Register (P0DMACR) ..... 3455
21.4.33 Idle Register (IDLE) ..... 3457
21.4.34 PHY Configuration Receive 0 Register (CFGRX0) ..... 3458
21.4.35 PHY Configuration Receive 1 Register (CFGRX1) ..... 3460
21.4.36 PHY Configuration Receive 2 Register (CFGRX2) ..... 3461
21.4.37 PHY Configuration Receive 3 Register (CFGRX3) ..... 3462
21.4.38 PHY Configuration Receive 4 Register (CFGRX4) ..... 3464
21.4.39 Receive Bus PHY-to-Controller Status Register (Phy STSRX) ..... 3465
21.4.40 PHY Configuration Transmit 0 Register (CFGTX0) ..... 3466
21.4.41 PHY Configuration Transmit 1 Register (CFGTX1) ..... 3469
21.4.42 PHY Configuration Transmit 2 Register (CFGTX2) ..... 3471
21.4.43 PHY Configuration Transmit 3 Register (CFGTX3) ..... 3473
21.4.44 PHY Configuration Transmit 4 Register (CFGTX4) ..... 3474
21.4.45 Transmit Bus Controller-to-PHY Status Register ..... 3474
21.5 SATA PLL Registers ..... 3475
21.5.1 SATA PLL Configuration 0 Register (SATA_PLLCFG0) ..... 3475
21.5.2 SATA PLL Configuration 1 Register (SATA_PLLCFG1) ..... 3477


Copyright © 2011–2016, Texas Instruments Incorporated

21.5.3 SATA PLL Configuration 2 Register (SATA_PLLCFG2) ... 3478
21.5.4 SATA PLL Configuration 3 Register (SATA_PLLCFG3) ... 3479
21.5.5 SATA PLL Configuration 4 Register (SATA_PLLCFG4) ... 3480
21.5.6 SATA PLL Status (SATA_PLLSTATUS) ... 3482
21.5.7 SATA Receive Status (SATA_RXSTATUS) ... 3482
21.5.8 SATA Transmit Status (SATA_TXSTATUS) ... 3483
21.5.9 SATA TEST Configuration (SATA_TESTCFG) ... 3483
21.6 SATA Subsystem Registers ... 3484
21.6.1 Clock Domain Power Control Register (CM_DEFAULT_L3_MED_CLKSTCTRL) ... 3484
21.6.2 Clock Domain SATA Control Register (CM_DEFAULT_SATA_CLKCTRL) ... 3485

**22 Serial Port Interface (SPI) ... 3486**

22.1 Introduction ... 3487
22.1.1 Overview ... 3487
22.1.2 Features ... 3487
22.2 Architecture ... 3488
22.2.1 SPI Interface ... 3488
22.2.2 SPI Transmission ... 3488
22.2.3 Master Mode ... 3495
22.2.4 Slave Mode ... 3512
22.2.5 Interrupts ... 3516
22.2.6 DMA Requests ... 3517
22.2.7 Emulation Mode ... 3518
22.2.8 Power Saving Management ... 3519
22.2.9 System Test Mode ... 3520
22.2.10 Reset ... 3521
22.2.11 Access to Data Registers ... 3521
22.2.12 Programming Aid ... 3522
22.2.13 Interrupt and DMA Events ... 3525
22.3 SPI Registers ... 3526
22.3.1 McSPI IP Revision Register (MCSPI_HL_REV) ... 3527
22.3.2 McSPI IP Hardware Information Register (MCSPI_HL_HWINFO) ... 3528
22.3.3 McSPI IP System Configuration Register (MCSPI_HL_SYSCONFIG) ... 3529
22.3.4 McSPI Revision Register (MCSPI_REVISION) ... 3530
22.3.5 McSPI System Configuration Register (MCSPI_SYSCONFIG) ... 3531
22.3.6 McSPI System Status Register (MCSPI_SYSSTATUS) ... 3532
22.3.7 McSPI Interrupt Status Register (MCSPI_IRQSTATUS) ... 3533
22.3.8 McSPI Interrupt Enable Register (MCSPI_IRQENABLE) ... 3536
22.3.9 McSPI Wakeup Enable Register (MCSPI_WAKEUPENABLE) ... 3538
22.3.10 McSPI System Test Register (MCSPI_SYST) ... 3539
22.3.11 McSPI Module Control Register (MCSPI_MODULCTRL) ... 3541
22.3.12 McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) ... 3543
22.3.13 McSPI Channel *(i)* Status Register (MCSPI_CH(i)STAT) ... 3547
22.3.14 McSPI Channel *(i)* Control Register (MCSPI_CH(I)CTRL) ... 3548
22.3.15 McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*) ... 3549
22.3.16 McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*) ... 3549
22.3.17 McSPI Transfer Levels Register (MCSPI_XFERLEVEL) ... 3550
22.3.18 DMA Address Aligned FIFO Transmitter Register (MCSPI_DAFTX) ... 3551
22.3.19 DMA Address Aligned FIFO Receiver Register (MCSPI_DAFRX) ... 3551

**23 Timers ... 3552**

23.1 Introduction ... 3553
23.1.1 Overview ... 3553
23.1.2 Features ... 3553
23.1.3 Functional Block Diagram ... 3554



Copyright © 2011–2016, Texas Instruments Incorporated

23.1.4 GP Timer External System Interface ..... 3555
23.2 Architecture ..... 3556
23.2.1 Functional Description ..... 3556
23.2.2 Accessing Registers ..... 3562
23.2.3 Posted Mode Selection ..... 3562
23.2.4 Write Registers Access ..... 3563
23.2.5 Read Registers Access ..... 3564
23.2.6 TIDR Register ..... 3566
23.2.7 TIOCP_CFG Register ..... 3567
23.2.8 IRQ_EOI Register ..... 3568
23.2.9 IRQSTATUS_RAW Register ..... 3569
23.2.10 IRQSTATUS Register ..... 3570
23.2.11 IRQENABLE_SET Register ..... 3571
23.2.12 IRQENABLE_CLR Register ..... 3572
23.2.13 IRQWAKEEN Register ..... 3573
23.2.14 TCLR Register ..... 3573
23.2.15 TCRR Register ..... 3575
23.2.16 TLDR Register ..... 3575
23.2.17 TTGR Register ..... 3575
23.2.18 TWPS Register ..... 3576
23.2.19 TMAR Register ..... 3577
23.2.20 TCAR1 Register ..... 3577
23.2.21 TSICR Register ..... 3578
23.2.22 TCAR2 Register ..... 3578

**24 UART/IrDA/CIR Module ..... 3579**

24.1 Introduction ..... 3580
24.1.1 Main Features ..... 3580
24.1.2 UART/Modem Functions ..... 3581
24.1.3 IrDA Functions ..... 3581
24.1.4 CIR Features ..... 3581
24.2 Architecture ..... 3582
24.2.1 UART Signal Descriptions ..... 3582
24.2.2 UART/IrDA/CIR Mode Selection ..... 3582
24.2.3 UART Mode ..... 3583
24.2.4 IrDA Mode ..... 3587
24.2.5 CIR Mode ..... 3596
24.2.6 FIFO Management ..... 3601
24.2.7 Interrupts ..... 3608
24.2.8 Sleep Modes ..... 3610
24.2.9 Idle Modes ..... 3610
24.2.10 Programmable Baud Rate Generator ..... 3611
24.3 UART Registers ..... 3614
24.3.1 Receiver Holding Register (RHR) ..... 3615
24.3.2 Transmit Holding Register (THR) ..... 3615
24.3.3 Interrupt Enable Register (IER) - UART Mode ..... 3616
24.3.4 Interrupt Enable Register (IER) - IrDA Mode ..... 3617
24.3.5 Interrupt Enable Register (IER) - CIR Mode ..... 3618
24.3.6 Interrupt Identification Register (IIR) - UART Mode ..... 3619
24.3.7 Interrupt Identification Register (IIR) - IrDA Mode ..... 3620
24.3.8 Interrupt Identification Register (IIR) - CIR Mode ..... 3621
24.3.9 FIFO Control Register (FCR) ..... 3622
24.3.10 Line Control Register (LCR) ..... 3623
24.3.11 Modem Control Register (MCR) ..... 3624


Copyright © 2011–2016, Texas Instruments Incorporated

24.3.12 Line Status Register (LSR) - UART Mode ... 3625
24.3.13 Line Status Register (LSR) - IrDA Mode ... 3626
24.3.14 Line Status Register (LSR) - CIR Mode ... 3627
24.3.15 Modem Status Register (MSR) ... 3628
24.3.16 Transmission Control Register (TCR) ... 3629
24.3.17 Scratchpad Register (SPR) ... 3629
24.3.18 Trigger Level Register (TLR) ... 3630
24.3.19 Mode Definition Register 1 (MDR1) ... 3631
24.3.20 Mode Definition Register 2 (MDR2) ... 3632
24.3.21 Mode Definition Register 3 (MDR3) ... 3633
24.3.22 Status FIFO Line Status Register (SFLSR) ... 3634
24.3.23 RESUME Register ... 3634
24.3.24 Status FIFO Register Low (SFREGL) ... 3635
24.3.25 Status FIFO Register High (SFREGH) ... 3635
24.3.26 BOF Control Register (BLR) ... 3636
24.3.27 Auxiliary Control Register (ACREG) ... 3637
24.3.28 Supplementary Control Register (SCR) ... 3638
24.3.29 Supplementary Status Register (SSR) ... 3639
24.3.30 BOF Length Register (EBLR) ... 3640
24.3.31 Module Version Register (MVR) ... 3641
24.3.32 System Configuration Register (SYSC) ... 3642
24.3.33 System Status Register (SYSS) ... 3642
24.3.34 Wake-Up Enable Register (WER) ... 3643
24.3.35 Carrier Frequency Prescaler Register (CFPS) ... 3644
24.3.36 Divisor Latches Low Register (DLL) ... 3645
24.3.37 Divisor Latches High Register (DLH) ... 3645
24.3.38 Enhanced Feature Register (EFR) ... 3646
24.3.39 XON1/ADDR1 Register ... 3647
24.3.40 XON2/ADDR2 Register ... 3647
24.3.41 XOFF1 Register ... 3648
24.3.42 XOFF2 Register ... 3648
24.3.43 Transmit Frame Length Low Register (TXFLL) ... 3649
24.3.44 Transmit Frame Length High Register (TXFLH) ... 3649
24.3.45 Received Frame Length Low Register (RXFLL) ... 3650
24.3.46 Received Frame Length High Register (RXFLH) ... 3650
24.3.47 UART Autobauding Status Register (UASR) ... 3651

**25 Universal Serial Bus Subsystem (USBSS) ... 3652**

25.1 Introduction ... 3653
25.2 Features Supported ... 3653
25.3 Functional Block Diagram ... 3654
25.3.1 L3 Slow Interconnect ... 3654
25.3.2 Memory Mapped Registers ... 3654
25.3.3 USB 2.0 Controller ... 3654
25.3.4 USB PHY ... 3654
25.3.5 USB Pins ... 3655
25.3.6 DMA ... 3655
25.4 Industry Standard(s) Compliance ... 3655
25.5 Architecture ... 3656
25.5.1 USB Controller Operational Mode ... 3656
25.5.2 Clock Control ... 3661
25.5.3 Signal Descriptions ... 3661
25.5.4 VBUS Voltage Sourcing Control ... 3661
25.5.5 D+/D- Pull-up/Pull-Down Resistors ... 3661


Copyright © 2011–2016, Texas Instruments Incorporated

25.5.6 Interrupts ... 3662
25.5.7 Subsystem Reset Considerations ... 3665
25.5.8 Clock, PLL, and PHY Initialization ... 3665
25.6 Error Handling ... 3666
25.6.1 Error Handling as a USB Device ... 3666
25.6.2 Babble Event Detection as a USB Host ... 3666
25.7 Protocol Description(s) ... 3667
25.7.1 USB Device Operation ... 3667
25.7.2 USB Host Operation ... 3683
25.8 DMA ... 3699
25.8.1 CPPI Terminology ... 3699
25.8.2 Data Structures ... 3701
25.8.3 Queue Manager ... 3710
25.8.4 Memory Regions and Linking RAM ... 3714
25.8.5 Zero Length Packets ... 3715
25.8.6 CPPI DMA Scheduler ... 3715
25.8.7 CPPI DMA State Registers ... 3717
25.8.8 CPPI DMA Protocols Supported ... 3717
25.8.9 USB Data Flow Using DMA ... 3720
25.9 Registers ... 3726
25.9.1 USBSS Registers ... 3726
25.9.2 USB0 and USB1 Controller Registers ... 3759
25.9.3 CPPI DMA Controller Registers ... 3827
25.9.4 CPPI DMA Scheduler Registers ... 3834
25.9.5 CPPI DMA Queue Manager Registers ... 3837
25.9.6 USB Mentor Core Registers ... 3852
25.9.7 FIFOs ... 3878

**26 Watchdog Timer ... 3888**

26.1 Introduction ... 3889
26.1.1 Overview ... 3889
26.1.2 Functional Block Diagram ... 3889
26.1.3 Features ... 3889
26.1.4 Watchdog Timer Environment ... 3889
26.2 Architecture ... 3890
26.2.1 Power Management ... 3890
26.2.2 Interrupts ... 3890
26.2.3 General Watchdog Timer Operation ... 3890
26.2.4 Reset Context ... 3891
26.2.5 Overflow/Reset Generation ... 3891
26.2.6 Prescaler Value/Timer Reset Frequency ... 3891
26.2.7 Triggering a Timer Reload ... 3893
26.2.8 Start/Stop Sequence for Watchdog Timers (Using the WDT_WSPR Register) ... 3893
26.2.9 Modifying Timer Count/Load Values and Prescaler Setting ... 3893
26.2.10 Watchdog Counter Register Access Restriction (WDT_WCRR Register) ... 3893
26.2.11 Watchdog Timer Interrupt Generation ... 3894
26.2.12 Watchdog Timers Under Emulation ... 3895
26.2.13 Accessing Watchdog Timer Registers ... 3895
26.3 Low-Level Programming Model ... 3896
26.3.1 Global Initialization ... 3896
26.3.2 Operational Mode Configuration ... 3896
26.4 Watchdog Timer Registers ... 3898
26.4.1 WDT_WIDR Register ... 3899
26.4.2 WDT_WDSC Register ... 3899


Copyright © 2011–2016, Texas Instruments Incorporated

26.4.3 WDT_WDST Register ..... 3900
26.4.4 WDT_WISR Register ..... 3900
26.4.5 WDT_WIER Register ..... 3901
26.4.6 WDT_WCLR Register ..... 3901
26.4.7 WDT_WCRR Register ..... 3902
26.4.8 WDT_WLDR Register ..... 3902
26.4.9 WDT_WTGR Register ..... 3902
26.4.10 WDT_WWPS Register ..... 3903
26.4.11 WDT_WDLY Register ..... 3904
26.4.12 WDT_WSPR Register ..... 3904
26.4.13 WDT_WIRQSTATRAW Register ..... 3905
26.4.14 WDT_WIRQSTAT Register ..... 3906
26.4.15 WDT_WIRQENSET Register ..... 3907
26.4.16 WDT_WIRQENCLR Register ..... 3908
**Revision History** ..... **3909**


Copyright © 2011–2016, Texas Instruments Incorporated

# List of Figures


1-1. ARM Cortex-A8 Subsystem ..... 156
1-2. Dual Cortex-M3 MPU Subsystem Overview ..... 158
1-3. Dual Cortex-M3 MPU Subsystem Integration Overview ..... 160
1-4. Dual Cortex-M3 MPU Subsystem Clocking Scheme ..... 161
1-5. Dual Cortex-M3 MPU Subsystem Reset Scheme ..... 162
1-6. Dual Cortex-M3 MPU Subsystem ..... 163
1-7. TMS320C674x Megamodule Block Diagram ..... 180
1-8. HDVICP2 Block Diagram ..... 184
1-9. MMU Intergration ..... 189
1-10. MMU Block Diagram ..... 190
1-11. MMU Address Translation Process ..... 191
1-12. Translation Hierarchy ..... 192
1-13. First-Level Descriptor Address Calculation ..... 192
1-14. Detailed First-Level Descriptor Address Calculation ..... 193
1-15. Section Translation Summary ..... 194
1-16. Supersection Translation Summary ..... 195
1-17. Two-Level Translation ..... 196
1-18. Small Page Translation Summary ..... 197
1-19. Large Page Translation Summary ..... 198
1-20. TLB-Entry Lock Mechanism ..... 199
1-21. TLB-Entry Structure ..... 200
1-22. MMU Global Initialization ..... 202
1-23. MMU_REVISION ..... 205
1-24. MMU_SYSCONFIG ..... 206
1-25. MMU_SYSSTATUS ..... 207
1-26. MMU_IRQSTATUS ..... 208
1-27. MMU_IRQENABLE ..... 209
1-28. MMU_WALKING_ST ..... 210
1-29. MMU_CNTL ..... 210
1-30. MMU_FAULT_AD ..... 211
1-31. MMU_TTB ..... 211
1-32. MMU_LOCK ..... 211
1-33. MMU_LD_TLB ..... 212
1-34. MMU_CAM ..... 212
1-35. MMU_RAM ..... 213
1-36. MMU_GFLUSH ..... 213
1-37. MMU_FLUSH_ENTRY ..... 214
1-38. MMU_READ_CAM ..... 214
1-39. MMU_READ_RAM ..... 215
1-40. MMU_EMU_FAULT_AD ..... 216
1-41. MMU_FAULT_PC ..... 217
1-42. Graphics Accelerator Highlight ..... 218
1-43. SGX Subsystem Integration ..... 221
1-44. SGX Block Diagram ..... 223
1-45. OCP Revision Register (OCP_REVISION) ..... 226
1-46. Hardware Implementation Information Register (OCP_HWINFO) ..... 226
1-47. System Configuration Register (OCP_SYSCONFIG) ..... 227



Copyright © 2011–2016, Texas Instruments Incorporated

1-48. Raw IRQ 0 Status Register (OCP_IRQSTATUS_RAW_0) ..... 228
1-49. Raw IRQ 1 Status Register (OCP_IRQSTATUS_RAW_1) ..... 228
1-50. Raw IRQ 2 Status Register (OCP_IRQSTATUS_RAW_2) ..... 229
1-51. Interrupt 0 Status Event Register (OCP_IRQSTATUS_0) ..... 229
1-52. Interrupt 1 Status Event Register (OCP_IRQSTATUS_1) ..... 230
1-53. Interrupt 2 Status Event Register (OCP_IRQSTATUS_2) ..... 230
1-54. Enable Interrupt 0 Register (OCP_IRQENABLE_SET_0) ..... 231
1-55. Enable Interrupt 1 Register (OCP_IRQENABLE_SET_1) ..... 231
1-56. Enable Interrupt 2 Register (OCP_IRQENABLE_SET_2) ..... 232
1-57. Disable Interrupt 0 Register (OCP_IRQENABLE_CLR_0) ..... 232
1-58. Disable Interrupt 1 Register (OCP_IRQENABLE_CLR_1) ..... 233
1-59. Disable Interrupt 2 Register (OCP_IRQENABLE_CLR_2) ..... 233
1-60. Configure Memory Page Register (OCP_PAGE_CONFIG) ..... 234
1-61. Interrupt Events Register (OCP_INTERRUPT_EVENT) ..... 235
1-62. Configuration of Debug Modes Register (OCP_DEBUG_CONFIG) ..... 237
1-63. Debug Status Register (OCP_DEBUG_STATUS) ..... 238
1-64. Device Interrupt Controllers ..... 241
1-65. EDMA Channel Synchronization Events ..... 252
1-66. IPC Overview Diagram ..... 256
1-67. System IPCs ..... 257
1-68. HDVICP2 IPC ..... 258
1-69. System Mailbox Integration ..... 261
1-70. Mailbox Block Diagram ..... 263
1-71. Revision Register (MAILBOX_REVISION) ..... 271
1-72. System Configuration Register (MAILBOX_SYSCONFIG) ..... 271
1-73. Message Register (MAILBOX_MESSAGE_m) ..... 272
1-74. FIFO Status Register (MAILBOX_FIFOSTATUS_m) ..... 272
1-75. Message Status Register (MAILBOX_MSGSTATUS_m) ..... 273
1-76. IRQ RAW Status Register (MAILBOX_IRQSTATUS_RAW_u) ..... 274
1-77. IRQ Clear Status Register (MAILBOX_IRQSTATUS_CLR_u) ..... 278
1-78. IRQ Enable Set Register (MAILBOX_IRQENABLE_SET_u) ..... 282
1-79. IRQ Enable Clear Register (MAILBOX_IRQENABLE_CLR_u) ..... 286
1-80. Spinlock Module ..... 290
1-81. Spinlock Integration ..... 291
1-82. SPINLOCK_LOCK_REG_i Register State Diagram ..... 293
1-83. Take and Release Spinlock ..... 295
1-84. Revision Register (SPINLOCK_REV) ..... 296
1-85. System Configuration Register (SPINLOCK_SYS_CFG) ..... 297
1-86. System Status Register (SPINLOCK_SYSSTAT) ..... 298
1-87. Lock Register (SPINLOCK_LOCK_REG_i) ..... 299
1-88. ELM Integration ..... 301
1-89. ELM Revision Register (ELM_REVISION) ..... 313
1-90. ELM System Configuration Register (ELM_SYSCONFIG) ..... 313
1-91. ELM System Status Register (ELM_SYSSTATUS) ..... 314
1-92. ELM Interrupt Status Register (ELM_IRQSTATUS) ..... 315
1-93. ELM Interrupt Enable Register (ELM_IRQENABLE) ..... 317
1-94. ELM Location Configuration Register (ELM_LOCATION_CONFIG) ..... 318
1-95. ELM Page Definition Register (ELM_PAGE_CTRL) ..... 319
1-96. ELM_SYNDROME_FRAGMENT_0_i Register ..... 320


Copyright © 2011–2016, Texas Instruments Incorporated

1-97. ELM_SYNDROME_FRAGMENT_1_i Register ..... 320
1-98. ELM_SYNDROME_FRAGMENT_2_i Register ..... 320
1-99. ELM_SYNDROME_FRAGMENT_3_i Register ..... 321
1-100. ELM_SYNDROME_FRAGMENT_4_i Register ..... 321
1-101. ELM_SYNDROME_FRAGMENT_5_i Register ..... 322
1-102. ELM_SYNDROME_FRAGMENT_6_i Register ..... 322
1-103. ELM_LOCATION_STATUS_i Register ..... 323
1-104. ELM_ERROR_LOCATION_0-15_i Registers ..... 324
1-105. Interconnect Overview ..... 326
1-106. L3 Interconnect Overview ..... 329
1-107. L3 Connectivity Matrix ..... 335
1-108. L3 Global Error-Routing Scheme ..... 339
1-109. Typical Error Analysis Sequence ..... 344
1-110. L4 Interconnect Overview ..... 408
1-111. L4 Initiator-Target Connectivity ..... 413
1-112. Example of CONNID_BIT_VECTOR L4_AP_PROT_GROUP_MEMBERS_k ..... 415
1-113. L4 Segmentation ..... 417
1-114. L4 Error Reporting ..... 425
1-115. Typical Error Analysis Sequence ..... 427
1-116. Typical Error Analysis Sequence ..... 428
2-1. Functional and Interface Clocks ..... 464
2-2. Generic Clock Domain ..... 467
2-3. Clock Domain State Transitions ..... 468
2-4. Power Domain Block Diagram ..... 469
2-5. Comparison of Energy Consumed With/Without DVFS ..... 471
2-6. Clock Structure ..... 475
2-7. Main PLL Clock Structure ..... 477
2-8. USB PLL Clock Structure ..... 479
2-9. : DDR PLL Clock Structure ..... 480
2-10. SERDES and Ethernet Clock Structure ..... 481
2-11. Video Clock Structure ..... 482
2-12. Audio Clock Structure ..... 484
2-13. Timer Clock Structure ..... 487
2-14. Watchdog and Sync Timer Clock Structure ..... 487
2-15. Device Clocking External Clock Sources ..... 489
2-16. Clock Output Pin Options ..... 489
2-17. Oscillator Connection ..... 490
2-18. DPLLS Clocks Structure ..... 491
2-19. DPLLJ Basic Structure ..... 494
2-20. MPU SS POR Sequence ..... 506
2-21. MPU Warm Reset Sequence ..... 507
2-22. POR Sequence of the HDVICP2 ..... 508
2-23. Software Warm Reset Sequence of the HDVICP ..... 509
2-24. Media Controller Power-On Reset Sequence ..... 510
2-25. Media Controller MPU Subsystem Software Warm Reset Sequence ..... 512
2-26. Generic Voltage Domain ..... 513
2-27. CONTROL_REVISION Register ..... 526
2-28. CONTROL_HWINFO Register ..... 527
2-29. CONTROL_SYSCONFIG Register ..... 528


Copyright © 2011–2016, Texas Instruments Incorporated

2-30. PLLSS_MMR_LOCK Register ..... 529
2-31. MPUPLL_PWRCTRL Register ..... 530
2-32. MPUPLL_CLKCTRL Register ..... 531
2-33. MPUPLL_TENABLE Register ..... 533
2-34. MPUPLL_TENABLEDIV Register ..... 534
2-35. MPUPLL_M2NDIV Register ..... 535
2-36. MPUPLL_MN2DIV Register ..... 536
2-37. MPUPLL_FRACDIV Register ..... 537
2-38. MPUPLL_BWCTRL Register ..... 538
2-39. MPUPLL_FRACCTRL Register ..... 539
2-40. MPUPLL_STATUS Register ..... 540
2-41. MPUPLL_M3DIV Register ..... 542
2-42. MPUPLL_RAMPCTRL Register ..... 543
2-43. DSPPLL_PWRCTRL Register ..... 544
2-44. DSPPLL_CLKCTRL Register ..... 545
2-45. DSPPLL_TENABLE Register ..... 547
2-46. DSPPLL_TENABLEDIV Register ..... 548
2-47. DSPPLL_M2NDIV Register ..... 549
2-48. DSPPLL_MN2DIV Register ..... 550
2-49. DSPPLL_FRACDIV Register ..... 551
2-50. DSPPLL_BWCTRL Register ..... 552
2-51. DSPPLL_FRACCTRL Register ..... 553
2-52. DSPPLL_STATUS Register ..... 554
2-53. SGXPLLL_PWRCTRL Register ..... 556
2-54. SGXPLL_CLKCTRL Register ..... 557
2-55. SGXPLLL_TENABLE Register ..... 559
2-56. SGXPLL_TENABLEDIV Register ..... 560
2-57. SGXPLL_M2NDIV Register ..... 561
2-58. SGXPLL_MN2DIV Register ..... 562
2-59. SGXPLL_FRACDIV Register ..... 563
2-60. SGXPLL_BWCTRL Register ..... 564
2-61. SGXPLL_FRACCTRL Register ..... 565
2-62. SGXPLL_STATUS Register ..... 566
2-63. HDVICPPLL_PWRCTRL Register ..... 568
2-64. HDVICPPLL_CLKCTRL Register ..... 569
2-65. HDVICPPLL_TENABLE Register ..... 571
2-66. HDVICPPLL_TENABLEDIV Register ..... 572
2-67. HDVICPPLL_M2NDIV Register ..... 573
2-68. HDVICPPLL_MN2DIV Register ..... 574
2-69. HDVICPPLL_FRACDIV Register ..... 575
2-70. HDVICPPLL_BWCTRL Register ..... 576
2-71. HDVICPPLL_FRACCTRL Register ..... 577
2-72. HDVICPPLL_STATUS Register ..... 578
2-73. L3PLL_PWRCTRL Register ..... 580
2-74. L3PLL_CLKCTRL Register ..... 581
2-75. L3PLL_TENABLE Register ..... 583
2-76. L3PLL_TENABLEDIV Register ..... 584
2-77. L3PLL_M2NDIV Register ..... 585
2-78. L3PLL_MN2DIV Register ..... 586


Copyright © 2011–2016, Texas Instruments Incorporated

2-79. L3PLL_FRACDIV Register ..... 587
2-80. L3PLL_BWCTRL Register ..... 588
2-81. L3PLL_FRACCTRL Register ..... 589
2-82. L3PLL_STATUS Register ..... 590
2-83. ISPPLL_PWRCTRL Register ..... 592
2-84. ISPPLL_CLKCTRL Register ..... 593
2-85. ISPPLL_TENABLE Register ..... 595
2-86. ISPPLL_TENABLEDIV Register ..... 596
2-87. ISPPLL_M2NDIV Register ..... 597
2-88. ISPPLL_MN2DIV Register ..... 598
2-89. ISPPLL_FRACDIV Register ..... 599
2-90. ISPPLL_BWCTRL Register ..... 600
2-91. ISPPLL_FRACCTRL Register ..... 601
2-92. ISPPLL_STATUS Register ..... 602
2-93. DSSPLL_PWRCTRL Register ..... 604
2-94. DSSPLL_CLKCTRL Register ..... 605
2-95. DSSPLL_TENABLE Register ..... 607
2-96. DSSPLL_TENABLEDIV Register ..... 608
2-97. DSSPLL_M2NDIV Register ..... 609
2-98. DSSPLL_MN2DIV Register ..... 610
2-99. DSSPLL_FRACDIV Register ..... 611
2-100. DSSPLL_BWCTRL Register ..... 612
2-101. DSSPLL_FRACCTRL Register ..... 613
2-102. DSSPLL_STATUS Register ..... 614
2-103. VIDEO0PLL_PWRCTRL Register ..... 616
2-104. VIDEO0PLL_CLKCTRL Register ..... 617
2-105. VIDEO0PLL_TENABLE Register ..... 619
2-106. VIDEO0PLL_TENABLEDIV Register ..... 620
2-107. VIDEO0PLL_M2NDIV Register ..... 621
2-108. VIDEO0PLL_MN2DIV Register ..... 622
2-109. VIDEO0PLL_FRACDIV Register ..... 623
2-110. VIDEO0PLL_BWCTRL Register ..... 624
2-111. VIDEO0PLL_FRACCTRL Register ..... 625
2-112. VIDEO0PLL_STATUS Register ..... 626
2-113. VIDEO1PLL_PWRCTRL Register ..... 628
2-114. VIDEO1PLL_CLKCTRL Register ..... 629
2-115. VIDEO1PLL_TENABLE Register ..... 631
2-116. VIDEO1PLL_TENABLEDIV Register ..... 632
2-117. VIDEO1PLL_M2NDIV Register ..... 633
2-118. VIDEO1PLL_MN2DIV Register ..... 634
2-119. VIDEO1PLL_FRACDIV Register ..... 635
2-120. VIDEO1PLL_BWCTRL Register ..... 636
2-121. VIDEO1PLL_FRACCTRL Register ..... 637
2-122. VIDEO1PLL_STATUS Register ..... 638
2-123. HDMIPLL_PWRCTRL Register ..... 640
2-124. HDMIPLL_CLKCTRL Register ..... 641
2-125. HDMIPLL_TENABLE Register ..... 643
2-126. HDMIPLL_TENABLEDIV Register ..... 644
2-127. HDMIPLL_M2NDIV Register ..... 645


Copyright © 2011–2016, Texas Instruments Incorporated

2-128. HDMIPLL_MN2DIV Register ..... 646
2-129. HDMIPLL_FRACDIV Register ..... 647
2-130. HDMIPLL_BWCTRL Register ..... 648
2-131. HDMIPLL_FRACCTRL Register ..... 649
2-132. HDMIPLL_STATUS Register ..... 650
2-133. AUDIOPLL_PWRCTRL Register ..... 652
2-134. AUDIOPLL_CLKCTRL Register ..... 653
2-135. AUDIOPLL_TENABLE Register ..... 655
2-136. AUDIOPLL_TENABLEDIV Register ..... 656
2-137. AUDIOPLL_M2NDIV Register ..... 657
2-138. AUDIOPLL_MN2DIV Register ..... 658
2-139. AUDIOPLL_FRACDIV Register ..... 659
2-140. AUDIOPLL_BWCTRL Register ..... 660
2-141. AUDIOPLL_FRACCTRL Register ..... 661
2-142. AUDIOPLL_STATUS Register ..... 662
2-143. USBPLL_PWRCTRL Register ..... 664
2-144. USBPLL_CLKCTRL Register ..... 665
2-145. USBPLL_TENABLE Register ..... 667
2-146. USBPLL_TENABLEDIV Register ..... 668
2-147. USBPLL_M2NDIV Register ..... 669
2-148. USBPLL_MN2DIV Register ..... 670
2-149. USBPLL_FRACDIV Register ..... 671
2-150. USBPLL_BWCTRL Register ..... 672
2-151. USBPLL_FRACCTRL Register ..... 673
2-152. USBPLL_STATUS Register ..... 674
2-153. DDRPLL_PWRCTRL Register ..... 676
2-154. DDRPLL_CLKCTRL Register ..... 677
2-155. DDRPLL_TENABLE Register ..... 679
2-156. DDRPLL_TENABLEDIV Register ..... 680
2-157. DDRPLL_M2NDIV Register ..... 681
2-158. DDRPLL_MN2DIV Register ..... 682
2-159. DDRPLL_FRACDIV Register ..... 683
2-160. DDRPLL_BWCTRL Register ..... 684
2-161. DDRPLL_FRACCTRL Register ..... 685
2-162. DDRPLL_STATUS Register ..... 686
2-163. OSC_SRC Register ..... 688
2-164. MPU_CLKSRC Register ..... 690
2-165. VIDEO_PLL_CLKSRC Register ..... 691
2-166. MLB_ATL_CLKSRC Register ..... 692
2-167. McASP345_AUX_CLKSRC Register ..... 693
2-168. McASP_AHCLK_CLKSRC Register ..... 694
2-169. McBSP_UART_CLKSRC Register ..... 696
2-170. HDMI_I2S_CLKSRC Register ..... 697
2-171. DMTIMER_CLKSRC Register ..... 698
2-172. CLKOUT_MUX Register ..... 700
2-173. RMII_REFCLK_SRC Register ..... 701
2-174. SYSCLK18_CLKSRC Register ..... 702
2-175. WDT0_CLKSRC Register ..... 703
2-176. DMTIMER_CLK_CHANGE Register ..... 704


Copyright © 2011–2016, Texas Instruments Incorporated

2-177. DEEPSLEEP_CTRL Register ..... 706
2-178. DEEPSLEEP_STATUS Register ..... 707
2-179. PRM_RSTCTRL Register ..... 708
2-180. PRM_RSTTIME Register ..... 709
2-181. PRM_RSTST Register ..... 710
2-182. CM_CLKOUT_CTRL Register ..... 712
2-183. REVISION_PRM Register ..... 713
2-184. CM_SYSCLK3_CLKSEL Register ..... 715
2-185. CM_SYSCLK10_CLKSEL Register ..... 716
2-186. CM_VPB3_CLKSEL Register ..... 717
2-187. CM_VPC1_CLKSEL Register ..... 718
2-188. CM_VPD1_CLKSEL Register ..... 719
2-189. CM_SYSCLK19_CLKSEL Register ..... 720
2-190. CM_SYSCLK20_CLKSEL Register ..... 721
2-191. CM_SYSCLK21_CLKSEL Register ..... 722
2-192. CM_SYSCLK22_CLKSEL Register ..... 723
2-193. CM_APA_CLKSEL Register ..... 724
2-194. CM_SYSCLK14_CLKSEL Register ..... 725
2-195. CM_SYSCLK16_CLKSEL Register ..... 726
2-196. CM_SYSCLK18_CLKSEL Register ..... 727
2-197. CM_AUDIOCLK_MCASP0_CLKSEL Register ..... 728
2-198. CM_AUDIOCLK_MCASP1_CLKSEL Register ..... 729
2-199. CM_AUDIOCLK_MCASP2_CLKSEL Register ..... 730
2-200. CM_AUDIOCLK_MCBSP_CLKSEL Register ..... 731
2-201. CM_HDMI_CLKSEL Register ..... 732
2-202. CM_SYSCLK23_CLKSEL Register ..... 733
2-203. CM_GEM_CLKSTCTRL Register ..... 734
2-204. CM_ACTIVE_GEM_CLKCTRL Register ..... 735
2-205. CM_DEFAULT_PCI_CLKSTCTRL Register ..... 737
2-206. CM_DEFAULT_DUCATI_CLKSTCTRL Register ..... 738
2-207. CM_DEFAULT_DMM_CLKCTRL Register ..... 739
2-208. CM_DEFAULT_USB_CLKCTRL Register ..... 740
2-209. CM_DEFAULT_SATA_CLKCTRL Register ..... 741
2-210. CM_DEFAULT_DUCATI_CLKCTRL Register ..... 742
2-211. CM_DEFAULT_PCI_CLKCTRL Register ..... 743
2-212. CM_HDVICP_CLKSTCTRL Register ..... 745
2-213. CM_HDVICP_CLKCTRL Register ..... 746
2-214. CM_HDVICP_SL2_CLKCTRL Register ..... 747
2-215. CM_ISP_CLKSTCTRL Register ..... 749
2-216. CM_ISP_ISP_CLKCTRL Register ..... 750
2-217. CM_ISP_FDIF_CLKCTRL Register ..... 751
2-218. CM_DSS_CLKSTCTRL Register ..... 752
2-219. CM_DSS_DSS_CLKCTRL Register ..... 753
2-220. CM_DSS_HDMI_CLKCTRL Register ..... 754
2-221. CM_SGX_CLKSTCTRL Register ..... 755
2-222. CM_SGX_SGX_CLKCTRL Register ..... 756
2-223. PM_ACTIVE_PWRSTCTRL Register ..... 758
2-224. PM_ACTIVE_PWRSTST Register ..... 759
2-225. RM_ACTIVE_RSTCTRL Register ..... 760


Copyright © 2011–2016, Texas Instruments Incorporated

2-226. RM_ACTIVE_RSTST Register ..... 761
2-227. PM_DEFAULT_PWRSTCTRL Register ..... 763
2-228. RM_DEFAULT_RSTCTRL Register ..... 764
2-229. RM_DEFAULT_RSTST Register ..... 765
2-230. PM_HDVICP_PWRSTCTRL Register ..... 767
2-231. PM_HDVICP_PWRSTST Register ..... 768
2-232. RM_HDVICP_RSTCTRL Register ..... 769
2-233. RM_HDVICP_RSTST Register ..... 770
2-234. PM_ISP_PWRSTCTRL Register ..... 772
2-235. PM_ISP_PWRSTST Register ..... 773
2-236. RM_ISP_RSTCTRL Register ..... 774
2-237. RM_ISP_RSTST Register ..... 775
2-238. PM_DSS_PWRSTCTRL Register ..... 777
2-239. PM_DSS_PWRSTST Register ..... 778
2-240. RM_DSS_RSTCTRL Register ..... 779
2-241. RM_DSS_RSTST Register ..... 780
2-242. PM_SGX_PWRSTCTRL Register ..... 782
2-243. RM_SGX_RSTCTRL Register ..... 783
2-244. PM_SGX_PWRSTST Register ..... 784
2-245. RM_SGX_RSTST Register ..... 785
2-246. CM_ALWON_L3_SLOW_CLKSTCTRL Register ..... 788
2-247. CM_ETHERNET_CLKSTCTRL Register ..... 791
2-248. CM_ALWON_L3_MED_CLKSTCTRL Register ..... 792
2-249. CM_MMU_CLKSTCTRL Register ..... 793
2-250. CM_MMUCFG_CLKSTCTRL Register ..... 794
2-251. CM_ALWON_OCMC_0_CLKSTCTRL Register ..... 795
2-252. CM_ALWON_VCP_CLKSTCTRL Register ..... 796
2-253. CM_ALWON_MPU_CLKSTCTRL Register ..... 797
2-254. CM_ALWON_SYSCLK4_CLKSTCTRL Register ..... 798
2-255. CM_ALWON_SYSCLK5_CLKSTCTRL Register ..... 799
2-256. CM_ALWON_SYSCLK6_CLKSTCTRL Register ..... 800
2-257. CM_ALWON_RTC_CLKSTCTRL Register ..... 801
2-258. CM_ALWON_L3_FAST_CLKSTCTRL Register ..... 802
2-259. CM_ALWON_MCASP0_CLKCTRL Register ..... 803
2-260. CM_ALWON_MCASP1_CLKCTRL Register ..... 804
2-261. CM_ALWON_MCASP2_CLKCTRL Register ..... 805
2-262. CM_ALWON_MCBSP_CLKCTRL Register ..... 806
2-263. CM_ALWON_UART_0_CLKCTRL Register ..... 807
2-264. CM_ALWON_UART_1_CLKCTRL Register ..... 808
2-265. CM_ALWON_UART_2_CLKCTRL Register ..... 809
2-266. CM_ALWON_GPIO_0_CLKCTRL Register ..... 810
2-267. CM_ALWON_GPIO_1_CLKCTRL Register ..... 811
2-268. CM_ALWON_I2C_0_CLKCTRL Register ..... 812
2-269. CM_ALWON_I2C_1_CLKCTRL Register ..... 813
2-270. CM_ALWON_MCASP_3_4_5_CLKCTRL Register ..... 814
2-271. CM_ALWON_ATL_CLKCTRL Register ..... 815
2-272. CM_ALWON_MLB_CLKCTRL Register ..... 816
2-273. CM_ALWON_PATA_CLKCTRL Register ..... 817
2-274. CM_ALWON_UART_3_CLKCTRL Register ..... 818


Copyright © 2011–2016, Texas Instruments Incorporated

2-275. CM_ALWON_UART_4_CLKCTRL Register ..... 819
2-276. CM_ALWON_UART_5_CLKCTRL Register ..... 820
2-277. CM_ALWON_WDTIMER_CLKCTRL Register ..... 821
2-278. CM_ALWON_SPI_CLKCTRL Register ..... 822
2-279. CM_ALWON_MAILBOX_CLKCTRL Register ..... 823
2-280. CM_ALWON_SPINBOX_CLKCTRL Register ..... 824
2-281. CM_ALWON_MMUDATA_CLKCTRL Register ..... 825
2-282. CM_ALWON_MMUCFG_CLKCTRL Register ..... 826
2-283. CM_ALWON_OCMC_0_CLKCTRL Register ..... 827
2-284. CM_ALWON_VCP_CLKCTRL Register ..... 828
2-285. CM_ALWON_CONTROL_CLKCTRL Register ..... 829
2-286. CM_ALWON_GPMC_CLKCTRL Register ..... 830
2-287. CM_ALWON_ETHERNET_0_CLKCTRL Register ..... 831
2-288. CM_ALWON_MPU_CLKCTRL Register ..... 832
2-289. CM_ALWON_DEBUGSS_CLKCTRL Register ..... 833
2-290. CM_ALWON_L3_CLKCTRL Register ..... 835
2-291. CM_ALWON_L4HS_CLKCTRL Register ..... 836
2-292. CM_ALWON_L4LS_CLKCTRL Register ..... 837
2-293. CM_ALWON_RTC_CLKCTRL Register ..... 838
2-294. CM_ALWON_TPCC_CLKCTRL Register ..... 839
2-295. CM_ALWON_TPTC0_CLKCTRL Register ..... 840
2-296. CM_ALWON_TPTC1_CLKCTRL Register ..... 841
2-297. CM_ALWON_TPTC2_CLKCTRL Register ..... 842
2-298. CM_ALWON_TPTC3_CLKCTRL Register ..... 843
2-299. CM_ALWON_DCAN_0_1_CLKCTRL Register ..... 844
2-300. CM_ALWON_MMCHS_0_CLKCTRL Register ..... 845
2-301. CM_ALWON_MMCHS_1_CLKCTRL Register ..... 846
2-302. CM_ALWON_MMCHS_2_CLKCTRL Register ..... 847
2-303. CM_ALWON_CUST_EFUSE_CLKCTRL Register ..... 848
2-304. RM_ALWON_RSTST Register ..... 850
3-1. MMR_LOCKx Transition from LOCKED/UNLOCKED State ..... 853
3-2. CONTROL_REVISION Register ..... 862
3-3. CONTROL_HWINFO Register ..... 863
3-4. CONTROL_SYSCONFIG Register ..... 864
3-5. CONTROL_STATUS Register ..... 865
3-6. BOOTSTAT Register ..... 867
3-7. DSPBOOTADDR Register ..... 868
3-8. MMR_LOCK0 Register ..... 869
3-9. MMR_LOCK1 Register ..... 870
3-10. MMR_LOCK2 Register ..... 871
3-11. MMR_LOCK3 Register ..... 872
3-12. MMR_LOCK4 Register ..... 873
3-13. CONTROL_SEC_CTRL Register ..... 874
3-14. DEVOSC Register ..... 875
3-15. AUXOSC Register ..... 876
3-16. PCIE_CFG Register ..... 877
3-17. DEVICE_ID Register ..... 878
3-18. DEV_FEATURE Register ..... 879
3-19. INIT_PRIORITY_0 Register ..... 881


Copyright © 2011–2016, Texas Instruments Incorporated

3-20. INIT_PRIORITY_1 Register ..... 882
3-21. MMU_CFG Register ..... 883
3-22. TPTC_CFG Register ..... 884
3-23. DSP_IDLE_CFG Register ..... 885
3-24. USB_CTRL0 Register ..... 886
3-25. USB_STS0 Register ..... 888
3-26. USB_CTRL1 Register ..... 889
3-27. USB_STS1 Register ..... 891
3-28. MAC_ID0_LO Register ..... 892
3-29. MAC_ID0_HI Register ..... 893
3-30. MAC_ID1_LO Register ..... 894
3-31. MAC_ID1_HI Register ..... 895
3-32. SW_REVISION Register ..... 896
3-33. DCAN_RAMINIT Register ..... 897
3-34. AUD_CTRL Register ..... 898
3-35. GMII_SEL Register ..... 899
3-36. OCMEM_PWRDN Register ..... 900
3-37. MEDIA_CONTROLLER_MEM_PWRDN Register ..... 901
3-38. SD_DAC_CTRL Register ..... 902
3-39. SD_DAC0_CAL Register ..... 904
3-40. SD_DAC1_CAL Register ..... 905
3-41. SD_DAC0_REGCTRL Register ..... 906
3-42. SD_DAC0_REGSTATUS Register ..... 907
3-43. SD_DAC1_REGCTRL Register ..... 908
3-44. SD_DAC1_REGSTATUS Register ..... 909
3-45. EMIF_CLK_GATE Register ..... 910
3-46. SMRT_CTRL Register ..... 911
3-47. ARM_CORTEX_A8_HW_DBG_SEL Register ..... 912
3-48. ARM_CORTEX_A8_HW_DBG_INFO Register ..... 913
3-49. PRCM_DEBUG_ALWON_DEFAULT Register ..... 914
3-50. PRCM_DEBUG_PD_DOMAIN_STATUS Register ..... 916
3-51. PCIE_PLLCFG0 Register ..... 919
3-52. PCIE_PLLCFG1 Register ..... 921
3-53. PCIE_PLLCFG2 Register ..... 922
3-54. PCIE_PLLCFG3 Register ..... 923
3-55. PCIE_PLLCFG4 Register ..... 925
3-56. PCIE_PLLSTATUS Register ..... 927
3-57. PCIE_RXSTATUS Register ..... 928
3-58. PCIE_TXSTATUS Register ..... 929
3-59. SATA_PLLCFG0 Register ..... 930
3-60. SATA_PLLCFG1 Register ..... 931
3-61. SATA_PLLCFG2 Register ..... 932
3-62. SATA_PLLCFG3 Register ..... 933
3-63. SATA_PLLCFG4 Register ..... 936
3-64. SATA_TXSTATUS Register ..... 938
3-65. SATA_TESTCFG Register ..... 939
3-66. VDD_MPU_OPP_050 Register ..... 940
3-67. VDD_MPU_OPP_100 Register ..... 941
3-68. VDD_MPU_OPP_120 Register ..... 942


Copyright © 2011–2016, Texas Instruments Incorporated

3-69. VDD_MPU_OPP_166 Register ..... 943
3-70. VDD_C674x_DSP_OPP_050 Register ..... 944
3-71. VDD_C674x_DSP_OPP_100 Register ..... 945
3-72. VDD_C674x_DSP_OPP_120 Register ..... 946
3-73. VDD_C674x_DSP_OPP_166 Register ..... 947
3-74. VDD_HDVICP_OPP_050 Register ..... 948
3-75. VDD_HDVICP_OPP_100 Register ..... 949
3-76. VDD_HDVICP_OPP_120 Register ..... 950
3-77. VDD_HDVICP_OPP_166 Register ..... 951
3-78. VDD_CORE_OPP_050 Register ..... 952
3-79. VDD_CORE_OPP_100 Register ..... 953
3-80. VDD_CORE_OPP_120 Register ..... 954
3-81. VDD_CORE_OPP_166 Register ..... 955
3-82. BB_SCALE Register ..... 956
3-83. USB_VID_PID Register ..... 957
3-84. PCIE_VID_PID Register ..... 958
3-85. PINCNTL[270:1] Register ..... 959
3-86. MLBP_SIG_IO_CTRL Register ..... 960
3-87. MLBP_DAT_IO_CTRL Register ..... 961
3-88. SERDES_REFCLK_CTL Register ..... 962
3-89. DSP_INTMUX_15_18 Register ..... 963
3-90. DSP_INTMUX_19_22 Register ..... 964
3-91. DSP_INTMUX_23_26 Register ..... 965
3-92. DSP_INTMUX_27_30 Register ..... 966
3-93. DSP_INTMUX_31_34 Register ..... 967
3-94. DSP_INTMUX_35_38 Register ..... 968
3-95. DSP_INTMUX_39_42 Register ..... 969
3-96. DSP_INTMUX_43_46 Register ..... 970
3-97. DSP_INTMUX_47_50 Register ..... 971
3-98. DSP_INTMUX_51_54 Register ..... 972
3-99. DSP_INTMUX_55_58 Register ..... 973
3-100. DSP_INTMUX_59_62 Register ..... 974
3-101. DSP_INTMUX_63_66 Register ..... 975
3-102. DSP_INTMUX_67_70 Register ..... 976
3-103. DSP_INTMUX_71_74 Register ..... 977
3-104. DSP_INTMUX_75_78 Register ..... 978
3-105. DSP_INTMUX_79_82 Register ..... 979
3-106. DSP_INTMUX_83_86 Register ..... 980
3-107. DSP_INTMUX_87_90 Register ..... 981
3-108. DSP_INTMUX_91_94 Register ..... 982
3-109. DSP_INTMUX_95 Register ..... 983
3-110. Media_Controller_INTMUX_0_3 Register ..... 984
3-111. Media_Controller_INTMUX_4_7 Register ..... 985
3-112. Media_Controller_INTMUX_8_11 Register ..... 986
3-113. Media_Controller_INTMUX_12_15 Register ..... 987
3-114. Media_Controller_INTMUX_16_19 Register ..... 988
3-115. Media_Controller_INTMUX_20_23 Register ..... 989
3-116. Media_Controller_INTMUX_24_27 Register ..... 990
3-117. Media_Controller_INTMUX_28_31 Register ..... 991


Copyright © 2011–2016, Texas Instruments Incorporated

3-118. Media_Controller_INTMUX_32_35 Register ..... 992
3-119. Media_Controller_INTMUX_36_39 Register ..... 993
3-120. Media_Controller_INTMUX_40_43 Register ..... 994
3-121. Media_Controller_INTMUX_44_47 Register ..... 995
3-122. Media_Controller_INTMUX_48_51 Register ..... 996
3-123. Media_Controller_INTMUX_52_55 Register ..... 997
3-124. Media_Controller_INTMUX_56 Register ..... 998
3-125. EDMA3CC_EVTMUX_0_3 Register ..... 999
3-126. EDMA3CC_EVTMUX_4_7 Register ..... 1000
3-127. EDMA3CC_EVTMUX_8_11 Register ..... 1001
3-128. EDMA3CC_EVTMUX_12_15 Register ..... 1002
3-129. EDMA3CC_EVTMUX_16_19 Register ..... 1003
3-130. EDMA3CC_EVTMUX_20_23 Register ..... 1004
3-131. EDMA3CC_EVTMUX_24_27 Register ..... 1005
3-132. EDMA3CC_EVTMUX_28_31 Register ..... 1006
3-133. EDMA3CC_EVTMUX_32_35 Register ..... 1007
3-134. EDMA3CC_EVTMUX_36_39 Register ..... 1008
3-135. EDMA3CC_EVTMUX_40_43 Register ..... 1009
3-136. EDMA3CC_EVTMUX_44_47 Register ..... 1010
3-137. EDMA3CC_EVTMUX_48_51 Register ..... 1011
3-138. EDMA3CC_EVTMUX_52_55 Register ..... 1012
3-139. EDMA3CC_EVTMUX_56_59 Register ..... 1013
3-140. EDMA3CC_EVTMUX_60_63 Register ..... 1014
3-141. GPIO_MUX Register ..... 1015
3-142. RESET_ISO Register ..... 1016
4-1. ROM Code Architecture ..... 1020
4-2. ROM Code Boot Procedure ..... 1021
4-3. ROM Memory Map ..... 1022
4-4. Public RAM Memory Map ..... 1024
4-5. ROM Code Start-up Sequence ..... 1026
4-6. ROM Code Booting Procedure ..... 1028
4-7. Fast External Boot ..... 1030
4-8. Memory Booting ..... 1031
4-9. GPMC XIP Timings ..... 1033
4-10. Image Shadowing on the Device ..... 1035
4-11. GPMC NAND Timings ..... 1036
4-12. NAND Device Detection ..... 1041
4-13. NAND Invalid Blocks Detection ..... 1042
4-14. NAND Read Sector Procedure ..... 1043
4-15. ECC Data Mapping for 2KB Page and 8b BCH Encoding ..... 1044
4-16. ECC Data Mapping for 4KB Page and 16b BCH Encoding ..... 1045
4-17. MMC/SD Booting ..... 1047
4-18. MMC/SD Detection Procedure ..... 1048
4-19. MMC/SD Booting, Get Booting File ..... 1050
4-20. MBR Detection Procedure ..... 1052
4-21. MBR, Get Partition ..... 1053
4-22. FAT Detection Procedure ..... 1056
4-23. Peripheral Booting Procedure ..... 1060
4-24. PCIe Peripheral Booting Procedure ..... 1063


Copyright © 2011–2016, Texas Instruments Incorporated

4-25. Image Formats ..... 1065
4-26. TOC Structure ..... 1066
5-1. DCAN Block Diagram ..... 1074
5-2. CAN Module General Initialization Flow ..... 1075
5-3. CAN Bit-Timing Configuration ..... 1076
5-4. CAN Core in Silent Mode ..... 1079
5-5. CAN Core in Loopback Mode ..... 1079
5-6. CAN Core in External Loopback Mode ..... 1080
5-7. CAN Core in Loop Back Combined With Silent Mode ..... 1081
5-8. CAN Interrupt Topology 1 ..... 1083
5-9. CAN Interrupt Topology 2 ..... 1083
5-10. Local Power-Down Mode Flow Diagram ..... 1085
5-11. CPU Handling of a FIFO Buffer (Interrupt Driven) ..... 1094
5-12. Bit Timing ..... 1095
5-13. The Propagation Time Segment ..... 1096
5-14. Synchronization on Late and Early Edges ..... 1098
5-15. Filtering of Short Dominant Spikes ..... 1099
5-16. Structure of the CAN Core's CAN Protocol Controller ..... 1100
5-17. Data Transfer Between IF1/IF2 Registers and Message RAM ..... 1104
5-18. CAN Control Register (DCAN CTL) ..... 1112
5-19. Error and Status Register (DCAN ES) ..... 1114
5-20. Parity Error End of Interrupt Register (PARITYERR_EOI) ..... 1114
5-21. Error Counter Register (DCAN ERRC) ..... 1116
5-22. Bit Timing Register (DCAN BTR) ..... 1117
5-23. Interrupt Register (DCAN INT) ..... 1118
5-24. Test Register (DCAN TEST) ..... 1119
5-25. Parity Error Code Register (DCAN PERR) ..... 1120
5-26. Auto-Bus-On Time Register (DCAN ABOTR) ..... 1121
5-27. Transmission Request X Register (DCAN TXRQ X) ..... 1122
5-28. Transmission Request X Register (DCAN TXRQ X) ..... 1124
5-29. Interrupt Pending X Register (DCAN INTPND X) ..... 1126
5-30. Message Valid X Register (DCAN MSGVAL X) ..... 1128
5-31. IF1 Command Registers (DCAN IF1CMD) ..... 1131
5-32. IF2 Command Registers (DCAN IF2CMD) ..... 1131
5-33. IF1 Mask Register (DCAN IF1MSK) ..... 1134
5-34. IF2 Mask Register (DCAN IF2MSK) ..... 1134
5-35. IF1 Arbitration Register (DCAN IF1ARB) ..... 1135
5-36. IF2 Arbitration Register (DCAN IF2ARB) ..... 1135
5-37. IF1 Message Control Register (DCAN IF1MCTL) ..... 1137
5-38. IF2 Message Control Register (DCAN IF2MCTL) ..... 1137
5-39. IF1 Data A Register (DCAN IF1DATA) ..... 1139
5-40. IF1 Data B Register (DCAN IF1DATA) ..... 1139
5-41. IF2 Data A Register (DCAN IF2DATA) ..... 1139
5-42. IF2 Data B Register (DCAN IF2DATA) ..... 1139
5-43. IF3 Observation Register (DCAN IF3OBS) ..... 1140
5-44. IF3 Mask Register (DCAN IF3MSK) ..... 1142
5-45. IF3 Arbitration Register (DCAN IF3ARB) ..... 1143
5-46. IF3 Message Control Register (DCAN IF3MCTL) ..... 1144
5-47. IF3 Data A Register (DCAN IF3DATA) ..... 1146


Copyright © 2011–2016, Texas Instruments Incorporated

5-48. IF3 Data A Register (DCAN IF3DATB) ..... 1146
5-49. CAN TX I/O Control Register (DCAN TIOC) ..... 1148
5-50. CAN RX IO control register (DCAN RIOC) ..... 1150
6-1. DMM Integration ..... 1153
6-2. DMM Block Diagram ..... 1154
6-3. DMM Look-Up Table ..... 1158
6-4. DMM PAT Direct Access Translation ..... 1158
6-5. DMM PAT In-Direct Access Translation ..... 1159
6-6. DMM Section and Memory Mapping ..... 1161
6-7. 128B and 256B Interleaving ..... 1162
6-8. 512KB and 1KB Interleaving ..... 1162
6-9. Overview of Request Conversion ..... 1163
6-10. Memory Map ..... 1164
6-11. DMM Address Translations ..... 1166
6-12. Image Stored in the Memory Linearly ..... 1167
6-13. Image Stored in the Memory by TILER ..... 1168
6-14. Address Space Structure for Tiled Modes ..... 1169
6-15. 4KB Page, 8-bit Mode ..... 1170
6-16. 4KB Page, 16-bit Mode ..... 1171
6-17. 4KB Page, 32-bit Mode ..... 1171
6-18. Four 1KB Tiles in One 4KB Page ..... 1172
6-19. 8-bit Sub-tile ..... 1173
6-20. 16-bit Sub-tile ..... 1173
6-21. 32-bit Sub-tile ..... 1174
6-22. Address Format ..... 1175
6-23. TILER Object Containers and Views ..... 1176
6-24. Using LUT to Translate Tiled Virtual Address to Physical SDRC Address ..... 1177
6-25. Object Container Geometry with 4KB Pages ..... 1177
6-26. TILER Page Mapping When Using 4KB Pages ..... 1178
6-27. Isometric Transforms in the TILER Container ..... 1179
6-28. Paged Mode Addressing ..... 1180
6-29. Tiled Mode Addressing in 0° or 180° Orientations, (S = 0) ..... 1181
6-30. Tiled Mode Addressing in 90° or 270° Orientations, (S = 1) ..... 1181
6-31. Tiled Mode Ordering of Elements in Natural View ..... 1183
6-32. Page Mode Ordering of Elements in Natural View ..... 1183
6-33. Tiled Mode Ordering of Elements in 0° View with Vertical Mirror ..... 1184
6-34. Page Mode Ordering of Elements in 0° View with Vertical Mirror ..... 1184
6-35. Tiled Mode Ordering of Elements in 0° View with Horizontal Mirror ..... 1185
6-36. Page Mode Ordering of Elements in 0° View with Horizontal Mirror ..... 1185
6-37. Tiled Mode Ordering of Elements in 180° View ..... 1186
6-38. Page Mode Ordering of Elements in 180° View ..... 1186
6-39. Tiled Mode Ordering of Elements in 90º View with Vertical Mirror ..... 1187
6-40. Page Mode Ordering of Elements in 90º View with Vertical Mirror ..... 1187
6-41. Tiled Mode Ordering of Elements in 270° View ..... 1188
6-42. Page Mode Ordering of Elements in 270° View ..... 1188
6-43. Tiled Mode Ordering of Elements in 90° View ..... 1189
6-44. Page Mode Ordering of Elements in 90° View ..... 1189
6-45. Tiled Mode Ordering of Elements in 90º View with Horizontal Mirror ..... 1190
6-46. Page Mode Ordering of Elements in 90º View with Horizontal Mirror ..... 1190


Copyright © 2011–2016, Texas Instruments Incorporated

6-47. Buffer Arrangement for HD Luma Buffers in 128MB 8-bit Mode Container ..... 1192
6-48. Buffer Arrangement for HD Chroma Buffers in 128MB 16-bit Mode Container ..... 1193
6-49. PAT Descriptor Node ..... 1194
6-50. PAT Area Description ..... 1195
6-51. DMM Simple Manual Area Refill ..... 1196
6-52. DMM Single Auto-configured Area Refill ..... 1197
6-53. DMM Chained Auto-configured Area Refill ..... 1198
6-54. DMM Synchronised Auto-configured Area Refill ..... 1199
6-55. DMM Cyclic Synchronised Auto-configured Area Refill ..... 1200
6-56. DMM Section Use-Case 2 ..... 1204
6-57. DMM_REVISION Register ..... 1206
6-58. DMM_SYSCONFIG Register ..... 1207
6-59. DMM_LISA_LOCK Register ..... 1207
6-60. DMM_LISA_MAP Registers ..... 1208
6-61. DMM_TILER_OR Registers ..... 1209
6-62. DMM_PAT_CONFIG Register ..... 1209
6-63. DMM_PAT_VIEW Registers ..... 1210
6-64. DMM_PAT_VIEW_MAP Registers ..... 1211
6-65. DMM_PAT_VIEW_MAP_BASE Register ..... 1212
6-66. DMM_PAT_IRQ_EOI Register ..... 1212
6-67. DMM_PAT_IRQSTATUS_RAW Register ..... 1214
6-68. DMM_PAT_IRQSTATUS Register ..... 1216
6-69. DMM_PAT_IRQENABLE_SET Register ..... 1218
6-70. DMM_PAT_IRQENABLE_CLR Register ..... 1220
6-71. DMM_PAT_STATUS Registers ..... 1222
6-72. DMM_PAT_DESCR Registers ..... 1223
6-73. DMM_PAT_AREA Registers ..... 1223
6-74. DMM_PAT_CTRL Registers ..... 1224
6-75. DMM_PAT_DATA Registers ..... 1224
6-76. DMM_PEG_PRIO Registers ..... 1225
6-77. DMM_PEG_PRIO_PAT Register ..... 1225
7-1. DDR Block Diagram ..... 1228
7-2. DDR2 and DDR3 Memory Controller Signals ..... 1229
7-3. DDR2/DDR3 Subsystem Block Diagram ..... 1231
7-4. DDR2/DDR3 Memory Controller FIFO Block Diagram ..... 1232
7-5. SDRAM Status Register (SDRSTAT) ..... 1262
7-6. SDRAM Configuration Register (SDRCR) ..... 1263
7-7. SDRAM Configuration Register 2 (SDRCR2) ..... 1266
7-8. SDRAM Refresh Control Register (SDRRCR) ..... 1267
7-9. SDRAM Refresh Control Shadow Register (SDRRCSR) ..... 1268
7-10. SDRAM Timing 1 Register (SDRTIM1) ..... 1269
7-11. SDRAM Timing 1 Shadow Register (SDRTIM1SR) ..... 1270
7-12. SDRAM Timing 2 Register (SDRTIM2) ..... 1271
7-13. SDRAM Timing 2 Shadow Register (SDRTIM2SR) ..... 1272
7-14. SDRAM Timing 3 Register (SDRTIM3) ..... 1273
7-15. SDRAM Timing 3 Shadow Register (SDRTIM3SR) ..... 1274
7-16. Power Management Control Register (PMCR) ..... 1275
7-17. Power Management Control Shadow Register (PMCSR) ..... 1277
7-18. Peripheral Bus Burst Priority Register (PBBPR) ..... 1278


Copyright © 2011–2016, Texas Instruments Incorporated

7-19. Performance Counter 1 Register (PERF_CNT_1) ..... 1278
7-20. Performance Counter 2 Register (PERF_CNT_2) ..... 1279
7-21. Performance Counter Config Register (PERF_CNT_CFG) ..... 1279
7-22. Performance Counter Master Region Select Register (PERF_CNT_SEL) ..... 1280
7-23. End of Interrupt Register (EOI) ..... 1280
7-24. System OCP Interrup RAW Status Register (SOIRSR) ..... 1281
7-25. System OCP Interrupt Status Register (SOISR) ..... 1281
7-26. System OCP Interrupt Enable Set Register (SOIESR) ..... 1282
7-27. System OCP Interrupt Enable Clear Register (SOIECR) ..... 1282
7-28. SDRAM Output Impedance Calibration Configuration Register (ZQCR) ..... 1283
7-29. DDR PHY Control Register (DDRPHYCR) ..... 1284
7-30. DDR PHY Control Shadow Register (DDRPHYCSR) ..... 1285
7-31. Priority to Class of Service Mapping Register (PRI_COS_MAP) ..... 1286
7-32. Connection ID to Class of Service 1 Mapping Register(CONNID_COS_1_MAP) ..... 1287
7-33. Connection ID to Class of Service 2 Mapping Register (CONNID_COS_2_MAP) ..... 1288
7-34. Read Write Execution Threshold Register (RD_WR_EXEC_THRSH) ..... 1289
7-35. DDR PHY Command 0/1/2 Address/Command Slave Ratio Register (CMD0/1/2_REG_PHY_CTRL_SLAVE_RATIO_0) ..... 1291
7-36. DDR PHY Command 0/1/2 Address/Command DLL Lock Difference Register (CMD0/1/2_REG_PHY_DLL_LOCK_DIFF_0) ..... 1292
7-37. DDR PHY Command 0/1/2 Invert Clockout Selection Register (CMD0/1/2_REG_PHY_INVERT_CLKOUT_0) ..... 1292
7-38. DDR PHY Data Macro 0/1/2/3 Read DQS Slave Ratio Register (DATA0/1/2/3_REG_PHY_RD_DQS_SLAVE_RATIO_0)) ..... 1293
7-39. DDR PHY Data Macro 0/1/2/3 DQS Gate Slave Ratio Register(DATA0/1/2/3_REG_PHY_FIFO_WE_SLAVE_RATIO_0) ..... 1294
7-40. DDR PHY Data Macro 0/1/2/3 Write Data Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DATA_SLAVE_RATIO_0) ..... 1295
7-41. DDR PHY Data Macro 0/1/2/3 Delay Selection Register (DATA0/1/2/3_REG_PHY_USE_RANK0_DELAYS) ..... 1296
7-42. DDR PHY Data Macro 0/1/2/3 DLL Lock Difference Register (DATA0/1/2/3_REG_PHY_DLL_LOCK_DIFF_0) ..... 1296
7-43. EMIF0/1 Clock Gate Control Register ..... 1297
7-44. DDR Memory Controller0_ IO Control Register (DDR0_IO_CTRL) ..... 1298
7-45. DDR Memory Controller1_ IO Control Register (DDR1_IO_CTRL) ..... 1299
7-46. DDR VTP Control Register (DDR_VTP_CTRL_0) ..... 1300
7-47. DDR VTP Control Register 1 (DDR_VTP_CTRL_1) ..... 1301
8-1. EDMA3 Controller Block Diagram ..... 1307
8-2. EDMA3 Channel Controller (EDMA3CC) Block Diagram ..... 1308
8-3. EDMA3 Transfer Controller (EDMA3TC) Block Diagram ..... 1309
8-4. Definition of ACNT, BCNT, and CCNT ..... 1310
8-5. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3) ..... 1311
8-6. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3) ..... 1312
8-7. PaRAM Set ..... 1313
8-8. Channel Options Parameter (OPT) ..... 1315
8-9. Linked Transfer ..... 1322
8-10. Link-to-Self Transfer ..... 1323
8-11. DMA Channel and QDMA Channel to PaRAM Mapping ..... 1328
8-12. QDMA Channel to PaRAM Mapping ..... 1329
8-13. Shadow Region Registers ..... 1330
8-14. Interrupt Diagram ..... 1334


Copyright © 2011–2016, Texas Instruments Incorporated

8-15. Error Interrupt Operation ..... 1337
8-16. PaRAM Set Content for Proxy Memory Protection Example ..... 1341
8-17. Channel Options Parameter (OPT) Example ..... 1341
8-18. Proxy Memory Protection Example ..... 1342
8-19. EDMA3 Prioritization ..... 1348
8-20. Block Move Example ..... 1350
8-21. Block Move Example PaRAM Configuration ..... 1351
8-22. Subframe Extraction Example ..... 1352
8-23. Subframe Extraction Example PaRAM Configuration ..... 1352
8-24. Data Sorting Example ..... 1353
8-25. Data Sorting Example PaRAM Configuration ..... 1354
8-26. Servicing Incoming McASP Data Example ..... 1355
8-27. Servicing Incoming McASP Data Example PaRAM Configuration ..... 1356
8-28. Servicing Peripheral Burst Example ..... 1357
8-29. Servicing Peripheral Burst Example PaRAM Configuration ..... 1358
8-30. Servicing Continuous McASP Data Example ..... 1359
8-31. Servicing Continuous McASP Data Example PaRAM Configuration ..... 1360
8-32. Servicing Continuous McASP Data Example Reload PaRAM Configuration ..... 1361
8-33. Ping-Pong Buffering for McASP Data Example ..... 1363
8-34. Ping-Pong Buffering for McASP Example PaRAM Configuration ..... 1364
8-35. Ping-Pong Buffering for McASP Example Pong PaRAM Configuration ..... 1365
8-36. Ping-Pong Buffering for McASP Example Ping PaRAM Configuration ..... 1365
8-37. Intermediate Transfer Completion Chaining Example ..... 1367
8-38. Single Large Block Transfer Example ..... 1368
8-39. Smaller Packet Data Transfers Example ..... 1368
8-40. Peripheral ID Register (PID) ..... 1372
8-41. EDMA3CC Configuration Register (CCCFG) ..... 1373
8-42. DMA Channel Map *n* Registers (DCHMAP*n*) ..... 1375
8-43. QDMA Channel Map *n* Registers (QCHMAP*n*) ..... 1376
8-44. DMA Channel Queue *n* Number Registers (DMAQNUM*n*) ..... 1377
8-45. QDMA Channel Queue Number Register (QDMAQNUM) ..... 1378
8-46. Event Missed Register (EMR) ..... 1379
8-47. Event Missed Register High (EMRH) ..... 1379
8-48. Event Missed Clear Register (EMCR) ..... 1380
8-49. Event Missed Clear Register High (EMCRH) ..... 1380
8-50. QDMA Event Missed Register (QEMR) ..... 1381
8-51. QDMA Event Missed Clear Register (QEMCR) ..... 1382
8-52. EDMA3CC Error Register (CCERR) ..... 1383
8-53. EDMA3CC Error Clear Register (CCERRCLR) ..... 1384
8-54. Error Evaluation Register (EEVAL) ..... 1385
8-55. DMA Region Access Enable Register for Region *m* (DRAE*m*) ..... 1386
8-56. DMA Region Access Enable High Register for Region *m* (DRAEH*m*) ..... 1386
8-57. QDMA Region Access Enable for Region *m* (QRAE*m*) ..... 1387
8-58. Event Queue Entry Registers (Q*x*E*y*) ..... 1388
8-59. Queue Status Register *n* (QSTAT*n*) ..... 1389
8-60. Queue Watermark Threshold A Register (QWMTHRA) ..... 1390
8-61. EDMA3CC Status Register (CCSTAT) ..... 1391
8-62. Memory Protection Fault Address Register (MPFAR) ..... 1393
8-63. Memory Protection Fault Status Register (MPFSR) ..... 1394


Copyright © 2011–2016, Texas Instruments Incorporated

8-64. Memory Protection Fault Command Register (MPFCR) ..... 1395
8-65. Memory Protection Page Attribute Register (MPPA*n*) ..... 1396
8-66. Event Register (ER) ..... 1398
8-67. Event Register High (ERH) ..... 1398
8-68. Event Clear Register (ECR) ..... 1399
8-69. Event Clear Register High (ECRH) ..... 1399
8-70. Event Set Register (ESR) ..... 1400
8-71. Event Set Register High (ESRH) ..... 1401
8-72. Chained Event Register (CER) ..... 1402
8-73. Chained Event Register High (CERH) ..... 1403
8-74. Event Enable Register (EER) ..... 1404
8-75. Event Enable Register High (EERH) ..... 1404
8-76. Event Enable Clear Register (EECR) ..... 1405
8-77. Event Enable Clear Register High (EECRH) ..... 1405
8-78. Event Enable Set Register (EESR) ..... 1406
8-79. Event Enable Set Register High (EESRH) ..... 1406
8-80. Secondary Event Register (SER) ..... 1407
8-81. Secondary Event Register High (SERH) ..... 1407
8-82. Secondary Event Clear Register (SECR) ..... 1408
8-83. Secondary Event Clear Register High (SECRH) ..... 1408
8-84. Interrupt Enable Register (IER) ..... 1409
8-85. Interrupt Enable Register High (IERH) ..... 1409
8-86. Interrupt Enable Clear Register (IECR) ..... 1410
8-87. Interrupt Enable Clear Register High (IECRH) ..... 1410
8-88. Interrupt Enable Set Register (IESR) ..... 1411
8-89. Interrupt Enable Set Register High (IESRH) ..... 1411
8-90. Interrupt Pending Register (IPR) ..... 1412
8-91. Interrupt Pending Register High (IPRH) ..... 1412
8-92. Interrupt Clear Register (ICR) ..... 1413
8-93. Interrupt Clear Register High (ICRH) ..... 1413
8-94. Interrupt Evaluate Register (IEVAL) ..... 1414
8-95. QDMA Event Register (QER) ..... 1415
8-96. QDMA Event Enable Register (QEER) ..... 1416
8-97. QDMA Event Enable Clear Register (QEECR) ..... 1417
8-98. QDMA Event Enable Set Register (QEESR) ..... 1418
8-99. QDMA Secondary Event Register (QSER) ..... 1419
8-100. QDMA Secondary Event Clear Register (QSECR) ..... 1420
8-101. Peripheral ID Register (PID) ..... 1422
8-102. EDMA3TC Configuration Register (TCCFG) ..... 1423
8-103. EDMA3TC Channel Status Register (TCSTAT) ..... 1424
8-104. Error Register (ERRSTAT) ..... 1426
8-105. Error Enable Register (ERREN) ..... 1427
8-106. Error Clear Register (ERRCLR) ..... 1428
8-107. Error Details Register (ERRDET) ..... 1429
8-108. Error Interrupt Command Register (ERRCMD) ..... 1430
8-109. Read Rate Register (RDRATE) ..... 1431
8-110. Source Active Options Register (SAOPT) ..... 1432
8-111. Source Active Source Address Register (SASRC) ..... 1434
8-112. Source Active Count Register (SACNT) ..... 1434


Copyright © 2011–2016, Texas Instruments Incorporated

8-113. Source Active Destination Address Register (SADST) ..... 1435
8-114. Source Active Source B-Dimension Index Register (SABIDX) ..... 1435
8-115. Source Active Memory Protection Proxy Register (SAMPPRXY) ..... 1436
8-116. Source Active Count Reload Register (SACNTRLD) ..... 1437
8-117. Source Active Source Address B-Reference Register (SASRCBREF) ..... 1437
8-118. Source Active Destination Address B-Reference Register (SADSTBREF) ..... 1438
8-119. Destination FIFO Options Register (DFOPT*n*) ..... 1439
8-120. Destination FIFO Source Address Register (DFSRC*n*) ..... 1441
8-121. Destination FIFO Count Register (DFCNT*n*) ..... 1441
8-122. Destination FIFO Destination Address Register (DFDST*n*) ..... 1442
8-123. Destination FIFO B-Index Register (DFBIDX*n*) ..... 1442
8-124. Destination FIFO Memory Protection Proxy Register (DFMPPRXY*n*) ..... 1443
8-125. Destination FIFO Count Reload Register (DFCNTRLD*n*) ..... 1444
8-126. Destination FIFO Source Address B-Reference Register (DFSRCBREF*n*) ..... 1444
8-127. Destination FIFO Destination Address B-Reference Register (DFDSTBREF*n*) ..... 1445
9-1. 3PSW Ethernet Subsystem Top Level Block Diagram ..... 1452
9-2. 3PSW Subsystem Clocking Block Diagram ..... 1455
9-3. CPSW_3G Block Diagram ..... 1457
9-4. G/MII Interface Connections ..... 1472
9-5. RMII Interface Connections ..... 1474
9-6. RGMII Interface Connections ..... 1476
9-7. CPTS Block Diagram ..... 1483
9-8. CPTS RCLK Source ..... 1483
9-9. Event FIFO Misalignment Condition ..... 1485
9-10. HW1/4_TSP_PUSH Connection ..... 1486
9-11. Partial Ethernet-II Frames Showing Register Mapping of EtherTypes for a Simple Frame (1), a Single 1Q Tag Added (2), and Two 1Q Tags Added (3) ..... 1487
9-12. TX Buffer Descriptor Format ..... 1489
9-13. Receive Buffer Descriptor Format ..... 1492
9-14. TX Queue Head Descriptor ..... 1504
9-15. RX Queue Head Descriptor ..... 1506
9-16. CPSW ID Version Register (CPSW_ID_VER) ..... 1514
9-17. CPSW Control Register (CPSW_CONTROL) ..... 1515
9-18. CPSW Software Reset Register (CPSW_SOFT_RESET) ..... 1516
9-19. CPSW Statistics Port Enable Register (CPSW_STAT_PORT_EN) ..... 1516
9-20. CPSW Transmit Priority Type Register (CPSW_PTYPE) ..... 1517
9-21. CPSW Software Idle Register (CPSW_SOFT_IDLE) ..... 1518
9-22. CPSW Throughput Rate Register (CPSW_THRU_RATE) ..... 1518
9-23. CPSW CPGMAC_SL Short Gap Threshold Register (CPSW_GAP_THRESH) ..... 1519
9-24. CPSW Transmit Start Words Register (CPSW_TX_START_WDS) ..... 1519
9-25. CPSW Flow Control Register (CPSW_FLOW_CONTROL) ..... 1520
9-26. CPSW Port 0 Maximum FIFO Blocks Register (P0_MAX_BLKS) ..... 1520
9-27. CPSW Port 0 FIFO Block Usage Count (Read Only) Register (P0_BLK_CNT) ..... 1521
9-28. CPSW Port 0 Transmit FIFO Control Register (P0_TX_IN_CTL) ..... 1521
9-29. CPSW Port 0 VLAN Register (P0_PORT_VLAN) ..... 1522
9-30. CPSW Port 0 TX Header Priority to Switch Priority Mapping Register (P0_TX_PRI_MAP) ..... 1522
9-31. CPSW CPDMA TX (Port 0 RX) Packet Priority to Header Priority Mapping Register (CPDMA_TX_PRI_MAP) ..... 1523
9-32. CPSW CPDMA RX (Port 0 TX) Switch Priority to DMA Channel Mapping Register (CPDMA_RX_CH_MAP) ..... 1524


Copyright © 2011–2016, Texas Instruments Incorporated

9-33. CPSW Port 1 Maximum FIFO Blocks Register (P1_MAX_BLKS) ..... 1525
9-34. CPSW Port 1 FIFO Block Usage Count (Read Only) Register (P1_BLK_CNT) ..... 1525
9-35. CPSW Port 1 Transmit FIFO Control Register (P1_TX_IN_CTL) ..... 1526
9-36. CPSW Port 1 VLAN Register (P1_PORT_VLAN) ..... 1526
9-37. CPSW Port 1 TX Header Priority to Switch Pri Mapping Register (P1_TX_PRI_MAP) ..... 1527
9-38. CPSW_3G Port 1 Time Sync Control Register (P1_TS_CTL) ..... 1528
9-39. CPSW_3G Port 1 Time Sync Sequence ID and LTYPE Register (P1_TS_SEQ_LTYPE) ..... 1529
9-40. CPSW_3G Port 1 Time Sync VLAN2 and VLAN2 Register (P1_TS_VLAN) ..... 1529
9-41. CPSW CPGMAC_SL1 Source Address Low Register (SL1_SA_LO) ..... 1530
9-42. CPSW CPGMAC_SL1 Source Address High Register (SL1_SA_HI) ..... 1530
9-43. CPSW Port 1 Transmit Queue Send Percentages Register (P1_SEND_PERCENT) ..... 1531
9-44. CPSW Port 2 Maximum FIFO Blocks Register (P2_MAX_BLKS) ..... 1532
9-45. CPSW Port 2 FIFO Block Usage Count (Read Only) Register (P2_BLK_CNT) ..... 1532
9-46. CPSW Port 2 Transmit FIFO Control Register (P2_TX_IN_CTL) ..... 1533
9-47. CPSW Port 2 VLAN Register (P2_PORT_VLAN) ..... 1533
9-48. CPSW Port 2 TX Header Priority to Switch Priority Mapping Register (P2_TX_PRI_MAP) ..... 1534
9-49. CPSW_3GF Port 2 Time Sync Control Register (P2_TS_CTL) ..... 1535
9-50. Port 2 Time Sync Sequence ID and LTYPE Register (P2_TS_SEQ_LTYPE) ..... 1536
9-51. CPSW_3GF Port 2 Time Sync VLAN2 and VLAN2 Register (P2_TS_VLAN) ..... 1536
9-52. CPSW CPGMAC_SL2 Source Address Low Register (SL2_SA_LO) ..... 1537
9-53. CPGMAC_SL2 Source Address High Register (SL2_SA_HI) ..... 1537
9-54. Port 2 Transmit Queue Send Percentages Register (P2_SEND_PERCENT) ..... 1538
9-55. CPDMA_REGS TX Identification and Version Register (TX_IDVER) ..... 1539
9-56. CPDMA_REGS TX Control Register (TX_CONTROL) ..... 1539
9-57. CPDMA_REGS TX Teardown Register (TX_TEARDOWN) ..... 1539
9-58. CPDMA_REGS RX Identification and Version Register (RX_IDVER) ..... 1540
9-59. CPDMA_REGS RX Control Register (RX_CONTROL) ..... 1540
9-60. CPDMA_REGS RX Teardown Register (RX_TEARDOWN) ..... 1540
9-61. CPDMA_REGS Software Reset Register (SOFT_RESET) ..... 1541
9-62. CPDMA_REGS Control Register (DMACONTROL) ..... 1541
9-63. CPDMA_REGS Status Register (DMASTATUS) ..... 1543
9-64. CPDMA_REGS Receive Buffer Offset Register (RX_BUFFER_OFFSET) ..... 1544
9-65. CPDMA_REGS Emulation Control Register (EMCONTROL) ..... 1544
9-66. CPDMA_REGS Transmit Priority Rate Registers ..... 1544
9-67. CPDMA_INT TX Interrupt Status (raw value) Register (TX_INTSTAT_RAW) ..... 1545
9-68. CPDMA_INT TX Interrupt Status (masked value) Register (TX_INTSTAT_MASKED) ..... 1545
9-69. CPDMA_INT TX Interrupt Mask Set Register (TX _INTMASK_SET) ..... 1546
9-70. CPDMA_INT TX Interrupt Mask Clear Register (TX_INTMASK_CLEAR) ..... 1546
9-71. CPDMA_INT Input Vector (CPDMA_IN_VECTOR) ..... 1547
9-72. CPDMA_INT End Of Interrupt Vector (CPDMA_EOI_VECTOR) ..... 1547
9-73. CPDMA_INT RX Interrupt Status (raw value) Register (RX_INTSTAT_RAW) ..... 1548
9-74. CPDMA_INT RX Interrupt Status (masked value) Register (RX_INTSTAT_MASKED) ..... 1549
9-75. CPDMA_INT RX Interrupt Mask Set Register (RX_INTMASK_SET) ..... 1550
9-76. CPDMA_INT RX Interrupt Mask Clear Register (RX_INTMASK_CLEAR) ..... 1551
9-77. CPDMA_INT DMA Interrupt Status (raw value) Register (DMA_INTSTAT_RAW) ..... 1552
9-78. CPDMA_INT DMA Interrupt Status (masked value) Register (DMA_INTSTAT_MASKED) ..... 1552
9-79. CPDMA_INT DMA Interrupt Mask Set Register (DMA_INTMASK_SET) ..... 1553
9-80. CPDMA_INT DMA Interrupt Mask Clear Register (DMA_INTMASK_CLEAR) ..... 1553
9-81. CPDMA_INT RX Channel Threshold Pending Channel Registers ..... 1554


Copyright © 2011–2016, Texas Instruments Incorporated

9-82. CPDMA_INT RX Channel Free Buffer Count Registers ..... 1554
9-83. CPDMA_STATERAM TX Channel Head Descriptor Pointers ..... 1554
9-84. CPDMA_STATERAM RX Channel Head Descriptor Pointers ..... 1555
9-85. CPDMA_STATERAM TX Channel Completion Pointer Registers ..... 1555
9-86. CPDMA_STATERAM RX Channel Completion Pointer Registers ..... 1555
9-87. Statistics Register ..... 1556
9-88. Statistics Register ..... 1560
9-89. Statistics Register ..... 1563
9-90. Identification and Version Register (CPTS_IDVER) ..... 1565
9-91. Time Sync Control Register (CPTS_CONTROL) ..... 1565
9-92. Reference Clock Select Register (CPTS_RFTCLK_SEL) ..... 1566
9-93. Time Stamp Event Push Register (CPTS_TS_PUSH) ..... 1566
9-94. Time Stamp Load Value Register (CPTS_TS_LOAD_VAL) ..... 1566
9-95. Time Stamp Load Enable Register (CPTS_TS_LOAD_EN) ..... 1567
9-96. Time Sync Interrupt Status Register Raw Register (CPTS_INTSTAT_RAW) ..... 1567
9-97. Time Sync Interrupt Status Register Masked Register (CPTS_INTSTAT_MASKED) ..... 1567
9-98. Time Sync Interrupt Enable Register (CPTS_INT_ENABLE) ..... 1568
9-99. Event Interrupt Pop Register (CPTS_EVENT_POP) ..... 1568
9-100. Event Low Register (CPTS_EVENT_LOW) ..... 1568
9-101. Event High Register (CPTS_EVENT_HIGH) ..... 1569
9-102. Address Lookup Engine ID/Version Register (ALE_IDVER) ..... 1569
9-103. ALE Control Register (ALE_CONTROL) ..... 1570
9-104. ALE Prescale Register (ALE_PRESCALE) ..... 1571
9-105. ALE Unknown VLAN Register (ALE_UNKNOWN_VLAN) ..... 1572
9-106. ALE Table Control Register (ALE_TBLCTL) ..... 1572
9-107. ALE Table Word 2 Register (ALE_TBLW2) ..... 1573
9-108. ALE Table Word 1 Register (ALE_TBLW1) ..... 1573
9-109. ALE Table Word 0 Register (ALE_TBLW0) ..... 1573
9-110. ALE Port Control Registers ..... 1574
9-111. CPGMAC_SL1 IDVER Register (SL1_IDVER) ..... 1575
9-112. CPGMAC_SL1 MAC Control Register (SL1_MACCONTROL) ..... 1575
9-113. CPGMAC_SL1 MAC Status Register (SL1_MACSTATUS) ..... 1577
9-114. CPGMAC_SL1 Software Reset Register (SL1_SOFT_RESET) ..... 1578
9-115. CPGMAC_SL1 RX Maximum Length Register (SL1_RX_MAXLEN) ..... 1578
9-116. CPGMAC_SL1 Backoff Random Number Generator Test Register (SL1_BOFFTEST) ..... 1579
9-117. CPGMAC_SL1 RX Pause Timer Register (SL1_RX_PAUSE) ..... 1579
9-118. CPGMAC_SL1 TX Pause Timer Register (SL1_TX_PAUSE) ..... 1580
9-119. CPGMAC_SL1 Emulation Control Register (SL1_EMCONTROL) ..... 1580
9-120. CPGMAC_SL1 RX Packet Priority to Header Priority Mapping Register (SL1_RX_Pri_Map) ..... 1581
9-121. CPGMAC_SL2 IDVER Register (SL2_IDVER) ..... 1582
9-122. CPGMAC_SL2 MAC Control Register (SL2_MACCONTROL) ..... 1582
9-123. CPGMAC_SL2 MAC Status Register (SL2_MACSTATUS) ..... 1585
9-124. CPGMAC_SL2 Software Reset Register (SL2_SOFT_RESET) ..... 1585
9-125. CPGMAC_SL2 RX Maximum Length Register (SL2_RX_MAXLEN) ..... 1586
9-126. CPGMAC_SL2 Backoff Random Number Generator Test Register (SL2_BOFFTEST) ..... 1586
9-127. CPGMAC_SL2 RX Pause Timer Register (SL2_RX_PAUSE) ..... 1587
9-128. CPGMAC_SL2 TX Pause Timer Register (SL2_TX_PAUSE) ..... 1587
9-129. CPGMAC_SL2 Emulation Control Register (SL2_EMCONTROL) ..... 1588
9-130. CPGMAC_SL2 RX Packet Priority to Header Priority Mapping Register (SL2_RX_PRI_MAP) ..... 1589


Copyright © 2011–2016, Texas Instruments Incorporated

9-131. MDIO Version Register (VERSION) ..... 1590
9-132. MDIO Control (CONTROL) Register ..... 1591
9-133. PHY Alive Status Register (ALIVE) ..... 1592
9-134. PHY Link Status Register (LINK) ..... 1592
9-135. MDIO Link Status Change Interrupt (raw value) Register (LINKINTRAW) ..... 1593
9-136. MDIO Link Status Change Interrupt (masked value) Register (LINKINTMASKED) ..... 1593
9-137. MDIO User Command Complete Interrupt (raw value) Register (USERINTRAW) ..... 1594
9-138. MDIO User Command Complete Interrupt (masked value) Register (USERINTMASKED) ..... 1594
9-139. MDIO User Command Complete Interrupt Mask Set Register (USERINTMASKSET) ..... 1595
9-140. MDIO User Command Complete Interrupt Mask Clear Register (USERINTMASKCLEAR) ..... 1595
9-141. MDIO User Access Register 0 (USERACCESS0) ..... 1596
9-142. MDIO User PHY Select Register 0 (USERPHYSEL0) ..... 1596
9-143. MDIO User Access Register 1 (USERACCESS1) ..... 1597
9-144. MDIO User PHY Select Register 1 (USERPHYSEL1) ..... 1597
9-145. Subsystem ID Version Register (IDVER) ..... 1598
9-146. Subsystem Software Reset Register (SOFT_RESET) ..... 1599
9-147. Subsystem Control Register (CONTROL) ..... 1599
9-148. Subsystem Interrupt Control Register (INT_CONTROL) ..... 1600
9-149. Subsystem Receive Threshold Interrupt Enable Register (RX_THRESH_EN) ..... 1600
9-150. Subsystem Receive Interrupt Enable Register (RX_EN) ..... 1601
9-151. Subsystem Transmit Interrupt Enable Register (TX_EN) ..... 1601
9-152. Subsystem Miscellaneous Interrupt Enable Register (MISC_EN) ..... 1601
9-153. Subsystem Receive Threshold Masked Interrupt Status Register (RX_THRESH_STAT) ..... 1602
9-154. Subsystem Receive Masked Interrupt Status Register (RX_STAT) ..... 1602
9-155. Subsystem Transmit Masked Interrupt Status Register (TX_STAT) ..... 1602
9-156. Subsystem Miscellaneous Masked Interrupt Status Register (MISC_STAT) ..... 1603
9-157. Subsystem Receive Interrupts per Millisecond Register (RX_IMAX) ..... 1603
9-158. Subsystem Transmit Interrupts per Millisecond Register (TX_IMAX) ..... 1603
9-159. RGMII Control Read Register (RGMII_CTL) ..... 1604
10-1. GPIO Block Diagram ..... 1607
10-2. Synchronous Path Block Diagram ..... 1607
10-3. Interrupt Request Generation ..... 1610
10-4. Write @ GPIO_CLEARDATAOUT Register Example ..... 1612
10-5. Write @ GPIO_SETIRQENABLEx Register Example ..... 1613
10-6. General-Purpose Interface Used as a Keyboard Interface ..... 1614
10-7. GPIO_REVISION Register ..... 1616
10-8. GPIO_SYSCONFIG Register ..... 1617
10-9. GPIO_EOI Register ..... 1618
10-10. GPIO_IRQSTATUS_RAW_0 Register ..... 1619
10-11. GPIO_IRQSTATUS_RAW_1 Register ..... 1619
10-12. GPIO_IRQSTATUS_0 Register ..... 1620
10-13. GPIO_IRQSTATUS_1 Register ..... 1620
10-14. GPIO_IRQENABLE_SET_0 Register ..... 1621
10-15. GPIO_IRQENABLE_SET_1 Register ..... 1621
10-16. GPIO_IRQENABLE_CLR_0 Register ..... 1622
10-17. GPIO_IRQENABLE_CLR_1 Register ..... 1622
10-18. GPIO_SYSSTATUS Register ..... 1623
10-19. GPIO_CTRL Register ..... 1624
10-20. GPIO_OE Register ..... 1624


Copyright © 2011–2016, Texas Instruments Incorporated

10-21. GPIO_DATAIN Register ..... 1625
10-22. GPIO_DATAOUT Register ..... 1625
10-23. GPIO_LEVELDETECT0 Register ..... 1626
10-24. GPIO_LEVELDETECT1 Register ..... 1626
10-25. GPIO_RISINGDETECT Register ..... 1627
10-26. GPIO_FALLINGDETECT Register ..... 1627
10-27. GPIO_DEBOUNCENABLE Register ..... 1628
10-28. GPIO_DEBOUNCINGTIME Register ..... 1628
10-29. GPIO_CLEARDATAOUT Register ..... 1629
10-30. GPIO_SETDATAOUT Register ..... 1629
11-1. GPMC Block Diagram ..... 1632
11-2. GPMC to 16-Bit Address/Data-Multiplexed Memory ..... 1635
11-3. GPMC to 16-Bit Nonmultiplexed Memory ..... 1636
11-4. GPMC to 8-Bit NAND Device ..... 1636
11-5. GPMC Integration ..... 1637
11-6. Chip-Select Address Mapping and Decoding Mask ..... 1642
11-7. Wait Behavior During an Asynchronous Single Read Access (GPMCFCLKDivider = 1) ..... 1645
11-8. Wait Behavior During a Synchronous Read Burst Access ..... 1647
11-9. Read to Read for an Address-Data Multiplexed Device, On Different CS, Without Bus Turnaround ($\overline{CS0}$ Attached to Fast Device) ..... 1649
11-10. Read to Read / Write for an Address-Data Multiplexed Device, On Different CS, With Bus Turnaround ..... 1649
11-11. Read to Read / Write for a Address-Data or AAD-Multiplexed Device, On Same CS, With Bus Turnaround ..... 1650
11-12. Asynchronous Single Read Operation on an Address/Data Multiplexed Device ..... 1659
11-13. Two Asynchronous Single Read Accesses on an Address/Data Multiplexed Device (32-Bit Read Split Into 2 × 16-Bit Read) ..... 1660
11-14. Asynchronous Single Write on an Address/Data-Multiplexed Device ..... 1661
11-15. Asynchronous Single-Read on an AAD-Multiplexed Device ..... 1662
11-16. Asynchronous Single Write on an AAD-Multiplexed Device ..... 1664
11-17. Synchronous Single Read (GPMCFCLKDIVIDER = 0) ..... 1666
11-18. Synchronous Single Read (GPMCFCLKDIVIDER = 1) ..... 1667
11-19. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 0) ..... 1669
11-20. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 1) ..... 1670
11-21. Synchronous Single Write on an Address/Data-Multiplexed Device ..... 1671
11-22. Synchronous Multiple Write (Burst Write) in Address/Data-Multiplexed Mode ..... 1672
11-23. Synchronous Multiple Write (Burst Write) in Address/Address/Data-Multiplexed Mode ..... 1673
11-24. Asynchronous Single Read on an Address/Data-Nonmultiplexed Device ..... 1675
11-25. Asynchronous Single Write on an Address/Data-Nonmultiplexed Device ..... 1676
11-26. Asynchronous Multiple (Page Mode) Read ..... 1677
11-27. NAND Command Latch Cycle ..... 1682
11-28. NAND Address Latch Cycle ..... 1683
11-29. NAND Data Read Cycle ..... 1684
11-30. NAND Data Write Cycle ..... 1685
11-31. Hamming Code Accumulation Algorithm (1 of 2) ..... 1689
11-32. Hamming Code Accumulation Algorithm (2 of 2) ..... 1690
11-33. ECC Computation for a 256-Byte Data Stream (Read or Write) ..... 1690
11-34. ECC Computation for a 512-Byte Data Stream (Read or Write) ..... 1691
11-35. 128 Word16 ECC Computation ..... 1692
11-36. 256 Word16 ECC Computation ..... 1692
11-37. Manual Mode Sequence and Mapping ..... 1697


Copyright © 2011–2016, Texas Instruments Incorporated

11-38. NAND Page Mapping and ECC: Per-Sector Schemes ..... 1702
11-39. NAND Page Mapping and ECC: Pooled Spare Schemes ..... 1703
11-40. NAND Page Mapping and ECC: Per-Sector Schemes, with Separate ECC ..... 1704
11-41. NAND Read Cycle Optimization Timing Description ..... 1711
11-42. Programming Model Top-Level Diagram ..... 1713
11-43. NOR Interfacing Timing Parameters Diagram ..... 1718
11-44. NAND Command Latch Cycle Timing Simplified Example ..... 1722
11-45. Synchronous NOR Single Read Simplified Example ..... 1728
11-46. Asynchronous NOR Single Write Simplified Example ..... 1730
11-47. GPMC Connection to an External NOR Flash Memory ..... 1732
11-48. Synchronous Burst Read Access (Timing Parameters in Clock Cycles) ..... 1734
11-49. Asynchronous Single Read Access (Timing Parameters in Clock Cycles) ..... 1736
11-50. Asynchronous Single Write Access (Timing Parameters in Clock Cycles) ..... 1738
11-51. GPMC_REVISION ..... 1744
11-52. GPMC_SYSCONFIG ..... 1744
11-53. GPMC_SYSSTATUS ..... 1745
11-54. GPMC_IRQSTATUS ..... 1746
11-55. GPMC_IRQENABLE ..... 1747
11-56. GPMC_TIMEOUT_CONTROL ..... 1748
11-57. GPMC_ERR_ADDRESS ..... 1748
11-58. GPMC_ERR_TYPE ..... 1749
11-59. GPMC_CONFIG ..... 1750
11-60. GPMC_STATUS ..... 1751
11-61. GPMC_CONFIG1_i ..... 1752
11-62. GPMC_CONFIG2_i ..... 1755
11-63. GPMC_CONFIG3_i ..... 1756
11-64. GPMC_CONFIG4_i ..... 1758
11-65. GPMC_CONFIG5_i ..... 1760
11-66. GPMC_CONFIG6_i ..... 1761
11-67. GPMC_CONFIG7_i ..... 1762
11-68. GPMC_NAND_COMMAND_i ..... 1763
11-69. GPMC_NAND_ADDRESS_i ..... 1763
11-70. GPMC_NAND_DATA_i ..... 1763
11-71. GPMC_PREFETCH_CONFIG1 ..... 1764
11-72. GPMC_PREFETCH_CONFIG2 ..... 1766
11-73. GPMC_PREFETCH_CONTROL ..... 1766
11-74. GPMC_PREFETCH_STATUS ..... 1767
11-75. GPMC_ECC_CONFIG ..... 1768
11-76. GPMC_ECC_CONTROL ..... 1769
11-77. GPMC_ECC_SIZE_CONFIG ..... 1770
11-78. GPMC_ECCj_RESULT ..... 1772
11-79. GPMC_BCH_RESULT0_i ..... 1773
11-80. GPMC_BCH_RESULT1_i ..... 1773
11-81. GPMC_BCH_RESULT2_i ..... 1773
11-82. GPMC_BCH_RESULT3_i ..... 1774
11-83. GPMC_BCH_RESULT4_i ..... 1774
11-84. GPMC_BCH_RESULT5_i ..... 1775
11-85. GPMC_BCH_RESULT6_i ..... 1775
11-86. GPMC_BCH_SWDATA ..... 1775


Copyright © 2011–2016, Texas Instruments Incorporated

12-1. HDVPSS Detailed Block Diagram ..... 1782
12-2. Primary Input Path (PRI) Detailed Block Diagram ..... 1783
12-3. Auxiliary Input Path Detailed Block Diagram ..... 1784
12-4. VIP Subsystem Detailed Block Diagram ..... 1785
12-5. HDVPSS Dual Display (PIP) Data Flow ..... 1790
12-6. HDVPSS Tri Display Data Flow 1 ..... 1792
12-7. HDVPSS Tri Display Data Flow 2 ..... 1793
12-8. HDVPSS Tri Display Dual Transcode ..... 1794
12-9. HDVPSS Tri Display Dual Transcode with Video Capture ..... 1796
12-10. HDVPSS Tri Display with Video Capture ..... 1798
12-11. HDVPSS Tri Display with Video Capture and Noise Filter ..... 1800
12-12. MMR Functionality ..... 1802
12-13. Module to HDVPSS Interrupt Mapping Level 1 ..... 1804
12-14. Module to HDVPSS Interrupt Mapping Level 2 ..... 1805
12-15. Video Encoder Clock/Control Diagram ..... 1806
12-16. 4:2:0 YCrCb Color Space with Chroma Left-aligned ..... 1808
12-17. Catmull-Rom Filter Definition ..... 1809
12-18. Definition of 'X' ..... 1809
12-19. Anchor Pixels ..... 1809
12-20. 4:2:0 Interlaced Scan ..... 1810
12-21. Ideal 4:2:2 Chroma Upsampling for Interlaced Scan ..... 1810
12-22. Matrix Format ..... 1813
12-23. Conversion from RGB to YCbCr ..... 1814
12-24. Conversion from YCbCr to RGB ..... 1814
12-25. Conversion from RGB to YCbCr ..... 1814
12-26. Conversion from YCbCr to RGB ..... 1814
12-27. Conversion from RGB to YCbCr ..... 1816
12-28. Conversion from YCbCr to RGB ..... 1816
12-29. Conversion from RGB to YCbCr ..... 1817
12-30. Conversion from YCbCr to RGB ..... 1817
12-31. COMP Module Block Diagram ..... 1820
12-32. Display Width and Height Parameters ..... 1821
12-33. Blender Diagram ..... 1823
12-34. Blending and Reordering ..... 1824
12-35. Alpha Blender Block Diagram ..... 1825
12-36. CIG Block Diagram ..... 1827
12-37. GRPX Functional Block Diagram ..... 1830
12-38. GRPX Output ..... 1830
12-39. Region Display Position and Gap Requirement Illustration ..... 1834
12-40. VENC Block Diagram ..... 1842
12-41. Example of Multiple VENC Synchronization ..... 1843
12-42. Video Timing ..... 1843
12-43. Input I/F Horizontal Timing ..... 1844
12-44. Input I/F Vertical Timing ..... 1845
12-45. Input Data Timing ..... 1845
12-46. Interrupt Horizontal Timing ..... 1846
12-47. Interrupt Vertical Timing ..... 1846
12-48. TVDETGP Horizontal Timing ..... 1847
12-49. TVDETGP Vertical Timing ..... 1847


Copyright © 2011–2016, Texas Instruments Incorporated

12-50. SDTV Horizontal Timing ..... 1848
12-51. 525i Vertical Timing ..... 1849
12-52. 625i Vertical Timing ..... 1849
12-53. SDTV DAC Video Output Pipeline Delay (PXLR = 0) ..... 1850
12-54. SDTV DAC Video Output Pipeline Delay (PXLR = 1) ..... 1850
12-55. 262p (Non-Interlaced NTSC) Vertical Timing ..... 1851
12-56. 312p (Non-Interlaced PAL) Vertical Timing ..... 1851
12-57. Horizontal Blanking Shaping ..... 1852
12-58. Slave Mode Horizontal Timing ..... 1852
12-59. Interlaced Slave Vertical Timing (FMD = 0,1) ..... 1853
12-60. Interlaced Slave Vertical Timing (FMD = 2) ..... 1853
12-61. Interlaced Slave Vertical Timing (FMD = 3) ..... 1854
12-62. Field Detection Mode ..... 1855
12-63. Field Detection by VD Phase (FMD=3) ..... 1855
12-64. CVBS Video Amplitude ..... 1864
12-65. Limiter Configuration ..... 1867
12-66. Sub-carrier Parameter Registers ..... 1868
12-67. WSS_Data Register Usage ..... 1869
12-68. WSS_Data Register Usage ..... 1869
12-69. VBI I/F Horizontal Timing ..... 1871
12-70. VBI I/F Vertical Timing ..... 1871
12-71. DAC I/F Logic ..... 1874
12-72. HD VENC Block Diagram ..... 1875
12-73. Vertical Sync Signals for OSD ..... 1877
12-74. Horizontal Sync-related Signal for OSD ..... 1877
12-75. Video Data Interface Between OSD and Encoder ..... 1878
12-76. OSD Interface in Progressive Mode ..... 1879
12-77. OSD Interface in Interlace Mode ..... 1880
12-78. Progressive Configuration of DVO ..... 1882
12-79. Interlace Configuration of DVO ..... 1883
12-80. Typical VBI Data Package ..... 1884
12-81. NTSC Analog Video Waveform for One Horizontal Line ..... 1888
12-82. Digitized Video ..... 1888
12-83. Code Word Embedded Video Format ..... 1889
12-84. Digitized Video with F, V, and H Flags in EAV/SAV ..... 1889
12-85. Planar Buffer Storage Description ..... 1890
12-86. 8b Interface Discrete Sync Pixel Multiplexing ..... 1891
12-87. 16b Interface Discrete Sync Pixel Multiplexing ..... 1891
12-88. 24b Interface RGB or YUV444 Discrete Sync ..... 1892
12-89. 24 Device Pins for Ports A and B ..... 1892
12-90. Discrete Sync Signals ..... 1893
12-91. Type 1, First Horizontal Blanking Pixel ..... 1894
12-92. Type 1, First Vertical Ancillary Data Pixel ..... 1894
12-93. Type 1, Horizontal Blanking in Video Region ..... 1895
12-94. Type 1, First Video Pixel ..... 1895
12-95. 4-Pin Reduced ACTVID Signaling with Vertical Ancillary Data ..... 1896
12-96. 4-Pin Reduced ACTVID Signaling with No Vertical Ancillary Data ..... 1896
12-97. 4-Pin Reduced HSYNC Signaling with Vertical Ancillary Data ..... 1896
12-98. VSYNC Pre and Post Window ..... 1897


Copyright © 2011–2016, Texas Instruments Incorporated

12-99. VSYNC Equivalence When Using Transition Window ..... 1898
12-100. FID Registering When Using HSYNC ..... 1899
12-101. FID Registering When Using ACTVID ..... 1899
12-102. Field ID Determination By VSYNC Skew ..... 1900
12-103. Example of 525-line FID Determination By VSYNC Skew ..... 1901
12-104. Horizontal Ancillary Data Packing When HSYNC Used as Sync Signal ..... 1902
12-105. Interlaced Field Vertical Blanking Ancillary Data Storage ..... 1902
12-106. Progressive Frame Vertical Blanking Ancillary Data Storage ..... 1903
12-107. Embedded Sync Data Entry ..... 1903
12-108. Code Word Format Example Followed by Video Data ..... 1904
12-109. Embedded Sync Packing ..... 1906
12-110. RGB Frame Storage ..... 1907
12-111. 2-Way Multiplexing ..... 1908
12-112. Example of 4-Way Multiplexing ..... 1909
12-113. Example of Line Multiplexing ..... 1909
12-114. 8-bit Line Mux Interface ..... 1910
12-115. 16-bit Line Mux Interface ..... 1910
12-116. BOP/EOP Definition of a Period ..... 1911
12-117. Channel ID Inserted Into Horizontal Blanking ..... 1914
12-118. Vertical Ancillary Data Cropping Region ..... 1918
12-119. Active Video Region ..... 1918
12-120. Discrete Sync Interface Signals ..... 1919
12-121. vsync and hblank Input Signals ..... 1920
12-122. vsync and hsync Input Signals ..... 1920
12-123. vsync and actvid Input Signals ..... 1921
12-124. vblank and hsync Input Signals ..... 1921
12-125. vblank and hblank Input Signals ..... 1922
12-126. vblank and actvid Input Signals ..... 1922
12-127. Line and Pixel Capture, vblank and hsync ..... 1923
12-128. Line and Pixel Capture, hsync ..... 1923
12-129. Line and Pixel Capture, actvid ..... 1924
12-130. Block Diagram of Motion-Adaptive Deinterlacer ..... 1926
12-131. VPDMA Transfer Ports ..... 1929
12-132. Auxiliary Data ..... 1930
12-133. Noise Filter Architecture Block Diagram ..... 1933
12-134. Top-left Corner of a Tile After the Boundary Replication ..... 1933
12-135. Motion versus Blending Factor Function ..... 1934
12-136. $\alpha_0$ versus *totalFrame_noise* function ..... 1935
12-137. Hardware Implementation Block Diagram ..... 1935
12-138. Incomplete Boundary Tile Data Masking ..... 1936
12-139. High Level Block Diagram ..... 1938
12-140. SC Block Diagram ..... 1939
12-141. Input Image Trimming ..... 1939
12-142. Filter Implementation and Parameter Description ..... 1940
12-143. Peaking Filter at fs/4 ..... 1941
12-144. Vertical Scaler Block Diagram ..... 1941
12-145. Mixed 2-tap and 5-tap vertical interpolation ..... 1942
12-146. Vertical Polyphase Scaler Block Diagram (SC) ..... 1942
12-147. Horizontal Scaler Block Diagram ..... 1944


Copyright © 2011–2016, Texas Instruments Incorporated

12-148. Polyphase Filtering Example ..... 1945
12-149. Non-linear Scaling Example ..... 1946
12-150. SRAM Layout for 7tap Coefficient ..... 1949
12-151. SRAM Layout for 5tap Coefficient ..... 1949
12-152. VPI Control I/F Coef Data Format (7tap) ..... 1950
12-153. VPI Control I/F Coef Data Format (5tap) ..... 1950
12-154. VPI Control I/F Coef Data Format (3tap) ..... 1950
12-155. VPI Control I/F Memory Map (Write) ..... 1950
12-156. VPI Control I/F Memory Map (Read) ..... 1951
12-157. Main and Auxiliary Layers Over Background ..... 1974
12-158. Overlap Region ..... 1975
12-159. Cropping of Input Layers ..... 1975
12-160. Single Plane Output ..... 1976
12-161. Main Layer Only ..... 1977
12-162. Aux Layer Only ..... 1977
12-163. No Layer Enabled ..... 1977
12-164. Conditions Where Valid Configurations May Not Work in a System ..... 1978
12-165. Example ..... 1979
12-166. Inbound Data Transfer Descriptor Format ..... 1983
12-167. Outbound Data Transfer Descriptor Format ..... 1983
12-168. HDVPSS Pressure and Priority Settings ..... 1988
12-169. Block Diagram ..... 2000
12-170. Example Graphics Output ..... 2034
12-171. Y 4:4:4 (Data Type 0) ..... 2036
12-172. Y 4:2:2 (Data Type 1) ..... 2037
12-173. Y 4:2:0 (Data Type 2) ..... 2037
12-174. C 4:4:4 (Data Type 4) ..... 2038
12-175. C 4:2:2 (Data Type 5) ..... 2039
12-176. C 4:2:0 (Data Type 6) ..... 2039
12-177. YC 4:2:2 (Data Type 7) ..... 2040
12-178. YC 4:4:4 (Data Type 8) ..... 2041
12-179. CY 4:2:2 (Data Type 23h) ..... 2042
12-180. Cb 4:4:4 (Data Type 14h) ..... 2043
12-181. Cb 4:2:2 (Data Type 15h) ..... 2043
12-182. Cb 4:2:0 (Data Type 16h) ..... 2044
12-183. CbY 4:2:2 (Data Type 17h) ..... 2044
12-184. YC 4:2:2 (Data Type 27h) ..... 2045
12-185. YC 4:2:2 (Data Type 37h) ..... 2045
12-186. RGB16-565 (Data Type 0) ..... 2046
12-187. ARGB-1555 (Data Type 1) ..... 2047
12-188. ARGB-4444 (Data Type 2) ..... 2047
12-189. RGBA-5551 (Data Type 3) ..... 2048
12-190. RGBA-4444 (Data Type 4) ..... 2048
12-191. ARGB24-6666 (Data Type 5) ..... 2049
12-192. RGB24-888 (Data Type 6) ..... 2049
12-193. ARGB32-8888 (Data Type 7) ..... 2050
12-194. RGBA24-6666 (Data Type 8) ..... 2051
12-195. RGBA32-8888 (Data Type 9) ..... 2052
12-196. Bitmap-8 (Data Type 20h) ..... 2053


Copyright © 2011–2016, Texas Instruments Incorporated

12-197. Bitmap-4 Lower (Data Type 22h) ..... 2053
12-198. Bitmap-4 Upper (Data Type 23h) ..... 2054
12-199. Bitmap-2 Offset0 (Data Type 24h) ..... 2054
12-200. Bitmap-2 Offset1 (Data Type 25h) ..... 2055
12-201. Bitmap-2 Offset2 (Data Type 26h) ..... 2055
12-202. Bitmap-2 Offset3 (Data Type 27h) ..... 2056
12-203. Bitmap-1 Offset0 (Data Type 28h) ..... 2056
12-204. Bitmap-1 Offset1 (Data Type 29h) ..... 2057
12-205. Bitmap-1 Offset2 (Data Type 2Ah) ..... 2057
12-206. Bitmap-1 Offset3 (Data Type 2Bh) ..... 2058
12-207. Bitmap-1 Offset4 (Data Type 2Ch) ..... 2058
12-208. Bitmap-1 Offset5 (Data Type 2Dh) ..... 2059
12-209. Bitmap-1 Offset6 (Data Type 2Eh) ..... 2059
12-210. Bitmap-1 Offset7 (Data Type 2Fh) ..... 2060
12-211. CHR_US_reg0 Register ..... 2064
12-212. CHR_US_reg1 Register ..... 2065
12-213. CHR_US_reg2 Register ..... 2066
12-214. CHR_US_reg3 Register ..... 2067
12-215. CHR_US_reg4 Register ..... 2068
12-216. CHR_US_reg5 Register ..... 2069
12-217. CHR_US_reg6 Register ..... 2070
12-218. CHR_US_reg7 Register ..... 2071
12-219. CIG_reg1 Register ..... 2074
12-220. CIG_reg2 Register ..... 2075
12-221. CIG_reg3 Register ..... 2076
12-222. CIG_reg4 Register ..... 2077
12-223. CIG_reg5 Register ..... 2078
12-224. CIG_reg6 Register ..... 2079
12-225. CIG_reg7 Register ..... 2080
12-226. CIG_reg8 Register ..... 2081
12-227. CIG_reg9 Register ..... 2082
12-228. CIG_reg10 Register ..... 2083
12-229. COMP_hdmi_settings Register ..... 2086
12-230. COMP_dvo2_settings Register ..... 2087
12-231. COMP_sd_settings Register ..... 2088
12-232. COMP_back_color_settings Register ..... 2089
12-233. CSC_csc00 Register ..... 2091
12-234. CSC_csc01 Register ..... 2092
12-235. CSC_csc02 Register ..... 2093
12-236. CSC_csc03 Register ..... 2094
12-237. CSC_csc04 Register ..... 2095
12-238. CSC_csc05 Register ..... 2096
12-239. dei_reg0 Register ..... 2098
12-240. dei_reg1 Register ..... 2099
12-241. dei_reg2 Register ..... 2100
12-242. dei_reg3 Register ..... 2101
12-243. dei_reg4 Register ..... 2102
12-244. dei_reg5 Register ..... 2103
12-245. dei_reg6 Register ..... 2104


Copyright © 2011–2016, Texas Instruments Incorporated

12-246. dei_reg7 Register ..... 2105
12-247. dei_reg8 Register ..... 2106
12-248. dei_reg9 Register ..... 2107
12-249. dei_reg10 Register ..... 2108
12-250. dei_reg11 Register ..... 2109
12-251. dei_reg12 Register ..... 2110
12-252. dei_reg13 Register ..... 2111
12-253. dei_reg14 Register ..... 2112
12-254. pid Register ..... 2116
12-255. sysconfig Register ..... 2117
12-256. intc_intr0_status_raw0 Register ..... 2118
12-257. intc_intr0_status_raw1 Register ..... 2120
12-258. intc_intr0_status_ena0 Register ..... 2122
12-259. intc_intr0_status_ena1 Register ..... 2124
12-260. intc_intr0_ena_set0 Register ..... 2126
12-261. intc_intr0_ena_set1 Register ..... 2128
12-262. intc_intr0_ena_clr0 Register ..... 2130
12-263. intc_intr0_ena_clr1 Register ..... 2132
12-264. intc_intr1_status_raw0 Register ..... 2134
12-265. intc_intr1_status_raw1 Register ..... 2136
12-266. intc_intr1_status_ena0 Register ..... 2138
12-267. intc_intr1_status_ena1 Register ..... 2140
12-268. intc_intr1_ena_set0 Register ..... 2142
12-269. intc_intr1_ena_set1 Register ..... 2144
12-270. intc_intr1_ena_clr0 Register ..... 2146
12-271. intc_intr1_ena_clr1 Register ..... 2148
12-272. intc_intr2_status_raw0 Register ..... 2150
12-273. intc_intr2_status_raw1 Register ..... 2152
12-274. intc_intr2_status_ena0 Register ..... 2154
12-275. intc_intr2_status_ena1 Register ..... 2156
12-276. intc_intr2_ena_set0 Register ..... 2158
12-277. intc_intr2_ena_set1 Register ..... 2160
12-278. intc_intr2_ena_clr0 Register ..... 2162
12-279. intc_intr2_ena_clr1 Register ..... 2164
12-280. intc_intr3_status_raw0 Register ..... 2166
12-281. intc_intr3_status_raw1 Register ..... 2168
12-282. intc_intr3_status_ena0 Register ..... 2170
12-283. intc_intr3_status_ena1 Register ..... 2172
12-284. intc_intr3_ena_set0 Register ..... 2174
12-285. intc_intr3_ena_set1 Register ..... 2176
12-286. intc_intr3_ena_clr0 Register ..... 2178
12-287. intc_intr3_ena_clr1 Register ..... 2180
12-288. intc_eoi Register ..... 2182
12-289. clkc_clken Register ..... 2183
12-290. clkc_dps Register ..... 2185
12-291. clkc_vip1dps Register ..... 2187
12-292. clkc_vip2dps Register ..... 2190
12-293. clkc_venc_ena Register ..... 2193
12-294. clkc_range_map Register ..... 2194


Copyright © 2011–2016, Texas Instruments Incorporated

12-295. clkc_underflow Register ..... 2195
12-296. VPDMA_pid Register ..... 2199
12-297. VPDMA_list_addr Register ..... 2200
12-298. VPDMA_list_attr Register ..... 2201
12-299. VPDMA_list_stat_sync Register ..... 2202
12-300. VPDMA_vpi_ctl_address Register ..... 2204
12-301. VPDMA_vpi_ctl_data Register ..... 2205
12-302. VPDMA_bg_rgb Register ..... 2206
12-303. VPDMA_bg_yuv Register ..... 2207
12-304. VPDMA_descriptor_top Register ..... 2208
12-305. VPDMA_descriptor_bottom Register ..... 2209
12-306. VPDMA_current_descriptor Register ..... 2210
12-307. VPDMA_descriptor_status_control Register ..... 2211
12-308. VPDMA_int0_channel0_int_stat Register ..... 2212
12-309. VPDMA_int0_channel0_int_mask Register ..... 2215
12-310. VPDMA_int0_channel1_int_stat Register ..... 2217
12-311. VPDMA_int0_channel1_int_mask Register ..... 2221
12-312. VPDMA_int0_channel2_int_stat Register ..... 2224
12-313. VPDMA_int0_channel2_int_mask Register ..... 2228
12-314. VPDMA_int0_channel3_int_stat Register ..... 2231
12-315. VPDMA_int0_channel3_int_mask Register ..... 2235
12-316. VPDMA_int0_channel4_int_stat Register ..... 2238
12-317. VPDMA_int0_channel4_int_mask Register ..... 2242
12-318. VPDMA_int0_channel5_int_stat Register ..... 2245
12-319. VPDMA_int0_channel5_int_mask Register ..... 2249
12-320. VPDMA_int0_channel6_int_stat Register ..... 2252
12-321. VPDMA_int0_channel6_int_mask Register ..... 2253
12-322. VPDMA_int0_client0_int_stat Register ..... 2254
12-323. VPDMA_int0_client0_int_mask Register ..... 2257
12-324. VPDMA_int0_client1_int_stat Register ..... 2259
12-325. VPDMA_int0_client1_int_mask Register ..... 2263
12-326. VPDMA_int0_list0_int_stat Register ..... 2266
12-327. VPDMA_int0_list0_int_mask Register ..... 2269
12-328. VPDMA_int1_channel0_int_stat Register ..... 2272
12-329. VPDMA_int1_channel0_int_mask Register ..... 2275
12-330. VPDMA_int1_channel1_int_stat Register ..... 2277
12-331. VPDMA_int1_channel1_int_mask Register ..... 2281
12-332. VPDMA_int1_channel2_int_stat Register ..... 2284
12-333. VPDMA_int1_channel2_int_mask Register ..... 2288
12-334. VPDMA_int1_channel3_int_stat Register ..... 2291
12-335. VPDMA_int1_channel3_int_mask Register ..... 2295
12-336. VPDMA_int1_channel4_int_stat Register ..... 2298
12-337. VPDMA_int1_channel4_int_mask Register ..... 2302
12-338. VPDMA_int1_channel5_int_stat Register ..... 2305
12-339. VPDMA_int1_channel5_int_mask Register ..... 2309
12-340. VPDMA_int1_channel6_int_stat Register ..... 2312
12-341. VPDMA_int1_channel6_int_mask Register ..... 2313
12-342. VPDMA_int1_client0_int_stat Register ..... 2314
12-343. VPDMA_int1_client0_int_mask Register ..... 2317


Copyright © 2011–2016, Texas Instruments Incorporated

12-344. VPDMA_int1_client1_int_stat Register ..... 2319
12-345. VPDMA_int1_client1_int_mask Register ..... 2323
12-346. VPDMA_int1_list0_int_stat Register ..... 2326
12-347. VPDMA_int1_list0_int_mask Register ..... 2329
12-348. VPDMA_int2_channel0_int_stat Register ..... 2332
12-349. VPDMA_int2_channel0_int_mask Register ..... 2335
12-350. VPDMA_int2_channel1_int_stat Register ..... 2337
12-351. VPDMA_int2_channel1_int_mask Register ..... 2341
12-352. VPDMA_int2_channel2_int_stat Register ..... 2344
12-353. VPDMA_int2_channel2_int_mask Register ..... 2348
12-354. VPDMA_int2_channel3_int_stat Register ..... 2351
12-355. VPDMA_int2_channel3_int_mask Register ..... 2355
12-356. VPDMA_int2_channel4_int_stat Register ..... 2358
12-357. VPDMA_int2_channel4_int_mask Register ..... 2362
12-358. VPDMA_int2_channel5_int_stat Register ..... 2365
12-359. VPDMA_int2_channel5_int_mask Register ..... 2369
12-360. VPDMA_int2_channel6_int_stat Register ..... 2372
12-361. VPDMA_int2_channel6_int_mask Register ..... 2373
12-362. VPDMA_int2_client0_int_stat Register ..... 2374
12-363. VPDMA_int2_client0_int_mask Register ..... 2377
12-364. VPDMA_int2_client1_int_stat Register ..... 2379
12-365. VPDMA_int2_client1_int_mask Register ..... 2383
12-366. VPDMA_int2_list0_int_stat Register ..... 2386
12-367. VPDMA_int2_list0_int_mask Register ..... 2389
12-368. VPDMA_int3_channel0_int_stat Register ..... 2392
12-369. VPDMA_int3_channel0_int_mask Register ..... 2395
12-370. VPDMA_int3_channel1_int_stat Register ..... 2397
12-371. VPDMA_int3_channel1_int_mask Register ..... 2401
12-372. VPDMA_int3_channel2_int_stat Register ..... 2404
12-373. VPDMA_int3_channel2_int_mask Register ..... 2408
12-374. VPDMA_int3_channel3_int_stat Register ..... 2411
12-375. VPDMA_int3_channel3_int_mask Register ..... 2415
12-376. VPDMA_int3_channel4_int_stat Register ..... 2418
12-377. VPDMA_int3_channel4_int_mask Register ..... 2422
12-378. VPDMA_int3_channel5_int_stat Register ..... 2425
12-379. VPDMA_int3_channel5_int_mask Register ..... 2429
12-380. VPDMA_int3_channel6_int_stat Register ..... 2432
12-381. VPDMA_int3_channel6_int_mask Register ..... 2433
12-382. VPDMA_int3_client0_int_stat Register ..... 2434
12-383. VPDMA_int3_client0_int_mask Register ..... 2437
12-384. VPDMA_int3_client1_int_stat Register ..... 2439
12-385. VPDMA_int3_client1_int_mask Register ..... 2443
12-386. VPDMA_int3_list0_int_stat Register ..... 2446
12-387. VPDMA_int3_list0_int_mask Register ..... 2449
12-388. VPDMA_dei_hq_1_chroma_cstat Register ..... 2452
12-389. VPDMA_dei_hq_1_luma_cstat Register ..... 2453
12-390. VPDMA_dei_hq_2_luma_cstat Register ..... 2454
12-391. VPDMA_dei_hq_2_chroma_cstat Register ..... 2455
12-392. VPDMA_dei_hq_3_luma_cstat Register ..... 2456


Copyright © 2011–2016, Texas Instruments Incorporated

12-393. VPDMA_dei_hq_3_chroma_cstat Register ..... 2457
12-394. VPDMA_dei_hq_mv_in_cstat Register ..... 2458
12-395. VPDMA_dei_hq_mv_out_cstat Register ..... 2459
12-396. VPDMA_dei_sc_out_cstat Register ..... 2460
12-397. VPDMA_pip_wrbk_cstat Register ..... 2461
12-398. VPDMA_sc_in_chroma_cstat Register ..... 2462
12-399. VPDMA_sc_in_luma_cstat Register ..... 2463
12-400. VPDMA_sc_out_cstat Register ..... 2464
12-401. VPDMA_comp_wrbk_cstat Register ..... 2465
12-402. VPDMA_grpx1_data_cstat Register ..... 2466
12-403. VPDMA_grpx2_data_cstat Register ..... 2467
12-404. VPDMA_grpx3_data_cstat Register ..... 2468
12-405. VPDMA_vip1_lo_y_cstat Register ..... 2469
12-406. VPDMA_vip1_lo_uv_cstat Register ..... 2470
12-407. VPDMA_vip1_up_y_cstat Register ..... 2471
12-408. VPDMA_vip1_up_uv_cstat Register ..... 2472
12-409. VPDMA_vip2_lo_y_cstat Register ..... 2473
12-410. VPDMA_vip2_lo_uv_cstat Register ..... 2474
12-411. VPDMA_vip2_up_y_cstat Register ..... 2475
12-412. VPDMA_vip2_up_uv_cstat Register ..... 2476
12-413. VPDMA_grpx1_st_cstat Register ..... 2477
12-414. VPDMA_grpx2_st_cstat Register ..... 2478
12-415. VPDMA_grpx3_st_cstat Register ..... 2479
12-416. VPDMA_nf_422_in_cstat Register ..... 2480
12-417. VPDMA_nf_420_y_in_cstat Register ..... 2481
12-418. VPDMA_nf_420_uv_in_cstat Register ..... 2482
12-419. VPDMA_nf_420_y_out_cstat Register ..... 2483
12-420. VPDMA_nf_420_uv_out_cstat Register ..... 2484
12-421. VPDMA_vbi_sdvenc_cstat Register ..... 2485
12-422. VPDMA_vpi_ctl_cstat Register ..... 2486
12-423. VPDMA_hdmi_wrbk_out_cstat Register ..... 2487
12-424. VPDMA_trans1_chroma_cstat Register ..... 2488
12-425. VPDMA_trans1_luma_cstat Register ..... 2489
12-426. VPDMA_trans2_chroma_cstat Register ..... 2490
12-427. VPDMA_trans2_luma_cstat Register ..... 2491
12-428. VPDMA_vip1_anc_a_cstat Register ..... 2492
12-429. VPDMA_vip1_anc_b_cstat Register ..... 2493
12-430. VPDMA_vip2_anc_a_cstat Register ..... 2494
12-431. VPDMA_vip2_anc_b_cstat Register ..... 2495
12-432. HD_VENC_D_cfg0 Register ..... 2497
12-433. HD_VENC_D_cfg1 Register ..... 2499
12-434. HD_VENC_D_cfg2 Register ..... 2500
12-435. HD_VENC_D_cfg3 Register ..... 2501
12-436. HD_VENC_D_cfg4 Register ..... 2502
12-437. HD_VENC_D_cfg5 Register ..... 2503
12-438. HD_VENC_D_cfg6 Register ..... 2504
12-439. HD_VENC_D_cfg7 Register ..... 2505
12-440. HD_VENC_D_cfg8 Register ..... 2506
12-441. HD_VENC_D_cfg9 Register ..... 2507


Copyright © 2011–2016, Texas Instruments Incorporated

12-442. HD_VENC_D_cfg10 Register ..... 2508
12-443. HD_VENC_D_cfg11 Register ..... 2509
12-444. HD_VENC_D_cfg12 Register ..... 2510
12-445. HD_VENC_D_cfg13 Register ..... 2511
12-446. HD_VENC_D_cfg14 Register ..... 2512
12-447. HD_VENC_D_cfg15 Register ..... 2513
12-448. HD_VENC_D_cfg16 Register ..... 2514
12-449. HD_VENC_D_cfg17 Register ..... 2515
12-450. HD_VENC_D_cfg18 Register ..... 2516
12-451. HD_VENC_D_cfg19 Register ..... 2517
12-452. HD_VENC_D_cfg20 Register ..... 2518
12-453. HD_VENC_D_cfg21 Register ..... 2519
12-454. HD_VENC_D_cfg22 Register ..... 2520
12-455. HD_VENC_D_cfg23 Register ..... 2521
12-456. HD_VENC_D_cfg24 Register ..... 2522
12-457. HD_VENC_D_cfg25 Register ..... 2523
12-458. HD_VEND_D_GAMMA_LUT Register ..... 2524
12-459. nf_reg0 Register ..... 2526
12-460. nf_reg1 Register ..... 2527
12-461. nf_reg2 Register ..... 2528
12-462. nf_reg3 Register ..... 2529
12-463. nf_reg4 Register ..... 2530
12-464. nf_reg5 Register ..... 2531
12-465. nf_reg6 Register ..... 2532
12-466. nf_reg7 Register ..... 2533
12-467. nf_reg8 Register ..... 2534
12-468. nf_reg9 Register ..... 2535
12-469. SC_M_cfg_sc0 Register ..... 2537
12-470. SC_M_cfg_sc1 Register ..... 2539
12-471. SC_M_cfg_sc2 Register ..... 2540
12-472. SC_M_cfg_sc3 Register ..... 2541
12-473. SC_M_cfg_sc4 Register ..... 2542
12-474. SC_M_cfg_sc5 Register ..... 2543
12-475. SC_M_cfg_sc6 Register ..... 2544
12-476. SC_M_cfg_sc8 Register ..... 2545
12-477. SC_M_cfg_sc9 Register ..... 2546
12-478. SC_M_cfg_sc10 Register ..... 2547
12-479. SC_M_cfg_sc11 Register ..... 2548
12-480. SC_M_cfg_sc12 Register ..... 2549
12-481. SC_M_cfg_sc13 Register ..... 2550
12-482. SC_M_cfg_sc17 Register ..... 2551
12-483. SC_M_cfg_sc18 Register ..... 2552
12-484. SC_M_cfg_sc19 Register ..... 2553
12-485. SC_M_cfg_sc20 Register ..... 2554
12-486. SC_M_cfg_sc21 Register ..... 2555
12-487. SC_M_cfg_sc22 Register ..... 2556
12-488. SC_M_cfg_sc23 Register ..... 2557
12-489. SC_M_cfg_sc24 Register ..... 2558
12-490. SC_M_cfg_sc25 Register ..... 2559


Copyright © 2011–2016, Texas Instruments Incorporated

12-491. SD_VENC_pid Register ..... 2562
12-492. SD_VENC_vmod Register ..... 2563
12-493. SD_VENC_slave Register ..... 2564
12-494. SD_VENC_size Register ..... 2565
12-495. SD_VENC_pol Register ..... 2566
12-496. SD_VENC_dtvs0 Register ..... 2567
12-497. SD_VENC_dtvs1 Register ..... 2568
12-498. SD_VENC_dtvs2 Register ..... 2569
12-499. SD_VENC_dtvs3 Register ..... 2570
12-500. SD_VENC_dtvs4 Register ..... 2571
12-501. SD_VENC_dtvs5 Register ..... 2572
12-502. SD_VENC_dtvs6 Register ..... 2573
12-503. SD_VENC_dtvs7 Register ..... 2574
12-504. SD_VENC_tvdetgp0 Register ..... 2575
12-505. SD_VENC_tvdetgp1 Register ..... 2576
12-506. SD_VENC_irq0 Register ..... 2577
12-507. SD_VENC_estat Register ..... 2578
12-508. SD_VENC_ectl Register ..... 2579
12-509. SD_VENC_etmg0 Register ..... 2580
12-510. SD_VENC_etmg1 Register ..... 2581
12-511. SD_VENC_etmg2 Register ..... 2582
12-512. SD_VENC_etmg3 Register ..... 2583
12-513. SD_VENC_etmg4 Register ..... 2584
12-514. SD_VENC_cvbs0 Register ..... 2585
12-515. SD_VENC_cvbs1 Register ..... 2586
12-516. SD_VENC_ccsc0 Register ..... 2587
12-517. SD_VENC_ccsc1 Register ..... 2588
12-518. SD_VENC_ccsc2 Register ..... 2589
12-519. SD_VENC_ccsc3 Register ..... 2590
12-520. SD_VENC_ccsc4 Register ..... 2591
12-521. SD_VENC_ccsc5 Register ..... 2592
12-522. SD_VENC_ccsc6 Register ..... 2593
12-523. SD_VENC_ccsc7 Register ..... 2594
12-524. SD_VENC_ccsc8 Register ..... 2595
12-525. SD_VENC_cygclp Register ..... 2596
12-526. SD_VENC_cubclp Register ..... 2597
12-527. SD_VENC_cvrclp Register ..... 2598
12-528. SD_VENC_ylpf0 Register ..... 2599
12-529. SD_VENC_ylpf1 Register ..... 2600
12-530. SD_VENC_clpf0 Register ..... 2601
12-531. SD_VENC_clpf1 Register ..... 2602
12-532. SD_VENC_upf0 Register ..... 2603
12-533. SD_VENC_upf1 Register ..... 2604
12-534. SD_VENC_l21ctl Register ..... 2605
12-535. SD_VENC_l21do Register ..... 2606
12-536. SD_VENC_l21de Register ..... 2607
12-537. SD_VENC_wss Register ..... 2608
12-538. SD_VENC_scctl0 Register ..... 2609
12-539. SD_VENC_scctl1 Register ..... 2610


Copyright © 2011–2016, Texas Instruments Incorporated

12-540. SD_VENC_dacsel Register ..... 2611
12-541. SD_VENC_dupf0 Register ..... 2612
12-542. SD_VENC_dupf1 Register ..... 2613
12-543. SD_VENC_dactst Register ..... 2614
12-544. SD_VENC_vtest Register ..... 2615
12-545. VCOMP_reg0 Register ..... 2617
12-546. VCOMP_reg1 Register ..... 2618
12-547. VCOMP_reg2 Register ..... 2619
12-548. VCOMP_reg3 Register ..... 2620
12-549. VCOMP_reg4 Register ..... 2621
12-550. VCOMP_reg5 Register ..... 2622
12-551. VCOMP_reg6 Register ..... 2623
12-552. VCOMP_reg7 Register ..... 2624
12-553. VCOMP_reg8 Register ..... 2625
12-554. VCOMP_reg9 Register ..... 2626
12-555. VCOMP_reg10 Register ..... 2627
12-556. VCOMP_reg11 Register ..... 2628
12-557. VCOMP_reg12 Register ..... 2629
12-558. VIP_PARSER_main Register ..... 2632
12-559. VIP_PARSER_port_a Register ..... 2633
12-560. VIP_PARSER_xtra_port_a Register ..... 2636
12-561. VIP_PARSER_port_b Register ..... 2637
12-562. VIP_PARSER_xtra_port_b Register ..... 2640
12-563. VIP_PARSER_fiq_mask Register ..... 2641
12-564. VIP_PARSER_fiq_clear Register ..... 2643
12-565. VIP_PARSER_fiq_status Register ..... 2645
12-566. VIP_PARSER_output_port_a_src_fid Register ..... 2647
12-567. VIP_PARSER_output_port_a_enc_fid Register ..... 2649
12-568. VIP_PARSER_output_port_b_src_fid Register ..... 2651
12-569. VIP_PARSER_output_port_b_enc_fid Register ..... 2653
12-570. VIP_PARSER_output_port_a_src0_size Register ..... 2655
12-571. VIP_PARSER_output_port_a_src1_size Register ..... 2656
12-572. VIP_PARSER_output_port_a_src2_size Register ..... 2657
12-573. VIP_PARSER_output_port_a_src3_size Register ..... 2658
12-574. VIP_PARSER_output_port_a_src4_size Register ..... 2659
12-575. VIP_PARSER_output_port_a_src5_size Register ..... 2660
12-576. VIP_PARSER_output_port_a_src6_size Register ..... 2661
12-577. VIP_PARSER_output_port_a_src7_size Register ..... 2662
12-578. VIP_PARSER_output_port_a_src8_size Register ..... 2663
12-579. VIP_PARSER_output_port_a_src9_size Register ..... 2664
12-580. VIP_PARSER_output_port_a_src10_size Register ..... 2665
12-581. VIP_PARSER_output_port_a_src11_size Register ..... 2666
12-582. VIP_PARSER_output_port_a_src12_size Register ..... 2667
12-583. VIP_PARSER_output_port_a_src13_size Register ..... 2668
12-584. VIP_PARSER_output_port_a_src14_size Register ..... 2669
12-585. VIP_PARSER_output_port_a_src15_size Register ..... 2670
12-586. VIP_PARSER_output_port_b_src0_size Register ..... 2671
12-587. VIP_PARSER_output_port_b_src1_size Register ..... 2672
12-588. VIP_PARSER_output_port_b_src2_size Register ..... 2673


Copyright © 2011–2016, Texas Instruments Incorporated

12-589. VIP_PARSER_output_port_b_src3_size Register ..... 2674
12-590. VIP_PARSER_output_port_b_src4_size Register ..... 2675
12-591. VIP_PARSER_output_port_b_src5_size Register ..... 2676
12-592. VIP_PARSER_output_port_b_src6_size Register ..... 2677
12-593. VIP_PARSER_output_port_b_src7_size Register ..... 2678
12-594. VIP_PARSER_output_port_b_src8_size Register ..... 2679
12-595. VIP_PARSER_output_port_b_src9_size Register ..... 2680
12-596. VIP_PARSER_output_port_b_src10_size Register ..... 2681
12-597. VIP_PARSER_output_port_b_src11_size Register ..... 2682
12-598. VIP_PARSER_output_port_b_src12_size Register ..... 2683
12-599. VIP_PARSER_output_port_b_src13_size Register ..... 2684
12-600. VIP_PARSER_output_port_b_src14_size Register ..... 2685
12-601. VIP_PARSER_output_port_b_src15_size Register ..... 2686
12-602. VIP_PARSER_port_a_vdet_vec Register ..... 2687
12-603. VIP_PARSER_port_b_vdet_vec Register ..... 2688
12-604. VIP_PARSER_xtra2_port_a Register ..... 2689
12-605. VIP_PARSER_xtra3_port_a Register ..... 2690
12-606. VIP_PARSER_xtra4_port_a Register ..... 2691
12-607. VIP_PARSER_xtra5_port_a Register ..... 2692
12-608. VIP_PARSER_xtra2_port_b Register ..... 2693
12-609. VIP_PARSER_xtra3_port_b Register ..... 2694
12-610. VIP_PARSER_xtra4_port_b Register ..... 2695
12-611. VIP_PARSER_xtra5_port_b Register ..... 2696
12-612. VIP_PARSER_xtra6_port_a Register ..... 2697
12-613. VIP_PARSER_xtra7_port_b Register ..... 2698
13-1. HDMI Overview ..... 2700
13-2. HDMI Block Diagram ..... 2701
13-3. HDMI Environment ..... 2703
13-4. HDMI Integration ..... 2706
13-5. HDMI Audio Interface Overview ..... 2711
13-6. Audio Data Stuffing Behavior ..... 2715
13-7. Audio Data Stuffing Behavior with Only Three Stereo Channels Active ..... 2716
13-8. L-PCM 16-Bit Format Adaptation ..... 2717
13-9. Transmitter Video Data Processing Path ..... 2719
13-10. IP Revision Identifier Register (HDMI_WP_REVISION) ..... 2720
13-11. Clock Management Configuration Register (HDMI_WP_SYSCONFIG) ..... 2721
13-12. Raw Interrupt Status Register (HDMI_WP_IRQSTATUS_RAW) ..... 2722
13-13. Interrupt Status Register (HDMI_WP_IRQSTATUS) ..... 2723
13-14. Interrupt Enable Register (HDMI_WP_IRQENABLE_SET) ..... 2724
13-15. Interrupt Disable Register (HDMI_WP_IRQENABLE_CLEAR) ..... 2725
13-16. Glitch Filter Register (HDMI_WP_DEBOUNCE) ..... 2726
13-17. Configuration of HDMI Wrapper Video Register (HDMI_WP_VIDEO_CFG) ..... 2727
13-18. Configuration of Clocks Register (HDMI_WP_CLK) ..... 2728
13-19. Audio Configuration in FIFO Register (HDMI_WP_AUDIO_CFG) ..... 2729
13-20. Audio Configuration of DMA Register (HDMI_WP_AUDIO_CFG2) ..... 2730
13-21. Audio FIFO Control Register (HDMI_WP_AUDIO_CTRL) ..... 2731
13-22. TX Data of FIFO Register (HDMI_WP_AUDIO_DATA) ..... 2732
13-23. Vendor ID Register (VND_IDL) ..... 2735
13-24. Vendor ID Register (VND_IDH) ..... 2736


Copyright © 2011–2016, Texas Instruments Incorporated

13-25. Device IDL Register (DEV_IDL) ..... 2736
13-26. Device IDH Register (DEV_IDH) ..... 2736
13-27. Device Revision Register (DEV_REV) ..... 2737
13-28. Software Reset Register (SRST) ..... 2737
13-29. System Control Register 1 (SYS_CTRL1) ..... 2738
13-30. System Status Register (SYS_STAT) ..... 2739
13-31. System Control Register 3 (SYS_CTRL3) ..... 2739
13-32. Data Control Register (DCTL) ..... 2740
13-33. HDCP Control Register (HDCP_CTRL) ..... 2741
13-34. HDCP BKSV Register (BKSV__0-BKSV__4) ..... 2742
13-35. HDCP AN Register (AN__0-AN__7) ..... 2742
13-36. HDCP AKSV Register (AKSV__0-AKSV__4) ..... 2742
13-37. HDCP Ri1 Register (RI1) ..... 2743
13-38. HDCP Ri2 Register (RI2) ..... 2743
13-39. HDCP Ri 128 Compare Register (RI_128_COMP) ..... 2743
13-40. HDCP I Counter Register (I_CNT) ..... 2744
13-41. Ri Status Register (RI_STAT) ..... 2744
13-42. Ri Command Register (RI_CMD) ..... 2745
13-43. Ri Line Start Register (RI_START) ..... 2745
13-44. Ri From RX Registers (Low) (RI_RX_L) ..... 2746
13-45. Ri From RX Registers (High) (RI_RX_H) ..... 2746
13-46. Ri Debug Registers (RI_DEBUG) ..... 2747
13-47. VIDEO DE Delay Register (DE_DLY) ..... 2747
13-48. VIDEO DE Control Register (DE_CTRL) ..... 2748
13-49. VIDEO DE Top Register (DE_TOP) ..... 2749
13-50. VIDEO DE Count Register (DE_CNTL) ..... 2749
13-51. VIDEO DE Count Register (DE_CNTH) ..... 2749
13-52. VIDEO DE Line Register (DE_LINL) ..... 2750
13-53. VIDEO DE Line Register (DE_LINH_1) ..... 2750
13-54. Video H Resolution Register (HRES_L) ..... 2751
13-55. Video H Resolution Register (HRES_H) ..... 2751
13-56. Video V Resolution Register (VRES_L) ..... 2752
13-57. Video V Resolution Register (VRES_H) ..... 2752
13-58. Video Interlace Adjustment Register (IADJUST) ..... 2753
13-59. Video SYNC Polarity Detection Register (POL_DETECT) ..... 2754
13-60. Video Hbit to HSYNC Register (HBIT_2HSYNC1) ..... 2755
13-61. Video Hbit to HSYNC Register (HBIT_2HSYNC2) ..... 2755
13-62. Video Field2 HSYNC Offset Register (FLD2_HS_OFSTL) ..... 2756
13-63. Video Field2 HSYNC Offset Register (FLD2_HS_OFSTH) ..... 2756
13-64. Video HSYNC Length Register (HWIDTH1) ..... 2757
13-65. Video HSYNC Length Register (HWIDTH2) ..... 2757
13-66. Video Vbit to VSYNC Register (VBIT_TO_VSYNC) ..... 2758
13-67. Video VSYNC Length Register (VWIDTH) ..... 2758
13-68. Video Control Register (VID_CTRL) ..... 2759
13-69. Video Action Enable Register (VID_ACEN) ..... 2760
13-70. Video Mode1 Register (VID_MODE) ..... 2761
13-71. Video Blanking Register (VID_BLANK1) ..... 2762
13-72. Video Blanking Register (VID_BLANK2) ..... 2762
13-73. Video Blanking Register (VID_BLANK3) ..... 2762


Copyright © 2011–2016, Texas Instruments Incorporated

13-74. Deep Color Header Register (DC_HEADER) ..... 2763
13-75. Video Mode2 Register (VID_DITHER) ..... 2764
13-76. RGB_2_xvYCC Control Register (RGB2XVYCC_CT) ..... 2765
13-77. RGB_2_xvYCC Conversion R_2_Y Register (R2Y_COEFF_LOW) ..... 2766
13-78. RGB_2_xvYCC Conversion R_2_Y Register (R2Y_COEFF_UP) ..... 2766
13-79. RGB_2_xvYCC Conversion G_2_Y Register (G2Y_COEFF_LOW) ..... 2767
13-80. RGB_2_xvYCC Conversion G_2_Y Register (G2Y_COEFF_UP) ..... 2767
13-81. RGB_2_xvYCC Conversion B_2_Y Register (B2Y_COEFF_LOW) ..... 2768
13-82. RGB_2_xvYCC Conversion B_2_Y Register (B2Y_COEFF_UP) ..... 2768
13-83. RGB_2_xvYCC Conversion R_2_Cb Register (R2CB_COEFF_LOW) ..... 2769
13-84. RGB_2_xvYCC Conversion R_2_Cb Register (R2CB_COEFF_UP) ..... 2769
13-85. RGB_2_xvYCC Conversion G_2_Cb Register (G2CB_COEFF_LOW) ..... 2770
13-86. RGB_2_xvYCC Conversion G_2_Cb Register (G2CB_COEFF_UP) ..... 2770
13-87. RGB_2_xvYCC Conversion B_2_Cb Register (B2CB_COEFF_LOW) ..... 2771
13-88. RGB_2_xvYCC Conversion B_2_Cb Register (B2CB_COEFF_UP) ..... 2771
13-89. RGB_2_xvYCC Conversion R_2_Cr Register (R2CR_COEFF_LOW) ..... 2772
13-90. RGB_2_xvYCC Conversion R_2_Cr Register (R2CR_COEFF_UP) ..... 2772
13-91. RGB_2_xvYCC Conversion G_2_Cr Register (G2CR_COEFF_LOW) ..... 2773
13-92. RGB_2_xvYCC Conversion G_2_Cr Register (G2CR_COEFF_UP) ..... 2773
13-93. RGB_2_xvYCC Conversion B_2_Cr Register (B2CR_COEFF_LOW) ..... 2774
13-94. RGB_2_xvYCC Conversion B_2_Cr Register (B2CR_COEFF_UP) ..... 2774
13-95. RGB_2_xvYCC RGB Input Offset Register (RGB_OFFSET_LOW) ..... 2775
13-96. RGB_2_xvYCC RGB Input Offset Register (RGB_OFFSET_UP) ..... 2775
13-97. RGB_2_xvYCC Conversion Y Output Offset Register (Y_OFFSET_LOW) ..... 2776
13-98. RGB_2_xvYCC Conversion Y Output Offset Register (Y_OFFSET_UP) ..... 2776
13-99. RGB_2_xvYCC Conversion CbCr Output Offset Register (CBCR_OFFSET_LOW) ..... 2777
13-100. RGB_2_xvYCC Conversion CbCr Output Offset Register (CBCR_OFFSET_UP) ..... 2777
13-101. Interrupt State Register (INTR_STATE) ..... 2777
13-102. Interrupt Source Register (INTR1) ..... 2778
13-103. Interrupt Source Register (INTR2) ..... 2779
13-104. Interrupt Source Register (INTR3) ..... 2780
13-105. Interrupt Source Register (INTR4) ..... 2781
13-106. Interrupt Unmask Register (INT_UNMASK1) ..... 2782
13-107. Interrupt Unmask Register (INT_UNMASK2) ..... 2783
13-108. Interrupt Unmask Register (INT_UNMASK3) ..... 2784
13-109. Interrupt Unmask Register (INT_UNMASK4) ..... 2785
13-110. Interrupt Control Register (INT_CTRL) ..... 2786
13-111. xvYCC_2_RGB Control Register (XVYCC2RGB_CTL) ..... 2787
13-112. xvYCC_2_RGB Conversion Y_2_R Register (Y2R_COEFF_LOW) ..... 2788
13-113. xvYCC_2_RGB Conversion Y_2_R Register (Y2R_COEFF_UP) ..... 2788
13-114. xvYCC_2_RGB Conversion Cr_2_R Register (CR2R_COEFF_LOW) ..... 2789
13-115. xvYCC_2_RGB Conversion Cr_2_R Register (C2R2R_COEFF_UP) ..... 2789
13-116. xvYCC_2_RGB Conversion Cb_2_B Register (CB2B_COEFF_LOW) ..... 2790
13-117. xvYCC_2_RGB Conversion Cb_2_B Register (CB2B_COEFF_UP) ..... 2790
13-118. xvYCC_2_RGB Conversion Cr_2_G Register (CR2G_COEFF_LOW) ..... 2791
13-119. xvYCC_2_RGB Conversion Cr_2_G Register (CR2G_COEFF_UP) ..... 2791
13-120. xvYCC_2_RGB Conversion Cb_2_G Register (CB2G_COEFF_LOW) ..... 2792
13-121. xvYCC_2_RGB Conversion Cb_2_G Register (CB2G_COEFF_UP) ..... 2792
13-122. xvYCC_2_RGB Conversion Y Offset Register (YOFFSET1_LOW) ..... 2793


Copyright © 2011–2016, Texas Instruments Incorporated

13-123. xvYCC_2_RGB Conversion Y Offset Register (YOFFSET1_UP) ..... 2793
13-124. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_LOW) ..... 2794
13-125. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_MID) ..... 2794
13-126. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_UP) ..... 2794
13-127. xvYCC_2_RGB Conversion Offset2 Register (OFFSET2_LOW) ..... 2795
13-128. xvYCC_2_RGB Conversion Offset2 Register (OFFSET2_UP) ..... 2795
13-129. xvYCC_2_RGB Conversion DC Level Register (DCLEVEL_LOW) ..... 2796
13-130. xvYCC_2_RGB Conversion DC Level Register (DCLEVEL_UP) ..... 2796
13-131. DDC I2C Manual Register (DDC_MAN) ..... 2797
13-132. DDC I2C Target Slave Address Register (DDC_ADDR) ..... 2798
13-133. DDC I2C Target Segment Address Register (DDC_SEGM) ..... 2798
13-134. DDC I2C Target Offset Address Register (DDC_OFFSET) ..... 2798
13-135. DDC I2C Data Count Register (DDC_COUNT1) ..... 2799
13-136. DDC I2C Data Count Register (DDC_COUNT2) ..... 2799
13-137. DDC I2C Status Register (DDC_STATUS) ..... 2800
13-138. DDC I2C Command Register (DDC_CMD) ..... 2801
13-139. DDC I2C Data Register (DDC_DATA) ..... 2802
13-140. DDC I2C FIFO Count Register (DDC_FIFOCNT) ..... 2802
13-141. ROM Status Register (EPST) ..... 2803
13-142. ROM Command Register (EPCM) ..... 2804
13-143. Gamut Metadata Register (GAMUT_HEADER1) ..... 2805
13-144. Gamut Metadata Register (GAMUT_HEADER2) ..... 2806
13-145. Gamut Metadata Register (GAMUT_HEADER3) ..... 2807
13-146. Gamut Metadata Registers (GAMUT_DBYTE__0-GAMUT_DBYTE__27) ..... 2807
13-147. ACR Control Register (ACR_CTRL) ..... 2809
13-148. ACR Audio Frequency Register (FREQ_SVAL) ..... 2810
13-149. ACR N Software Value Register 1 (N_SVAL1) ..... 2811
13-150. ACR N Software Value Register 2 (N_SVAL2) ..... 2811
13-151. ACR N Software Value Register 3 (N_SVAL3) ..... 2811
13-152. ACR CTS Software Value Register 1 (CTS_SVAL1) ..... 2812
13-153. ACR CTS Software Value Register 2 (CTS_SVAL2) ..... 2812
13-154. ACR CTS Software Value Register 3 (CTS_SVAL3) ..... 2812
13-155. ACR CTS Hardware Value Register 1 (CTS_HVAL1) ..... 2813
13-156. ACR CTS Hardware Value Register 2 (CTS_HVAL2) ..... 2813
13-157. ACR CTS Hardware Value Register 3 (CTS_HVAL3) ..... 2813
13-158. Audio In Mode Register (AUD_MODE) ..... 2814
13-159. Audio In S/PDIF Control Register (SPDIF_CTRL) ..... 2815
13-160. Audio In S/PDIF Extracted Fs and Length Register (HW_SPDIF_FS) ..... 2816
13-161. Audio In I2S Channel Swap Register (SWAP_I2S) ..... 2817
13-162. Audio Error Threshold Register (SPDIF_ERTH) ..... 2818
13-163. Audio In I2S Data In Map Register (I2S_IN_MAP) ..... 2819
13-164. Audio In I2S Control Register (I2S_IN_CTRL) ..... 2820
13-165. Audio In I2S Channel Status Register 0 (I2S_CHST0) ..... 2821
13-166. Audio In I2S Channel Status Register 0 (I2S_CHST1) ..... 2821
13-167. Audio In I2S Channel Status Register 2 (I2S_CHST2) ..... 2822
13-168. Audio In I2S Channel Status Register 3 (I2S_CHST3) ..... 2822
13-169. Audio In I2S Channel Status Register 4 (I2S_CHST4) ..... 2823
13-170. Audio In I2S Channel Status Register 5 (I2S_CHST5) ..... 2824
13-171. Audio Sample Rate Conversion Register (ASRC) ..... 2825


Copyright © 2011–2016, Texas Instruments Incorporated

13-172. Audio I2S Input Length Register (I2S_IN_LEN) ..... 2826
13-173. HDMI Control Register (HDMI_CTRL) ..... 2827
13-174. Audio Path Status Register (AUDO_TXSTAT) ..... 2828
13-175. Audio Input Data Rate Adjustment Register 1 (AUD_PAR_BUSCLK_1) ..... 2829
13-176. Audio Input Data Rate Adjustment Register 2 (AUD_PAR_BUSCLK_2) ..... 2829
13-177. Audio Input Data Rate Adjustment Register 3 (AUD_PAR_BUSCLK_3) ..... 2829
13-178. Test Control Register (TEST_TXCTRL) ..... 2830
13-179. Diagnostic Power Down Register (DPD) ..... 2831
13-180. Packet Buffer Control 1 Register (PB_CTRL1) ..... 2832
13-181. Packet Buffer Control 2 Register (PB_CTRL2) ..... 2833
13-182. AVI InfoFrame Register (AVI_TYPE) ..... 2834
13-183. AVI InfoFrame Register (AVI_VERS) ..... 2834
13-184. AVI InfoFrame Register (AVI_LEN) ..... 2834
13-185. AVI InfoFrame Register (AVI_CHSUM) ..... 2835
13-186. AVI InfoFrame Registers (AVI_DBYTE_0-AVI_DBYTE_14) ..... 2835
13-187. SPD InfoFrame Register (SPD_TYPE) ..... 2836
13-188. SPD InfoFrame Register (SPD_VERS) ..... 2836
13-189. SPD InfoFrame Register (SPD_LEN) ..... 2836
13-190. SPD InfoFrame Register (SPD_CHSUM) ..... 2837
13-191. SPD InfoFrame Registers (SPD_DBYTE_0-SPD_DBYTE_26) ..... 2837
13-192. Audio InfoFrame Register (AUDIO_TYPE) ..... 2838
13-193. Audio InfoFrame Register (AUDIO_VERS) ..... 2838
13-194. Audio InfoFrame Register (AUDIO_LEN) ..... 2838
13-195. Audio InfoFrame Register (AUDIO_CHSUM) ..... 2839
13-196. Audio InfoFrame Registers (AUDIO_DBYTE_0-AUDIO_DBYTE_9) ..... 2839
13-197. MPEG InfoFrame Register (MPEG_TYPE) ..... 2840
13-198. MPEG InfoFrame Register (MPEG_VERS) ..... 2840
13-199. MPEG InfoFrame Register (MPEG_LEN) ..... 2840
13-200. MPEG InfoFrame Register (MPEG_CHSUM) ..... 2841
13-201. MPEG InfoFrame Registers (MPEG_DBYTE_0-MPEG_DBYTE_26) ..... 2841
13-202. Generic Packet Registers (GEN_DBYTE_0-GEN_DBYTE_30) ..... 2841
13-203. General Control Packet Register (CP_BYTE1) ..... 2842
13-204. Generic Packet 2 Registers (GEN2_DBYTE_0-GEN2_DBYTE_30) ..... 2842
13-205. CEC Slave ID Register (CEC_ADDR_ID) ..... 2843
13-206. CEC Device ID Register (CEC_DEV_ID) ..... 2844
13-207. CEC Specification Register (CEC_SPEC) ..... 2845
13-208. EC Specification Suffix Register (CEC_SUFF) ..... 2845
13-209. CEC Firmware Revision Register (CEC_FW) ..... 2846
13-210. CEC Debug Register 0 (CEC_DBG_0) ..... 2846
13-211. CEC Debug Register 1 (CEC_DBG_1) ..... 2846
13-212. CEC Debug Register 2 (CEC_DBG_2) ..... 2847
13-213. CEC Debug Register 3 (CEC_DBG_3) ..... 2848
13-214. CEC Tx Initialization Register (CEC_TX_INIT) ..... 2849
13-215. CEC Tx Destination Register (CEC_TX_DEST) ..... 2849
13-216. CEC Setup Register (CEC_SETUP) ..... 2850
13-217. CEC Tx Command Register (CEC_TX_COMMAND) ..... 2850
13-218. CEC Tx Operand Registers (CEC_TX_OPERAND_0-CEC_TX_OPERAND_14) ..... 2851
13-219. CEC Transmit Data Register (CEC_TRANSMIT_DATA) ..... 2851
13-220. CEC Capture ID0 Register (CEC_CA_7_0) ..... 2852


Copyright © 2011–2016, Texas Instruments Incorporated

13-221. CEC Capture ID0 Register (CEC_CA_15_8) ..... 2852
13-222. CEC Interrupt Enable Register 0 (CEC_INIT_ENABLE_0) ..... 2853
13-223. CEC Interrupt Enable Register 1 (CEC_INIT_ENABLE_1) ..... 2854
13-224. CEC Interrupt Status Register 0 (CEC_INIT_STATUS_0) ..... 2855
13-225. CEC Interrupt Status Register 1 (CEC_INIT_STATUS_1) ..... 2856
13-226. CEC RX Control Register (CEC_RX_CONTROL) ..... 2857
13-227. CEC Rx Count Register (CEC_RX_COUNT) ..... 2857
13-228. CEC Rx Command Header Register (CEC_RX_CMD_HEADER) ..... 2858
13-229. CEC Rx Command Register (CEC_RX_COMMAND) ..... 2858
13-230. CEC Rx Operand Registers (CEC_RX_OPERAND_0-CEC_RX_OPERAND_14) ..... 2859
13-231. TMDS Control Register 2 (TMDS_CNTL2) ..... 2860
13-232. TMDS Control Register 3 (TMDS_CNTL3) ..... 2861
13-233. BIST Control Register (BIST_CNTL) ..... 2862
13-234. TMDS Control Register 9 (TMDS_CNTL9) ..... 2862
14-1. Interrupt Controller ..... 2864
14-2. Interrupt Controller Block Diagram ..... 2865
14-3. ARM A8 Subsystem INTC Integration ..... 2866
14-4. IRQ/FIQ Processing Sequence ..... 2873
14-5. Nested IRQ/FIQ Processing Sequence ..... 2877
14-6. INTCPS_REVISION Register ..... 2880
14-7. INTCPS_SYSCONFIG Register ..... 2880
14-8. INTCPS_SYSSTATUS Register ..... 2881
14-9. INTCPS_SIR_IRQ Register ..... 2881
14-10. INTCPS_SIR_FIQ Register ..... 2882
14-11. INTCPS_CONTROL Register ..... 2882
14-12. INTCPS_PROTECTION Register ..... 2883
14-13. INTCPS_IDLE Register ..... 2883
14-14. INTCPS_IRQ_PRIORITY Register ..... 2884
14-15. INTCPS_FIQ_PRIORITY Register ..... 2884
14-16. INTCPS_THRESHOLD Register ..... 2885
14-17. INTCPS_ITRn Register ..... 2885
14-18. INTCPS_MIRn Register ..... 2886
14-19. INTCPS_MIR_CLEARn Register ..... 2886
14-20. INTCPS_MIR_SETn Register ..... 2886
14-21. INTCPS_ISR_SETn Register ..... 2887
14-22. INTCPS_ISR_CLEARn Register ..... 2887
14-23. INTCPS_PENDING_IRQn Register ..... 2887
14-24. INTCPS_PENDING_FIQn Register ..... 2888
14-25. INTCPS_ILRm Register ..... 2888
15-1. I2C Functional Block Diagram ..... 2890
15-2. Multiple I2C Modules Connected ..... 2891
15-3. Bit Transfer on the I2C Bus ..... 2892
15-4. Start and Stop Condition Events ..... 2893
15-5. I2C Data Transfer ..... 2893
15-6. I2C Data Transfer Formats ..... 2894
15-7. Arbitration Procedure Between Two Master Transmitters ..... 2895
15-8. Synchronization of Two I2C Clock Generators ..... 2895
15-9. Receive FIFO Interrupt Request Generation ..... 2897
15-10. Transmit FIFO Interrupt Request Generation ..... 2898


Copyright © 2011–2016, Texas Instruments Incorporated

15-11. Receive FIFO DMA Request Generation ..... 2899
15-12. Transmit FIFO DMA Request Generation (High Threshold) ..... 2899
15-13. Transmit FIFO DMA Request Generation (Low Threshold) ..... 2900
15-14. I2C_REVNB_LO Register (Module Revision) (LOW BYTES) ..... 2904
15-15. I2C_REVNB_HI Register (HIGH BYTES) (Module Revision) ..... 2905
15-16. I2C_SYSC Register (System Configuration) ..... 2906
15-17. I2C_EOI Register (I2C End of Interrupt) ..... 2907
15-18. I2C_IRQSTATUS_RAW Register (I2C Status Raw) ..... 2908
15-19. I2C_IRQSTATUS Register (I2C Status) ..... 2912
15-20. I2C_IRQENABLE_SET Register (I2C Interrupt Enable Set) ..... 2914
15-21. I2C_IRQENABLE_CLR Register (I2C Interrupt Enable Clear) ..... 2916
15-22. I2C_WE Register (I2C Wakeup Enable) ..... 2918
15-23. I2C_DMARXENABLE_SET Register (Receive DMA Enable Set) ..... 2921
15-24. I2C_DMATXENABLE_SET Register (Transmit DMA Enable Set) ..... 2921
15-25. I2C_DMARXENABLE_CLR Register (Receive DMA Enable Clear) ..... 2922
15-26. I2C_DMATXENABLE_CLR Register (Transmit DMA Enable Clear) ..... 2922
15-27. I2C_DMARXWAKE_EN Register (Receive DMA Wakeup) ..... 2923
15-28. I2C_DMATXWAKE_EN Register (Transmit DMA Wakeup) ..... 2925
15-29. I2C_SYSS Register (System Status) ..... 2927
15-30. I2C_BUF Register (Buffer Configuration) ..... 2928
15-31. I2C_CNT Register (Data Counter) ..... 2930
15-32. I2C_DATA Register (Data Access) ..... 2931
15-33. I2C_CON Register (I2C Configuration) ..... 2932
15-34. I2C_OA Register (I2C Own Address) ..... 2934
15-35. I2C_SA Register (I2C Own Address) ..... 2935
15-36. I2C_PSC Register (I2C Own Address) ..... 2936
15-37. Clock Divider ..... 2936
15-38. I2C_SCLL Register (I2C SCL Low Time) ..... 2937
15-39. I2C_SCLH Register (I2C SCL High Time) ..... 2937
15-40. I2C_SYSTEST Register (System Test) ..... 2938
15-41. I2C_BUFSTAT Register (I2C Buffer Status) ..... 2941
15-42. I2C_OA1 Register (OA1) (Own Address 1) ..... 2942
15-43. I2C_OA2 Register (I2C Own Address 2) ..... 2943
15-44. I2C_OA3 Register (I2C Own Address 3) ..... 2944
15-45. I2C_ACTOA Register (Active Own Address) ..... 2945
15-46. I2C_SBLOCK Register (I2C Clock Blocking Enable) ..... 2946
16-1. McASP Block Diagram ..... 2950
16-2. McASP to Parallel 2-Channel DACs ..... 2951
16-3. McASP to 6-Channel DAC and 2-Channel DAC ..... 2951
16-4. McASP to Digital Amplifier ..... 2952
16-5. McASP as Digital Audio Encoder ..... 2952
16-6. McASP as 16 Channel Digital Processor ..... 2952
16-7. TDM Format–6 Channel TDM Example ..... 2953
16-8. TDM Format Bit Delays from Frame Sync ..... 2954
16-9. Inter-Integrated Sound (I2S) Format ..... 2954
16-10. Biphase-Mark Code (BMC) ..... 2955
16-11. S/PDIF Subframe Format ..... 2956
16-12. S/PDIF Frame Format ..... 2957
16-13. Definition of Bit, Word, and Slot ..... 2958


Copyright © 2011–2016, Texas Instruments Incorporated

16-14. Bit Order and Word Alignment Within a Slot Examples ..... 2958
16-15. Definition of Frame and Frame Sync Width ..... 2959
16-16. Transmit Clock Generator Block Diagram ..... 2961
16-17. Receive Clock Generator Block Diagram ..... 2962
16-18. Frame Sync Generator Block Diagram ..... 2963
16-19. Burst Frame Sync Mode ..... 2965
16-20. Transmit DMA Event (AXEVT) Generation in TDM Time Slots ..... 2967
16-21. Individual Serializer and Connections Within McASP ..... 2972
16-22. Receive Format Unit ..... 2974
16-23. Transmit Format Unit ..... 2974
16-24. McASP I/O Pin Control Block Diagram ..... 2976
16-25. Processor Service Time Upon Transmit DMA Event (AXEVT) ..... 2978
16-26. Processor Service Time Upon Receive DMA Event (AREVT) ..... 2980
16-27. McASP Audio FIFO (AFIFO) Block Diagram ..... 2982
16-28. Data Flow Through Transmit Format Unit, Illustrated ..... 2985
16-29. Data Flow Through Receive Format Unit, Illustrated ..... 2987
16-30. Transmit Clock Failure Detection Circuit Block Diagram ..... 2991
16-31. Receive Clock Failure Detection Circuit Block Diagram ..... 2992
16-32. Serializers in Loopback Mode ..... 2993
16-33. Audio Mute (AMUTE) Block Diagram ..... 2999
16-34. DMA Events in an Audio Example–Two Events (Scenario 1) ..... 3001
16-35. DMA Events in an Audio Example–Four Events (Scenario 2) ..... 3001
16-36. DMA Events in an Audio Example ..... 3002
16-37. Revision Identification Register (REV) ..... 3006
16-38. Pin Function Register (PFUNC) ..... 3006
16-39. Pin Direction Register (PDIR) ..... 3008
16-40. Pin Data Output Register (PDOUT) ..... 3010
16-41. Pin Data Input Register (PDIN) ..... 3012
16-42. Pin Data Set Register (PDSET) ..... 3014
16-43. Pin Data Clear Register (PDCLR) ..... 3016
16-44. Global Control Register (GBLCTL) ..... 3018
16-45. Audio Mute Control Register (AMUTE) ..... 3020
16-46. Digital Loopback Control Register (DLBCTL) ..... 3022
16-47. Digital Mode Control Register (DITCTL) ..... 3023
16-48. Receiver Global Control Register (RGBLCTL) ..... 3024
16-49. Receive Format Unit Bit Mask Register (RMASK) ..... 3025
16-50. Receive Bit Stream Format Register (RFMT) ..... 3026
16-51. Receive Frame Sync Control Register (AFSRCTL) ..... 3028
16-52. Receive Clock Control Register (ACLKRCTL) ..... 3029
16-53. Receive High-Frequency Clock Control Register (AHCLKRCTL) ..... 3030
16-54. Receive TDM Time Slot Register (RTDM) ..... 3031
16-55. Receiver Interrupt Control Register (RINTCTL) ..... 3032
16-56. Receiver Status Register (RSTAT) ..... 3033
16-57. Current Receive TDM Time Slot Registers (RSLOT) ..... 3034
16-58. Receive Clock Check Control Register (RCLKCHK) ..... 3035
16-59. Receiver DMA Event Control Register (REVTCTL) ..... 3036
16-60. Transmitter Global Control Register (XGBLCTL) ..... 3037
16-61. Transmit Format Unit Bit Mask Register (XMASK) ..... 3038
16-62. Transmit Bit Stream Format Register (XFMT) ..... 3039


Copyright © 2011–2016, Texas Instruments Incorporated

16-63. Transmit Frame Sync Control Register (AFSXCTL) ..... 3041
16-64. Transmit Clock Control Register (ACLKXCTL) ..... 3042
16-65. Transmit High-Frequency Clock Control Register (AHCLKXCTL) ..... 3043
16-66. Transmit TDM Time Slot Register (XTDM) ..... 3044
16-67. Transmitter Interrupt Control Register (XINTCTL) ..... 3045
16-68. Transmitter Status Register (XSTAT) ..... 3046
16-69. Current Transmit TDM Time Slot Register (XSLOT) ..... 3047
16-70. Transmit Clock Check Control Register (XCLKCHK) ..... 3048
16-71. Transmitter DMA Event Control Register (XEVTCTL) ..... 3049
16-72. Serializer Control Registers (SRCTL*n*) ..... 3050
16-73. DIT Left Channel Status Registers (DITCSRA0-DITCSRA5) ..... 3051
16-74. DIT Right Channel Status Registers (DITCSRB0-DITCSRB5) ..... 3051
16-75. DIT Left Channel User Data Registers (DITUDRA0-DITUDRA5) ..... 3051
16-76. DIT Right Channel User Data Registers (DITUDRB0-DITUDRB5) ..... 3051
16-77. Transmit Buffer Registers (XBUFn) ..... 3052
16-78. Receive Buffer Registers (RBUF*n*) ..... 3052
16-79. Write FIFO Control Register (WFIFOCTL) ..... 3053
16-80. Write FIFO Status Register (WFIFOSTS) ..... 3054
16-81. Read FIFO Control Register (RFIFOCTL) ..... 3055
16-82. Read FIFO Status Register (RFIFOSTS) ..... 3056
17-1. McBSP Block Diagram ..... 3059
17-2. McBSP Data Transfer Paths ..... 3060
17-3. Clock Signal Control of Bit Transfer Timing ..... 3061
17-4. McBSP Operating at Maximum Packet Frequency ..... 3062
17-5. Single-Phase Frame for a McBSP Data Transfer ..... 3064
17-6. Dual-Phase Frame for a McBSP Data Transfer ..... 3064
17-7. McBSP Reception Physical Data Path ..... 3065
17-8. McBSP Reception Signal Activity ..... 3065
17-9. McBSP Transmission Physical Data Path ..... 3066
17-10. McBSP Transmission Signal Activity ..... 3066
17-11. Sample Rate Generator Block Diagram ..... 3067
17-12. CLKG Synchronization and FSG Generation (GSYNC = 1 and CLKGDV = 1) ..... 3071
17-13. CLKG Synchronization and FSG Generation (GSYNC = 1 and CLKGDV = 3h) ..... 3071
17-14. Overrun in the McBSP Receiver ..... 3073
17-15. Unexpected Frame-Synchronization Pulse During a McBSP Reception ..... 3074
17-16. Proper Positioning of Receive Frame-Synchronization Pulses ..... 3075
17-17. Unexpected Frame-Synchronization Pulse During a McBSP Transmission ..... 3076
17-18. Proper Positioning of Transmit Frame-Synchronization Pulses ..... 3077
17-19. McBSP Data Transfer in the 8-Partition Mode ..... 3080
17-20. Alternating Between Partitions A and B Channels ..... 3081
17-21. Activity on McBSP Pins for the Possible Values of XMCM Bit ..... 3083
17-22. Transmit Full Cycle Mode ..... 3085
17-23. Transmit Half Cycle Mode ..... 3085
17-24. Receive Full Cycle Mode ..... 3086
17-25. Receive Half Cycle Mode ..... 3086
17-26. Range of Programmable Data Delay ..... 3094
17-27. 2-Bit Data Delay Used to Skip a Framing Bit ..... 3095
17-28. Data Externally Clocked on a Rising Edge and Sampled on a Falling Edge ..... 3097
17-29. Frame of 16 CLKG Periods and Active Width of 2 CLKG Periods ..... 3098


Copyright © 2011–2016, Texas Instruments Incorporated

17-30. Range of Programmable Data Delay ..... 3103
17-31. 2-Bit Data Delay Used to Skip a Framing Bit ..... 3103
17-32. Frame of 16 CLKG Periods and Active Width of 2 CLKG Periods ..... 3106
17-33. Revision Number Register (REVNB) ..... 3112
17-34. System Configuration Register (SYSCONFIG_REG) ..... 3113
17-35. End of Interrupt Register (EOI) ..... 3114
17-36. Interrupt Status Raw Register (IRQSTATUS_RAW) ..... 3114
17-37. Interrupt Status Register (IRQSTATUS) ..... 3117
17-38. Interrupt Enable Set Register (IRQENABLE_SET) ..... 3119
17-39. Interrupt Enable Clear Register (IRQENABLE_CLR) ..... 3121
17-40. DMA Rx Enable Set Register (DMARXENABLE_SET) ..... 3123
17-41. DMA Tx Enable Set Register (DMATXENABLE_SET) ..... 3123
17-42. DMA Rx Enable Clear Register (DMARXENABLE_CLR) ..... 3124
17-43. DMA Tx Enable Clear Register (DMATXENABLE_CLR) ..... 3124
17-44. DMA Rx Wake Enable Register (DMARXWAKE_EN) ..... 3125
17-45. DMA Tx Wake Enable Register (DMATXWAKE_EN) ..... 3127
17-46. McBSP_DRR_REG ..... 3129
17-47. McBSP_DRR_REG ..... 3129
17-48. McBSP_SPCR2_REG ..... 3129
17-49. McBSP_SPCR1_REG ..... 3131
17-50. McBSP_RCR2_REG ..... 3132
17-51. McBSP_RCR1_REG ..... 3133
17-52. McBSP_XCR2_REG ..... 3134
17-53. McBSP_XCR1_REG ..... 3135
17-54. McBSP_SRGR2_REG ..... 3136
17-55. McBSP_SRGR1_REG ..... 3137
17-56. McBSP_MCR2_REG ..... 3138
17-57. McBSP_MCR1_REG ..... 3140
17-58. McBSP_RCERA_REG ..... 3142
17-59. McBSP_RCERB_REG ..... 3142
17-60. McBSP_XCERA_REG ..... 3143
17-61. McBSP_XCERB_REG ..... 3143
17-62. McBSP_PCR_REG ..... 3144
17-63. McBSP_RCERC_REG ..... 3146
17-64. McBSP_RCERD_REG ..... 3146
17-65. McBSP_XCERC_REG ..... 3147
17-66. McBSP_XCERD_REG ..... 3147
17-67. McBSP_RCERE_REG ..... 3148
17-68. McBSP_RCERF_REG ..... 3148
17-69. McBSP_XCERE_REG ..... 3149
17-70. McBSP_XCERF_REG ..... 3149
17-71. McBSP_RCERG_REG ..... 3150
17-72. McBSP_RCERH_REG ..... 3150
17-73. McBSP_XCERG_REG ..... 3151
17-74. McBSP_XCERH_REG ..... 3151
17-75. McBSP_THRSH2_REG ..... 3152
17-76. McBSP_THRSH1_REG ..... 3152
17-77. McBSP_IRQSTATUS_REG ..... 3153
17-78. McBSP_IRQENABLE_REG ..... 3155


Copyright © 2011–2016, Texas Instruments Incorporated

17-79. McBSP_WAKEUPEN_REG ..... 3157
17-80. McBSP_XCCR_REG ..... 3158
17-81. McBSP_RCCR_REG ..... 3160
17-82. McBSP_XBUFFSTAT_REG ..... 3161
17-83. McBSP_RBUFFSTAT_REG ..... 3161
17-84. McBSP_STATUS_REG ..... 3162
18-1. MMC/SD/SDIO Overview ..... 3164
18-2. MMC/SD Connectivity to an MMC/SD Card ..... 3166
18-3. MMC/SD Connectivity to an MMC/SD Card ..... 3167
18-4. Sequential Read Operation (MMC Cards Only) ..... 3169
18-5. Sequential Write Operation (MMC Cards Only) ..... 3169
18-6. Multiple Block Read Operation (MMC Cards Only) ..... 3170
18-7. Multiple Block Write Operation (MMC Cards Only) ..... 3170
18-8. Command Token Format ..... 3171
18-9. 48-Bit Response Packet (R1, R3, R4, R5, R6) ..... 3171
18-10. 136-Bit Response Packet (R2) ..... 3171
18-11. Data Packet for Sequential Transfer (1-Bit) ..... 3172
18-12. Data Packet for Block Transfer (1-Bit) ..... 3172
18-13. Data Packet for Block Transfer (4-Bit) ..... 3172
18-14. Data Packet for Block Transfer (8-Bit) ..... 3173
18-15. DMA Receive Mode ..... 3181
18-16. DMA Transmit Mode ..... 3182
18-17. Buffer Management for a Write ..... 3184
18-18. Buffer Management for a Read ..... 3185
18-19. Busy Timeout for R1b, R5b Responses ..... 3188
18-20. Busy Timeout After Write CRC Status ..... 3188
18-21. Write CRC Status Timeout ..... 3189
18-22. Read Data Timeout ..... 3189
18-23. Boot Acknowledge Timeout When Using CMD0 ..... 3190
18-24. Boot Acknowledge Timeout When CMD Held Low ..... 3190
18-25. Auto CMD12 Timing During Write Transfer ..... 3191
18-26. Auto Command 12 Timings During Read Transfer ..... 3192
18-27. Output Driven on Falling Edge ..... 3194
18-28. Output Driven on Rising Edge ..... 3195
18-29. Boot Mode With CMD0 ..... 3196
18-30. Boot Mode With CMD Line Tied to 0 ..... 3196
18-31. MMC/SD/SDIO Controller Software Reset Flow ..... 3200
18-32. MMC/SD/SDIO Controller Bus Configuration Flow ..... 3201
18-33. MMC/SD/SDIO Controller Card Identification and Selection - Part 1 ..... 3202
18-34. MMC/SD/SDIO Controller Card Identification and Selection - Part 2 ..... 3203
18-35. IP Revision Identifier Register (MMCHS_HL_REV) ..... 3205
18-36. IP Module Hardware Configuration Register (MMCHS_HL_HWINFO) ..... 3206
18-37. Clock Management Configuration (MMCHS_HL_SYSCONFIG ..... 3207
18-38. System Configuration Register (MMCHS_SYSCONFIG) ..... 3208
18-39. System Status Register (MMCHS_SYSSTATUS) ..... 3210
18-40. Card Status Response Error (MMCHS_CSRE) ..... 3210
18-41. System Test Register (MMCHS_SYSTEST) ..... 3211
18-42. Configuration Register (MMCHS_CON) ..... 3215
18-43. Power Counter Register (MMCHS_PWCNT) ..... 3219


Copyright © 2011–2016, Texas Instruments Incorporated

18-44. Card Status Response Error (MMCHS_SDMASA) ..... 3219
18-45. Transfer Length Configuration Register (MMCHS_BLK) ..... 3220
18-46. Command Argument Register (MMCHS_ARG) ..... 3221
18-47. Command and Transfer Mode Register (MMCHS_CMD) ..... 3221
18-48. Command Response[31:0] Register (MMCHS_RSP10) ..... 3225
18-49. Command Response[63:32] Register (MMCHS_RSP32) ..... 3225
18-50. Command Response[95:64] Register (MMCHS_RSP54) ..... 3226
18-51. Command Response[127:96] Register (MMCHS_RSP76) ..... 3226
18-52. Data Register (MMCHS_DATA) ..... 3227
18-53. Present State Register (MMCHS_PSTATE) ..... 3228
18-54. Control Register (MMCHS_HCTL) ..... 3230
18-55. SD System Control Register (MMCHS_SYSCTL) ..... 3233
18-56. Interrupt Status Register (MMCHS_STAT) ..... 3235
18-57. Interrupt SD Enable Register (MMCHS_IE) ..... 3240
18-58. Interrupt Signal Enable Register (MMCHS_ISE) ..... 3243
18-59. Auto CMD12 Error Status Register (MMCHS_AC12) ..... 3246
18-60. Capabilities Register (MMCHS_CAPA) ..... 3247
18-61. Interrupt Signal Enable Register (MMCHS_FE) ..... 3249
18-62. ADMA Error Status Register (MMCHS_ADMAES) ..... 3251
18-63. ADMA System Address Low Bits (MMCHS_ADMASAL) ..... 3252
18-64. ADMA System Address High Bits Register (MMCHS_ADMASAH) ..... 3252
18-65. Versions Register (MMCHS_REV) ..... 3253
19-1. PCIe Subsystem (PCIESS) Block Diagram ..... 3257
19-2. Address Space Zero Regions ..... 3266
19-3. PCI Express Power Management State Transitions ..... 3274
19-4. ASPM Link State Transitions Diagram ..... 3275
19-5. Software driven Link Power State Transition ..... 3275
19-6. PID Register ..... 3284
19-7. CMD_STATUS Register ..... 3284
19-8. CFG_SETUP Register ..... 3286
19-9. IOBASE Register ..... 3286
19-10. TLPCFG Register ..... 3286
19-11. RSTCMD Register ..... 3287
19-12. PMCMD Register ..... 3287
19-13. PMCFG Register ..... 3288
19-14. ACT_STATUS Register ..... 3288
19-15. OB_SIZE Register ..... 3288
19-16. DIAG_CTRL Register ..... 3289
19-17. ENDIAN Register ..... 3289
19-18. PRIORITY Register ..... 3290
19-19. IRQ_EOI Register ..... 3290
19-20. MSI_IRQ Register ..... 3290
19-21. EP_IRQ_SET Register ..... 3291
19-22. EP_IRQ_CLR Register ..... 3291
19-23. EP_IRQ_STATUS Register ..... 3292
19-24. GPR0 Register ..... 3292
19-25. GPR1 Register ..... 3292
19-26. GPR2 Register ..... 3292
19-27. GPR3 Register ..... 3293


Copyright © 2011–2016, Texas Instruments Incorporated

19-28. MSI0_STATUS_RAW Register ..... 3293
19-29. MSI0_IRQ_STATUS Register ..... 3293
19-30. MSI0_IRQ_ENABLE_SET Register ..... 3294
19-31. MSI0_IRQ_ENABLE_CLR Register ..... 3294
19-32. IRQ_STATUS_RAW Register ..... 3294
19-33. IRQ_STATUS Register ..... 3295
19-34. IRQ_ENABLE_SET Register ..... 3295
19-35. IRQ_ENABLE_CLR Register ..... 3296
19-36. ERR_IRQ_STATUS_RAW Register ..... 3296
19-37. ERR_IRQ_STATUS Register ..... 3297
19-38. ERR_IRQ_ENABLE_SET Register ..... 3297
19-39. ERR_IRQ_ENABLE_CLR Register ..... 3298
19-40. PMRST_IRQ_STATUS_RAW Register ..... 3298
19-41. PMRST_IRQ_STATUS Register ..... 3299
19-42. PMRST_ENABLE_SET Register ..... 3299
19-43. PMRST_ENABLE_CLR Register ..... 3300
19-44. OB_OFFSET_INDEXn Register ..... 3300
19-45. OB_OFFSETn_HI Register ..... 3300
19-46. IB_BAR0 Register ..... 3301
19-47. IB_START0_LO Register ..... 3301
19-48. IB_START0_HI Register ..... 3301
19-49. IB_OFFSET0 Register ..... 3302
19-50. IB_BAR1 Register ..... 3302
19-51. IB_START1_LO Register ..... 3303
19-52. IB_START1_HI Register ..... 3303
19-53. IB_OFFSET1 Register ..... 3303
19-54. IB_BAR2 Register ..... 3304
19-55. IB_START2_LO Register ..... 3304
19-56. IB_START2_HI Register ..... 3304
19-57. IB_OFFSET2 Register ..... 3305
19-58. IB_BAR3 Register ..... 3305
19-59. IB_START3_LO Register ..... 3305
19-60. IB_START3_HI Register ..... 3306
19-61. IB_OFFSET3 Register ..... 3306
19-62. PCS_CFG0 Register ..... 3306
19-63. PCS_CFG1 Register ..... 3307
19-64. PCS_STATUS Register ..... 3307
19-65. SERDES_STATUS Register ..... 3308
19-66. SERDES_RXCFG0 Register ..... 3308
19-67. SERDES_RXCFG1 Register ..... 3310
19-68. SERDES_RXCFG2 Register ..... 3312
19-69. SERDES_RXCFG3 Register ..... 3313
19-70. SERDES_RXCFG4 Register ..... 3314
19-71. SERDES_TXCFG0 Register ..... 3316
19-72. SERDES_TXCFG1 Register ..... 3319
19-73. SERDES_TXCFG2 Register ..... 3321
19-74. SERDES_TXCFG3 Register ..... 3322
19-75. SERDES_TXCFG4 Register ..... 3323
19-76. VENDOR_DEVICE_ID Register ..... 3324


Copyright © 2011–2016, Texas Instruments Incorporated

19-77. STATUS_COMMAND Register ..... 3324
19-78. CLASSCODE_REVID Register ..... 3325
19-79. BIST_HEADER Register ..... 3326
19-80. BAR0 Register ..... 3326
19-81. BAR1 Register ..... 3327
19-82. BAR1 Register ..... 3328
19-83. BAR2 Register ..... 3328
19-84. BAR3 Register ..... 3329
19-85. BAR3 Register ..... 3329
19-86. BAR4 Register ..... 3329
19-87. BAR5 Register ..... 3330
19-88. BAR5 Register ..... 3330
19-89. CARDBUS Register ..... 3331
19-90. SUBSYS_VNDR_ID Register ..... 3331
19-91. EXPNSN_ROM Register ..... 3331
19-92. CAP_PTR Register ..... 3332
19-93. Reserved Register ..... 3332
19-94. INT_PIN Register ..... 3333
19-95. BIST_HEADER Register ..... 3333
19-96. BAR0 Register ..... 3334
19-97. BAR1 Register ..... 3334
19-98. BAR1 Register ..... 3335
19-99. BUSNUM Register ..... 3335
19-100. SECSTAT Register ..... 3336
19-101. MEMSPACE Register ..... 3336
19-102. PREFETCH_MEM Register ..... 3337
19-103. PREFETCH_BASE Register ..... 3337
19-104. PREFETCH_LIMIT Register ..... 3338
19-105. IOSPACE Register ..... 3338
19-106. CAP_PTR Register ..... 3338
19-107. EXPNSN_ROM Register ..... 3339
19-108. BRIDGE_INT Register ..... 3339
19-109. PMCAP Register ..... 3340
19-110. PM_CTL_STAT Register ..... 3341
19-111. MSI_CAP Register ..... 3342
19-112. MSI_LOW32 Register ..... 3342
19-113. MSI_UP32 Register ..... 3342
19-114. MSI_DATA Register ..... 3343
19-115. PCIES_CAP Register ..... 3344
19-116. DEVICE_CAP Register ..... 3344
19-117. DEV_STAT_CTRL Register ..... 3345
19-118. LINK_CAP Register ..... 3346
19-119. LINK_STAT_CTRL Register ..... 3347
19-120. SLOT_CAP Register ..... 3348
19-121. SLOT_STAT_CTRL Register ..... 3348
19-122. ROOT_CTRL_CAP Register ..... 3349
19-123. ROOT_STATUS Register ..... 3350
19-124. DEV_CAP2 Register ..... 3350
19-125. DEV_STAT_CTRL2 Register ..... 3351


Copyright © 2011–2016, Texas Instruments Incorporated

19-126. LINK_CTRL2 Register ..... 3351
19-127. PCIE_EXTCAP Register ..... 3352
19-128. PCIE_UNCERR Register ..... 3353
19-129. PCIE_UNCERR_MASK Register ..... 3353
19-130. PCIE_UNCERR_SVRTY Register ..... 3354
19-131. PCIE_CERR Register ..... 3355
19-132. PCIE_CERR_MASK Register ..... 3355
19-133. PCIE_ACCR Register ..... 3356
19-134. HDR_LOG0 Register ..... 3356
19-135. HDR_LOG1 Register ..... 3357
19-136. HDR_LOG2 Register ..... 3357
19-137. HDR_LOG3 Register ..... 3357
19-138. RC_ERR_CMD Register ..... 3358
19-139. RC_ERR_ST Register ..... 3358
19-140. ERR_SRC_ID Register ..... 3359
19-141. PL_ACKTIMER Register ..... 3359
19-142. PL_OMSG Register ..... 3360
19-143. PL_FORCE_LINK Register ..... 3360
19-144. ACK_FREQ Register ..... 3360
19-145. PL_LINK_CTRL Register ..... 3361
19-146. LANE_SKEW Register ..... 3362
19-147. SYM_NUM Register ..... 3362
19-148. SYMTIMER_FLTMASK Register ..... 3363
19-149. FLT_MASK2 Register ..... 3364
19-150. DEBUG0 Register ..... 3364
19-151. DEBUG1 Register ..... 3365
19-152. PL_GEN2 Register ..... 3366
20-1. RTC Block Diagram ..... 3368
20-2. RTC Functional Block Diagram ..... 3369
20-3. Kick Register State Machine Diagram ..... 3371
20-4. Flow Control for Updating RTC Registers ..... 3373
20-5. Compensation Illustration ..... 3374
20-6. Seconds Register (SECONDS_REG) ..... 3377
20-7. Minutes Register (MINUTES_REG) ..... 3377
20-8. Hours Register (HOURS_REG) ..... 3378
20-9. Days of the Month Register (DAYS_REG) ..... 3378
20-10. Month Register (MONTHS_REG) ..... 3379
20-11. Year Register (YEARS_REG) ..... 3379
20-12. Day of the Week Register (WEEKS_REG) ..... 3380
20-13. Alarm Second Register (ALARM_SECONDS_REG) ..... 3380
20-14. Alarm Minute Register (ALARM_MINUTES_REG) ..... 3381
20-15. Alarm Hour Register (ALARM_HOURS_REG) ..... 3381
20-16. Alarm Day of the Month (ALARM_DAYS_REG) ..... 3382
20-17. Alarm Month Register (ALARM_MONTHS_REG) ..... 3382
20-18. Alarm Year Register (ALARM_YEARS_REG) ..... 3383
20-19. Control Register (CTRL_REG) ..... 3384
20-20. Status Register (STATUS_REG) ..... 3386
20-21. Interrupt Register (INTERRUPTS_REG) ..... 3387
20-22. Compensation (LSB) Register (COMP_LSB_REG) ..... 3388


Copyright © 2011–2016, Texas Instruments Incorporated

20-23. Compensation (MSB) Register (COMP_MSB_REG) ..... 3389
20-24. Oscillator Register (OSC_REG) ..... 3390
20-25. Scratch Registers (SCRATCHx_REG) ..... 3390
20-26. Kick0 Register (KICK0R) ..... 3391
20-27. Kick1 Register (KICK1R) ..... 3391
20-28. RTC Revision Register (RTC_REVISION) ..... 3392
20-29. System Configuration Register (RTC_SYSCONFIG) ..... 3392
20-30. Wakeup Enable Register (RTC_IRQWAKEEN) ..... 3393
21-1. SATA Core Block Diagram ..... 3397
21-2. HBA Capabilities Register (CAP) ..... 3426
21-3. Global HBA Control Register (GHC) ..... 3427
21-4. Interrupt Status Register (IS) ..... 3428
21-5. Ports Implemented Register (PI) ..... 3429
21-6. AHCI Version Register (VS) ..... 3429
21-7. Command Completion Coalescing Control Register (CCC_CTL) ..... 3430
21-8. Command Completion Coalescing Ports Register (CCC_PORTS) ..... 3431
21-9. BIST Active FIS Register (BISTAFR) ..... 3432
21-10. BIST Control Register (BISTCR) ..... 3433
21-11. BIST FIS Count Register (BISTFCTR) ..... 3435
21-12. BIST Status Register (BISTSR) ..... 3435
21-13. BIST DWORD Error Count Register (BISTDECR) ..... 3436
21-14. BIST DWORD Error Count Register (TIMER1MS) ..... 3436
21-15. Global Parameter 1 Register (GPARAM1R) ..... 3437
21-16. Global Parameter 2 Register (GPARAM2R) ..... 3438
21-17. Port Parameter Register (PPARAMR) ..... 3439
21-18. Version Register (VERSIONR) ..... 3440
21-19. ID Register (IDR) ..... 3440
21-20. Port Command List Base Address Register (P0CLB) ..... 3441
21-21. Port FIS Base Address Register (P0FB) ..... 3441
21-22. Port Interrupt Status Register (P0IS) ..... 3442
21-23. Port Interrupt Enable Register (P0IE) ..... 3444
21-24. Port Command Register (P0CMD) ..... 3445
21-25. Port Task File Data Register (P0TFD) ..... 3447
21-26. Port Signature Register (P0SIG) ..... 3448
21-27. Port Serial ATA Status Register (P0SSTS) ..... 3448
21-28. Port Serial ATA Control Register (P0SCTL) ..... 3450
21-29. Port Serial ATA Error Register (P0SERR) ..... 3451
21-30. Port Serial ATA Active Register (P0SACT) ..... 3453
21-31. Port Command Issue Register (P0CI) ..... 3453
21-32. Port Serial ATA Notification Register (P0SNTF) ..... 3454
21-33. Port DMA Control Register (P0DMACR) ..... 3455
21-34. Idle Register (IDLE) ..... 3457
21-35. PHY Configuration Receive 0 Register (CFGRX0) ..... 3458
21-36. PHY Configuration Receive 1 Register (CFGRX1) ..... 3460
21-37. PHY Configuration Receive 2 Register (CFGRX2) ..... 3461
21-38. PHY Configuration Receive 3 Register (CFGRX3) ..... 3462
21-39. PHY Configuration Receive 4 Register (CFGRX4) ..... 3464
21-40. Receive Bus PHY-to-Controller Status Register ..... 3465
21-41. PHY Configuration Transmit 0 Register (CFGTX0) ..... 3466


Copyright © 2011–2016, Texas Instruments Incorporated

21-42. SERDES_TXCFG1 Register ..... 3469
21-43. PHY Configuration Transmit 2 Register (CFGTX2) ..... 3471
21-44. PHY Configuration Transmit 3 Register (CFGTX3) ..... 3473
21-45. PHY Configuration Transmit 4 Register (CFGTX4) ..... 3474
21-46. Transmit Bus Controller-to-PHY Status Register ..... 3474
21-47. SATA PLL Configuration 0 Register (SATA_PLLCFG0) ..... 3475
21-48. SATA PLL Configuration 1 Register (SATA_PLLCFG1) ..... 3477
21-49. SATA PLL Configuration 2 Register (SATA_PLLCFG2) ..... 3478
21-50. SATA PLL Configuration 3 Register (SATA_PLLCFG3) ..... 3479
21-51. SATA PLL Configuration 4 Register (SATA_PLLCFG4) ..... 3480
21-52. SATA PLL Status (SATA_PLLSTATUS) ..... 3482
21-53. SATA Receive Status (SATA_RXSTATUS) ..... 3482
21-54. SATA Transmit Status (SATA_TXSTATUS) ..... 3483
21-55. SATA TEST Configuration (SATA_TESTCFG) ..... 3483
21-56. Clock Domain Power Control Register (CM_DEFAULT_L3_MED_CLKSTCTRL) ..... 3484
21-57. Clock Domain SATA Control Register (CM_DEFAULT_SATA_CLKCTRL) ..... 3485
22-1. SPI System Overview ..... 3487
22-2. SPI Full-Duplex Transmission ..... 3489
22-3. SPI Half-Duplex Transmission (Receive-only Slave) ..... 3490
22-4. SPI Half-Duplex Transmission (Transmit-Only Slave) ..... 3490
22-5. Phase and Polarity Combinations ..... 3492
22-6. Full Duplex Single Transfer Format with PHA = 0 ..... 3493
22-7. Full Duplex Single Transfer Format With PHA = 1 ..... 3494
22-8. Continuous Transfers With $\overline{\text{SPI_SCS}}$[n] Maintained Active (Single-Data-Pin Interface Mode) ..... 3499
22-9. Continuous Transfers With $\overline{\text{SPI_SCS}}$[n] Maintained Active (Dual-Data-Pin Interface Mode) ..... 3499
22-10. Extended SPI Transfer With Start Bit PHA = 1 ..... 3501
22-11. Chip-Select $\overline{\text{SPI_SCS}}$[n] Timing Controls ..... 3502
22-12. Transmit/Receive Mode With No FIFO Used ..... 3505
22-13. Transmit/Receive Mode With Only Receive FIFO Enabled ..... 3505
22-14. Transmit/Receive Mode With Only Transmit FIFO Used ..... 3506
22-15. Transmit/Receive Mode With Both FIFO Direction Used ..... 3506
22-16. Transmit-Only Mode With FIFO Used ..... 3507
22-17. Receive-Only Mode With FIFO Used ..... 3507
22-18. Buffer Almost Full Level (AFL) ..... 3508
22-19. Buffer Almost Empty Level (AEL) ..... 3509
22-20. Master Single Channel Initial Delay ..... 3510
22-21. 3-Pin Mode System Overview ..... 3511
22-22. Example of SPI Slave with One Master and Multiple Slave Devices on Channel 0 ..... 3513
22-23. SPI Half-Duplex Transmission (Receive-Only Slave) ..... 3515
22-24. SPI Half-Duplex Transmission (Transmit-Only Slave) ..... 3516
22-25. McSPI IP Revision Register (MCSPI_HL_REV) ..... 3527
22-26. McSPI IP Hardware Information Register (MCSPI_HL_HWINFO) ..... 3528
22-27. McSPI IP System Configuration Register (MCSPI_HL_SYSCONFIG) ..... 3529
22-28. McSPI Revision Register (MCSPI_REVISION) ..... 3530
22-29. McSPI System Configuration Register (MCSPI_SYSCONFIG) ..... 3531
22-30. McSPI System Status Register (MCSPI_SYSSTATUS) ..... 3532
22-31. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) ..... 3533
22-32. McSPI Interrupt Enable Register (MCSPI_IRQENABLE) ..... 3536
22-33. McSPI Wakeup Enable Register (MCSPI_WAKEUPENABLE) ..... 3538


Copyright © 2011–2016, Texas Instruments Incorporated

22-34. McSPI System Test Register (MCSPI_SYST) ..... 3539
22-35. McSPI Module Control Register (MCSPI_MODULCTRL) ..... 3541
22-36. McSPI Channel *(i* ) Configuration Register (MCSPI_CH(i)CONF) ..... 3543
22-37. McSPI Channel *(i)* Status Register (MCSPI_CH(i)STAT) ..... 3547
22-38. McSPI Channel *(i*) Control Register (MCSPI_CH(I)CTRL) ..... 3548
22-39. McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*) ..... 3549
22-40. McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*) ..... 3549
22-41. McSPI Transfer Levels Register (MCSPI_XFERLEVEL) ..... 3550
22-42. DMA Address Aligned FIFO Transmitter Register (MCSPI_DAFTX) ..... 3551
22-43. DMA Address Aligned FIFO Receiver Register (MCSPI_DAFRX) ..... 3551
23-1. Timer Block Diagram ..... 3554
23-2. GP Timers External System Interface ..... 3555
23-3. TCRR Timing Value ..... 3556
23-4. Capture Wave Example for CAPT_MODE = 0 ..... 3557
23-5. Capture Wave Example for CAPT_MODE = 1 ..... 3558
23-6. Timing Diagram of Pulse-Width Modulation with SCPWM = 0 ..... 3559
23-7. Timing Diagram of Pulse-Width Modulation with SCPWM = 1 ..... 3560
23-8. TIDR Register ..... 3566
23-9. TIOCP_CFG Register ..... 3567
23-10. IRQ_EOI Register ..... 3568
23-11. IRQSTATUS_RAW Register ..... 3569
23-12. IRQSTATUS Register ..... 3570
23-13. IRQENABLE_SET Register ..... 3571
23-14. IRQENABLE_CLR Register ..... 3572
23-15. IRQWAKEEN Register ..... 3573
23-16. TCLR Register ..... 3573
23-17. TCRR Register ..... 3575
23-18. TLDR Register ..... 3575
23-19. TTGR Register ..... 3576
23-20. TWPS Register ..... 3576
23-21. TMAR Register ..... 3577
23-22. TCAR1 Register ..... 3577
23-23. TSICR Register ..... 3578
23-24. TCAR2 Register ..... 3578
24-1. UART to UART Connection With Full Handshaking ..... 3583
24-2. UART Frame Data Format ..... 3583
24-3. UART IrDA to External IR Device ..... 3587
24-4. IrDA SIR Frame Format ..... 3588
24-5. IrDA SIR Encoding Mechanism ..... 3589
24-6. IrDA SIR Decoding Mechanism ..... 3590
24-7. SIR Free Format Mode ..... 3591
24-8. MIR Transmit Frame Format ..... 3593
24-9. MIR Baud Rate Adjustment Mechanism ..... 3594
24-10. Serial Infrared Interaction Pulse (SIP) ..... 3594
24-11. RC-5 Bit Encoding ..... 3597
24-12. SIRC Bit Encoding ..... 3597
24-13. RC-5 Standard Packet Format ..... 3598
24-14. SIRC Packet Format ..... 3598
24-15. SIRC Bit Transmission Example ..... 3598


Copyright © 2011–2016, Texas Instruments Incorporated

24-16. CIR Pulse Modulation ..... 3599
24-17. CIR Modulation Duty Cycle ..... 3600
24-18. FIFO Management ..... 3601
24-19. Receive FIFO Interrupt Request Generation ..... 3602
24-20. Transmit FIFO Interrupt Request Generation ..... 3602
24-21. Receive FIFO DMA Request Generation (32 Characters) ..... 3604
24-22. Transmit FIFO DMA Request Generation (56 Spaces) ..... 3604
24-23. Transmit FIFO DMA Request Generation (8 Spaces) ..... 3605
24-24. Transmit FIFO DMA Request Generation (1 Space) ..... 3605
24-25. Transmission Process ..... 3606
24-26. Reception Process ..... 3606
24-27. BAUD Rate Generator ..... 3611
24-28. Receiver Holding Register (RHR) ..... 3615
24-29. Transmit Holding Register (THR) ..... 3615
24-30. UART Interrupt Enable Register (IER) ..... 3616
24-31. IrDA Interrupt Enable Register (IER) ..... 3617
24-32. CIR Interrupt Enable Register (IER) ..... 3618
24-33. UART Interrupt Identification Register (IIR) ..... 3619
24-34. IrDA Interrupt Identification Register (IIR) ..... 3620
24-35. CIR Interrupt Identification Register (IIR) ..... 3621
24-36. FIFO Control Register (FCR) ..... 3622
24-37. Line Control Register (LCR) ..... 3623
24-38. Modem Control Register (MCR) ..... 3624
24-39. UART Line Status Register (LSR) ..... 3625
24-40. IrDA Line Status Register (LSR) ..... 3626
24-41. CIR Line Status Register (LSR) ..... 3627
24-42. Modem Status Register (MSR) ..... 3628
24-43. Transmission Control Register (TCR) ..... 3629
24-44. Scratchpad Register (SPR) ..... 3629
24-45. Trigger Level Register (TLR) ..... 3630
24-46. Mode Definition Register 1 (MDR1) ..... 3631
24-47. Mode Definition Register 2 (MDR2) ..... 3632
24-48. Mode Definition Register 3 (MDR3) ..... 3633
24-49. Status FIFO Line Status Register (SFLSR) ..... 3634
24-50. RESUME Register ..... 3634
24-51. Status FIFO Register Low (SFREGL) ..... 3635
24-52. Status FIFO Register High (SFREGH) ..... 3635
24-53. BOF Control Register (BLR) ..... 3636
24-54. Auxiliary Control Register (ACREG) ..... 3637
24-55. Supplementary Control Register (SCR) ..... 3638
24-56. Supplementary Status Register (SSR) ..... 3639
24-57. BOF Length Register (EBLR) ..... 3640
24-58. Module Version Register (MVR) ..... 3641
24-59. System Configuration Register (SYSC) ..... 3642
24-60. System Status Register (SYSS) ..... 3642
24-61. Wake-Up Enable Register (WER) ..... 3643
24-62. Carrier Frequency Prescaler Register (CFPS) ..... 3644
24-63. Divisor Latches Low Register (DLL) ..... 3645
24-64. Divisor Latches High Register (DLH) ..... 3645


Copyright © 2011–2016, Texas Instruments Incorporated

24-65. Enhanced Feature Register (EFR) ..... 3646
24-66. XON1/ADDR1 Register ..... 3647
24-67. XON2/ADDR2 Register ..... 3647
24-68. XOFF1 Register ..... 3648
24-69. XOFF2 Register ..... 3648
24-70. Transmit Frame Length Low Register (TXFLL) ..... 3649
24-71. Transmit Frame Length High Register (TXFLH) ..... 3649
24-72. Received Frame Length Low Register (RXFLL) ..... 3650
24-73. Received Frame Length High Register (RXFLH) ..... 3650
24-74. UART Autobauding Status Register (UASR) ..... 3651
25-1. USB Subsystem Block Diagram ..... 3654
25-2. CPU Actions at Transfer Phases ..... 3669
25-3. TX State ..... 3669
25-4. RX State ..... 3670
25-5. No Data Phase ..... 3670
25-6. Flow Chart of Setup Stage of a Control Transfer in Peripheral Mode ..... 3672
25-7. Flow Chart of Transmit Data Stage of a Control Transfer in Peripheral Mode ..... 3673
25-8. Flow Chart of Receive Data Stage of a Control Transfer in Peripheral Mode ..... 3674
25-9. Flow Chart of Setup Stage of a Control Transfer in Host Mode ..... 3684
25-10. Flow Chart of Data Stage (IN Data Phase) of a Control Transfer in Host Mode ..... 3686
25-11. Flow Chart of Data Stage (OUT Data Phase) of a Control Transfer in Host Mode ..... 3688
25-12. Flow Chart of Status Stage of Zero Data Request or Write Request of a Control Transfer in Host Mode ..... 3690
25-13. Chart of Status Stage of a Read Request of a Control Transfer in Host Mode ..... 3692
25-14. Packet Descriptor Layout ..... 3701
25-15. Buffer Descriptor (BD) Layout ..... 3705
25-16. Teardown Descriptor Layout ..... 3708
25-17. Relationship Between Memory Regions and Linking RAM ..... 3715
25-18. High-level Transmit and Receive Data Transfer Example ..... 3720
25-19. Transmit Descriptors and Queue Status Configuration ..... 3721
25-20. Transmit USB Data Flow Example (Initialization) ..... 3722
25-21. Receive Buffer Descriptors and Queue Status Configuration ..... 3724
25-22. Receive USB Data Flow Example (Initialization) ..... 3725
25-23. USBSS Revision Register (REVREG) ..... 3727
25-24. USBSS SYSCONFIG Register (SYSCONFIG) ..... 3728
25-25. USBSS End of Interrupt Register (EOI) ..... 3730
25-26. USBSS IRQ_STATUS_RAW (IRQSTATRAW) ..... 3731
25-27. USBSS IRQ_STATUS (IRQSTAT) ..... 3732
25-28. USBSS IRQ_ENABLE_SET Register (IRQENABLER) ..... 3733
25-29. USBSS IRQ_ENABLE_CLR Register (IRQCLEARR) ..... 3734
25-30. USBSS IRQ_DMA_THRESHOLD_TX0_0 Register (IRQDMATHOLDTX00) ..... 3735
25-31. USBSS IRQ_DMA_THRESHOLD_TX0_1 Register (IRQDMATHOLDTX01) ..... 3735
25-32. USBSS IRQ_DMA_THRESHOLD_TX0_2 Register (IRQDMATHOLDTX02) ..... 3736
25-33. USBSS IRQ_DMA_THRESHOLD_TX0_3 Register (IRQDMATHOLDTX03) ..... 3736
25-34. USBSS IRQ_DMA_THRESHOLD_RX0_0 Register (IRQDMATHOLDRX00) ..... 3737
25-35. USBSS IRQ_DMA_THRESHOLD_RX0_1 Register (IRQDMATHOLDRX01) ..... 3737
25-36. USBSS IRQ_DMA_THRESHOLD_RX0_2 Register (IRQDMATHOLDRX02) ..... 3738
25-37. USBSS IRQ_DMA_THRESHOLD_RX0_3 Register (IRQDMATHOLDRX03) ..... 3738
25-38. USBSS IRQ_DMA_THRESHOLD_TX1_0 Register (IRQDMATHOLDTX10) ..... 3739
25-39. USBSS IRQ_DMA_THRESHOLD_TX1_1 Register (IRQDMATHOLDTX11) ..... 3739


Copyright © 2011–2016, Texas Instruments Incorporated

25-40. USBSS IRQ_DMA_THRESHOLD_TX1_2 Register (IRQDMATHOLDTX12) ..... 3740
25-41. USBSS IRQ_DMA_THRESHOLD_TX1_3 Register (IRQDMATHOLDTX13) ..... 3740
25-42. USBSS IRQ_DMA_THRESHOLD_RX1_0 Register (IRQDMATHOLDRX10) ..... 3741
25-43. USBSS IRQ_DMA_THRESHOLD_RX1_1 Register (IRQDMATHOLDRX11) ..... 3741
25-44. USBSS IRQ_DMA_THRESHOLD_RX1_2 Register (IRQDMATHOLDRX12) ..... 3742
25-45. USBSS IRQ_DMA_THRESHOLD_RX1_3 Register (IRQDMATHOLDRX13) ..... 3742
25-46. USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0) ..... 3743
25-47. USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1) ..... 3745
25-48. USBSS IRQ_FRAME_THRESHOLD_TX0_0 Register (IRQFRAMETHOLDTX00) ..... 3747
25-49. USBSS IRQ_FRAME_THRESHOLD_TX0_1 Register (IRQFRAMETHOLDTX01) ..... 3747
25-50. USBSS IRQ_FRAME_THRESHOLD_TX0_2 Register (IRQFRAMETHOLDTX02) ..... 3748
25-51. USBSS IRQ_FRAME_THRESHOLD_TX0_3 Register (IRQFRAMETHOLDTX03) ..... 3748
25-52. USBSS IRQ_FRAME_THRESHOLD_RX0_0 Register (IRQFRAMETHOLDRX00) ..... 3749
25-53. USBSS IRQ_FRAME_THRESHOLD_RX0_1 Register (IRQFRAMETHOLDRX01) ..... 3749
25-54. USBSS IRQ_FRAME_THRESHOLD_RX0_2 Register (IRQFRAMETHOLDRX02) ..... 3750
25-55. USBSS IRQ_FRAME_THRESHOLD_RX0_3 Register (IRQFRAMETHOLDRX03) ..... 3750
25-56. USBSS IRQ_FRAME_THRESHOLD_TX1_0 Register (IRQFRAMETHOLDTX10) ..... 3751
25-57. USBSS IRQ_FRAME_THRESHOLD_TX1_1 Register (IRQFRAMETHOLDTX11) ..... 3751
25-58. USBSS IRQ_FRAME_THRESHOLD_TX1_2 Register (IRQFRAMETHOLDTX12) ..... 3752
25-59. USBSS IRQ_FRAME_THRESHOLD_TX1_3 Register (IRQFRAMETHOLDTX13) ..... 3752
25-60. USBSS IRQ_FRAME_THRESHOLD_RX1_0 Register (IRQFRAMETHOLDRX10) ..... 3753
25-61. USBSS IRQ_FRAME_THRESHOLD_RX1_1 Register (IRQFRAMETHOLDRX11) ..... 3753
25-62. USBSS IRQ_FRAME_THRESHOLD_RX1_2 Register (IRQFRAMETHOLDRX12) ..... 3754
25-63. USBSS IRQ_FRAME_THRESHOLD_RX1_3 Register (IRQFRAMETHOLDRX13) ..... 3754
25-64. USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0) ..... 3755
25-65. USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1) ..... 3757
25-66. USB0 Revision Register (USB0REV) ..... 3760
25-67. USB0 Control Register (USB0CTRL) ..... 3761
25-68. USB0 Status Register (USB0STAT) ..... 3762
25-69. USB0 IRQ_MERGED_STATUS Register (USB0IRQMSTAT) ..... 3762
25-70. USB0 IRQ_EOI Register (USB0IRQEOI) ..... 3763
25-71. USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0) ..... 3764
25-72. USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1) ..... 3766
25-73. USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0) ..... 3768
25-74. USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1) ..... 3770
25-75. USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0) ..... 3772
25-76. USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1) ..... 3774
25-77. USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0) ..... 3776
25-78. USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1) ..... 3778
25-79. USB0 Tx Mode Register (USB0TXMODE) ..... 3780
25-80. USB0 Rx Mode Register (USB0RXMODE) ..... 3782
25-81. USB0 Generic RNDIS EPn Size Register (USB0GENRNDISEPn) ..... 3784
25-82. USB0 Auto Req Register (USB0AUTOREQ) ..... 3785
25-83. USB0 SRP Fix Time Register (USB0SRPFIXTIME) ..... 3788
25-84. USB0 Teardown Register (USB0TDOWN) ..... 3788
25-85. USB0 Threshold XDMA Idle Register ..... 3789
25-86. USB0 PHY UTMI Register (USB0UTMI) ..... 3790
25-87. USB0 MGC UTMI Loopback Register (USB0UTMILB) ..... 3791
25-88. USB0 Mode Register (USB0MODE) ..... 3792


Copyright © 2011–2016, Texas Instruments Incorporated

25-89. USB1 Revision Register (USB1REV) ..... 3794
25-90. USB1 Control Register (USB1CTRL) ..... 3795
25-91. USB1 Status Register (USB1STAT) ..... 3796
25-92. USB1 IRQ_MERGED_STATUS Register (USB1IRQMSTAT) ..... 3796
25-93. USB1 IRQ_EOI Register (USB1IRQEOI) ..... 3797
25-94. USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW0) ..... 3798
25-95. USB1 IRQ_STATUS_RAW_1 Register (USB1IRQSTATRAW1) ..... 3800
25-96. USB1 IRQ_STATUS_0 Register (USB1IRQSTAT0) ..... 3802
25-97. USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1) ..... 3804
25-98. USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0) ..... 3806
25-99. USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1) ..... 3808
25-100. USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0) ..... 3810
25-101. USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1) ..... 3812
25-102. USB1 Tx Mode Register (USB1TXMODE) ..... 3814
25-103. USB1 Rx Mode Register (USB1RXMODE) ..... 3816
25-104. USB1 Generic RNDIS EP N Size Register (USB1GENRNDISEPn) ..... 3818
25-105. USB1 Auto Req Register (USB1AUTOREQ) ..... 3819
25-106. USB1 SRP Fix Time Register (USB1SRPFIXTIME) ..... 3822
25-107. USB1 Teardown Register (USB1TDOWN) ..... 3822
25-108. USB1 Threshold XDMA Idle Register ..... 3823
25-109. USB1 PHY UTMI Register (USB1UTMI) ..... 3824
25-110. USB1 MGC UTMI Loopback Register (USB1UTMILB) ..... 3825
25-111. USB1 Mode Register (USB1MODE) ..... 3826
25-112. CPPI DMA Revision Register (DMAREVID) ..... 3828
25-113. CPPI DMA Teardown Free Descriptor Queue Control Register (TDFDQ) ..... 3828
25-114. CPPI DMA Emulation Control Register (DMAEMU) ..... 3829
25-115. Tx Channel N Global Configuration Register (TXGCRn) ..... 3829
25-116. Rx Channel N Global Configuration Register (RXGCRn) ..... 3830
25-117. Rx Channel N Host Packet Configuration Register A (RXHPCRAn) ..... 3832
25-118. Rx Channel N Host Packet Configuration Register B (RXHPCRBn) ..... 3833
25-119. CPPI DMA Scheduler Control Register (DMA_SCHED_CTRL) ..... 3834
25-120. CPPI DMA Scheduler Table Word N (WORDn) ..... 3835
25-121. Queue Manager Revision Register (QMGRREVID) ..... 3838
25-122. Queue Manager Queue Diversion Register (DIVERSION) ..... 3838
25-123. Queue Manager Free Descriptor/Buffer Starvation Count Register 0 (FDBSC0) ..... 3839
25-124. Queue Manager Free Descriptor/Buffer Starvation Count Register 1 (FDBSC1) ..... 3839
25-125. Queue Manager Free Descriptor/Buffer Starvation Count Register 2 (FDBSC2) ..... 3840
25-126. Queue Manager Free Descriptor/Buffer Starvation Count Register 3 (FDBSC3) ..... 3840
25-127. Queue Manager Free Descriptor/Buffer Starvation Count Register 4 (FDBSC4) ..... 3841
25-128. Queue Manager Free Descriptor/Buffer Starvation Count Register 5 (FDBSC5) ..... 3841
25-129. Queue Manager Free Descriptor/Buffer Starvation Count Register 6 (FDBSC6) ..... 3842
25-130. Queue Manager Free Descriptor/Buffer Starvation Count Register 7 (FDBSC7) ..... 3842
25-131. Queue Manager Linking RAM Region 0 Base Address Register (LRAM0BASE) ..... 3843
25-132. Queue Manager Linking RAM Region 0 Size Register (LRAM0SIZE) ..... 3843
25-133. Queue Manager Linking RAM Region 1 Base Address Register (LRAM1BASE) ..... 3844
25-134. Queue Manager Queue Pending Register 0 (PEND0) ..... 3844
25-135. Queue Manager Queue Pending Register 1 (PEND1) ..... 3845
25-136. Queue Manager Queue Pending Register 2 (PEND2) ..... 3845
25-137. Queue Manager Queue Pending Register 3 (PEND3) ..... 3845


Copyright © 2011–2016, Texas Instruments Incorporated

25-138. Queue Manager Queue Pending Register 4 (PEND4) ..... 3846
25-139. Queue Manager Memory Region R Base Address Register (QMEMRBASEr) ..... 3846
25-140. Queue Manager Memory Region R Control Register (QMEMRCTRLr) ..... 3847
25-141. Queue Manager Queue N Register A (CTRLAn) ..... 3848
25-142. Queue Manager Queue N Register B (CTRLBn) ..... 3848
25-143. Queue Manager Queue N Register C (CTRLCn) ..... 3849
25-144. Queue Manager Queue N Register D (CTRLDn) ..... 3849
25-145. Queue Manager Queue N Register A (QSTATAn) ..... 3850
25-146. Queue Manager Queue N Status Register B (QSTATBn) ..... 3850
25-147. Queue Manager Queue N Status Register C (QSTATCn) ..... 3851
25-148. Function Address Register (USBn_FADDR) ..... 3853
25-149. Power Management Register (USBn_POWER) ..... 3853
25-150. Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 (USBn_INTRTX) ..... 3854
25-151. Interrupt Register for Receive Endpoints 1 to 15 (INTRRX) ..... 3855
25-152. Interrupt Enable Register for INTRTX (INTRTXE) ..... 3856
25-153. Interrupt Enable Register for INTRRX (INTRRXE) ..... 3857
25-154. Interrupt Register for Common USB Interrupts (USBn_INTRUSB) ..... 3858
25-155. Interrupt Enable Register for INTRUSB (USBn_INTRUSBE) ..... 3858
25-156. Frame Number Register (USBn_FRAME) ..... 3859
25-157. Index Register for Selecting the Endpoint Status and Control Registers (USBn_INDEX) ..... 3859
25-158. Register to Enable the USB 2.0 Test Modes (USBn_TESTMODE) ..... 3860
25-159. Maximum Packet Size for Peripheral/Host Transmit Endpoint Register (USBn_TXMAXP) ..... 3861
25-160. Control Status Register for Endpoint 0 in Peripheral Mode (USBn_PERI_CSR0) ..... 3862
25-161. Control Status Register for Endpoint 0 in Host Mode (USBn_HOST_CSR0) ..... 3863
25-162. Control Status Register for Peripheral Transmit Endpoint (USBn_PERI_TXCSR) ..... 3864
25-163. Control Status Register for Host Transmit Endpoint (USBn_HOST_TXCSR) ..... 3865
25-164. Maximum Packet Size for Peripheral/Host Receive Endpoint Register (USBn_RXMAXP) ..... 3866
25-165. Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR) ..... 3867
25-166. Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR) ..... 3869
25-167. Count 0 Register (USBn_COUNT0) ..... 3870
25-168. Receive Count Register (USBn_RXCOUNT) ..... 3871
25-169. Type Register (Host Mode Only) (USBn_HOST_TYPE0) ..... 3871
25-170. Transmit Type Register (Host Mode Only) (USBn_HOST_TXTYPE) ..... 3872
25-171. NAKLimit0 Register (Host Mode Only) (USBn_HOST_NAKLIMIT0) ..... 3873
25-172. Transmit Interval Register (Host Mode Only) (USBn_HOST_TXINTERVAL) ..... 3874
25-173. Receive Type Register (Host Mode Only) (USBn_HOST_RXTYPE) ..... 3875
25-174. Receive Interval Register (Host Mode Only) (USBn_HOST_RXINTERVAL) ..... 3876
25-175. Configuration Data Register (USBn_CONFIGDATA) ..... 3877
25-176. Transmit and Receive FIFO Register for Endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15) ..... 3878
25-177. Device Control Register (USBn_DEVCTL) ..... 3879
25-178. Transmit Endpoint FIFO Size Register (USBn_TXFIFOSZ) ..... 3880
25-179. Receive Endpoint FIFO Size Register (USBn_RXFIFOSZ) ..... 3881
25-180. Transmit Endpoint FIFO Address Register (USBn_TXFIFOADDR) ..... 3881
25-181. Receive Endpoint FIFO Address Register (USBn_RXFIFOADDR) ..... 3882
25-182. Hardware Version Register (USBn_HWVERS) ..... 3882
25-183. Transmit Function Address Register (USBn_TXFUNCADDRm) ..... 3883
25-184. Transmit Hub Address Register (USBn_TXHUBADDRm) ..... 3884
25-185. Transmit Hub Port Register (USBn_TXHUBPORTm) ..... 3884
25-186. Receive Function Address Register (USBn_RXFUNCADDRm) ..... 3885


Copyright © 2011–2016, Texas Instruments Incorporated

25-187. Receive Hub Address Register (USBn_RXHUBADDRm) ..... 3885
25-188. Receive Hub Port Register (USBn_RXHUBPORTm) ..... 3886
26-1. Watchdog Timer Block Diagram ..... 3889
26-2. 32-Bit Watchdog Timer Functional Block Diagram ..... 3890
26-3. Watchdog Timers General Functional View ..... 3891
26-4. WDT_WIDR Register ..... 3899
26-5. WDT_WDSC Register ..... 3899
26-6. WDT_WDST Register ..... 3900
26-7. WDT_WISR Register ..... 3900
26-8. WDT_WIER Register ..... 3901
26-9. WDT_WCLR Register ..... 3901
26-10. WDT_WCRR Register ..... 3902
26-11. WDT_WLDR Register ..... 3902
26-12. WDT_WTGR Register ..... 3902
26-13. WDT_WWPS Register ..... 3903
26-14. WDT_WDLY ..... 3904
26-15. WDT_WSPR Register ..... 3904
26-16. WDT_WIRQSTATRAW Register ..... 3905
26-17. WDT_WIRQSTAT Register ..... 3906
26-18. WDT_WIRQENSET Register ..... 3907
26-19. WDT_WIRQENCLR Register ..... 3908


Copyright © 2011–2016, Texas Instruments Incorporated

# List of Tables


1-1. Integration Attributes ..... 160
1-2. Hardware Requests ..... 160
1-3. Clocks ..... 160
1-4. Resets ..... 161
1-5. Shared Cache Configuration ..... 164
1-6. Shared Cache MMU Configuration ..... 164
1-7. Registers ..... 166
1-8. Configuration Registers (CACHE_CFG) Field Descriptions ..... 167
1-9. Interrupt Register (CACHE_INT) Field Descriptions ..... 167
1-10. Interface Configuration Register (CACHE_OCP) Field Descriptions ..... 167
1-11. Maintenance Configuration Register (CACHE_MAINT) Field Descriptions ..... 168
1-12. Maintenance Start Configuration Register (CACHE_MTSTART) Field Descriptions ..... 168
1-13. Maintenance End Configuration Register (CACHE_MTEND) Field Descriptions ..... 169
1-14. Cache Test Address Register (CACHE_CTADDR) Field Descriptions ..... 169
1-15. Cache Test Data Register (CACHE_CTDATA) Field Descriptions ..... 169
1-16. Counter Timer Control Register (CACHE_SCTM_CTCNTL) Field Descriptions ..... 170
1-17. Counter Timer STM Control Register (CACHE_SCTM_CTSTMCNTL) Field Descriptions ..... 170
1-18. Counter Timer STM Master ID Register (CACHE_SCTM_CTSTMMSTID) Field Descriptions ..... 171
1-19. Counter Timer STM Interval Register (CACHE_SCTM_CTSTMINTVL) Field Descriptions ..... 171
1-20. Counter Timer STM Count Select Register 0(CACHE_SCTM_CTSTMSEL0) Field Descriptions ..... 171
1-21. Counter Timer Interval Number Debug Event Register (CACHE_SCTM_TINTVL_R_i) Field Descriptions ..... 171
1-22. Counter Timer Number Debug Event Register (CACHE_SCTM_CTDBGNUM) Field Descriptions ..... 171
1-23. Counter Timer Global Enable Register 0(CACHE_SCTM_CTGNBL0) Field Description ..... 172
1-24. Counter Timer Global Reset Register 0(CACHE_SCTM_CTGRST0) Field Descriptions ..... 172
1-25. Counter Timer Control Register 0-31(CACHE_SCTM_CTCRWT_i) Field Descriptions ..... 172
1-26. Counter Timer Count Register 0-31(CACHE_SCTM_CTCNTR_k) Field Descriptions ..... 172
1-27. Large Page Address (CACHE_MMU_LARGE_ADDR_l) Field Descriptions ..... 173
1-28. Large Page Translated Address (CACHE_MMU_LARGE_XLTE_l) Field Descriptions ..... 173
1-29. Large Page Policy (CACHE_MMU_LARGE_POLY_l) Field Descriptions ..... 173
1-30. Medium Page Address (CACHE_MMU_MED_ADDR_m) Field Descriptions ..... 174
1-31. Medium Page Translated Address (CACHE_MMU_MED_XLTE_m) Field Descriptions ..... 174
1-32. Medium Page Policy (CACHE_MMU_MED_POLY_m) Field Descriptions ..... 174
1-33. Small Page Address (CACHE_MMU_SMALL_ADDR_n) Field Descriptions ..... 176
1-34. Small Page Translated Address (CACHE_MMU_SMALL_XLTE_n) Field Descriptions ..... 176
1-35. Small Page Policy (CACHE_MMU_SMALL_POLY_n) Field Descriptions ..... 176
1-36. Small Page Maintenance Configuration (CACHE_MMU_SMALL_MAINT_n) Field Descriptions ..... 177
1-37. MMU Start/End Maintenance Config Register (CACHE_MMU_MAINT) Field Descriptions ..... 177
1-38. Maintenance Start Address Register (CACHE_MMU_MTSTART) Field Descriptions ..... 179
1-39. Maintenance End Address Register (CACHE_MMU_MTEND) Field Descriptions ..... 179
1-40. Maintenance Status Register (CACHE_MMU_MAINTST) Field Descriptions ..... 179
1-41. MMU Configuration Register (CACHE_MMU_MMUCONFIG) Field Descriptions ..... 179
1-42. First-Level Descriptor Format ..... 193
1-43. Second-Level Descriptor Format ..... 197
1-44. MMU Local Power Management Features ..... 201
1-45. Events ..... 201
1-46. Error Handling ..... 201
1-47. Configure a TLB Entry ..... 202



Copyright © 2011–2016, Texas Instruments Incorporated

1-48. MMU Writing TLB Entries Statically ..... 203
1-49. Protecting TLB Entries ..... 203
1-50. Deleting TLB Entries ..... 204
1-51. Read TLB Entries ..... 204
1-52. MMU Instance Summary ..... 205
1-53. MMU Registers ..... 205
1-54. MMU_REVISION Field Descriptions ..... 205
1-55. MMU_SYSCONFIG Field Descriptions ..... 206
1-56. MMU_SYSSTATUS Field Descriptions ..... 207
1-57. MMU_IRQSTATUS Field Descriptions ..... 208
1-58. MMU_IRQENABLE Field Descriptions ..... 209
1-59. MMU_WALKING_ST Field Descriptions ..... 210
1-60. MMU_CNTL Field Descriptions ..... 210
1-61. MMU_FAULT_AD Field Descriptions ..... 211
1-62. MMU_TTB Field Descriptions ..... 211
1-63. MMU_LOCK Field Descriptions ..... 211
1-64. MMU_LD_TLB Field Descriptions ..... 212
1-65. MMU_CAM Field Descriptions ..... 212
1-66. MMU_RAM Field Descriptions ..... 213
1-67. MMU_GFLUSH Field Descriptions ..... 213
1-68. MMU_FLUSH_ENTRY Field Descriptions ..... 214
1-69. MMU_READ_CAM Field Descriptions ..... 214
1-70. MMU_READ_RAM Field Descriptions ..... 215
1-71. MMU_EMU_FAULT_AD Field Descriptions ..... 216
1-72. MMU_FAULT_PC Field Descriptions ..... 217
1-73. Clock Descriptions ..... 221
1-74. SGX Registers Mapping Summary ..... 225
1-75. OCP Revision Register (OCP_REVISION) Field Descriptions ..... 226
1-76. Hardware Implementation Information Register (OCP_HWINFO) Field Descriptions ..... 226
1-77. System Configuration Register (OCP_SYSCONFIG) Field Descriptions ..... 227
1-78. Raw IRQ 0 Status Register (OCP_IRQSTATUS_RAW_0) Field Descriptions ..... 228
1-79. Raw IRQ 1 Status Register (OCP_IRQSTATUS_RAW_1) Field Descriptions ..... 228
1-80. Raw IRQ 2 Status Register (OCP_IRQSTATUS_RAW_2) Field Descriptions ..... 229
1-81. Interrupt 0 Status Event Register (OCP_IRQSTATUS_0) Field Descriptions ..... 229
1-82. Interrupt 1 Status Event Register (OCP_IRQSTATUS_1) Field Descriptions ..... 230
1-83. Interrupt 2 Status Event Register (OCP_IRQSTATUS_2) Field Descriptions ..... 230
1-84. Enable Interrupt 0 Register (OCP_IRQENABLE_SET_0) Field Descriptions ..... 231
1-85. Enable Interrupt 1 Register (OCP_IRQENABLE_SET_1) Field Descriptions ..... 231
1-86. Enable Interrupt 2 Register (OCP_IRQENABLE_SET_2) Field Descriptions ..... 232
1-87. Disable Interrupt 0 Register (OCP_IRQENABLE_CLR_0) Field Descriptions ..... 232
1-88. Disable Interrupt 1 Register (OCP_IRQENABLE_CLR_1) Field Descriptions ..... 233
1-89. Disable Interrupt 2 Register (OCP_IRQENABLE_CLR_2) Field Descriptions ..... 233
1-90. Configure Memory Page Register (OCP_PAGE_CONFIG) Field Descriptions ..... 234
1-91. Interrupt Events Register (OCP_INTERRUPT_EVENT) Field Descriptions ..... 235
1-92. Configuration of Debug Modes Register (OCP_DEBUG_CONFIG) Field Descriptions ..... 237
1-93. Debug Status Register (OCP_DEBUG_STATUS) Field Descriptions ..... 238
1-94. Cortex-A8 MPU INTC Interrupt Mapping ..... 241
1-95. C674x INTC Default Interrupt Mapping ..... 244
1-96. C674x INTC Interrupt Crossbar Mapping ..... 248


Copyright © 2011–2016, Texas Instruments Incorporated

1-97. Media Controller INTC Default Interrupt Mapping ..... 248
1-98. Media Controller INTC Interrupt Crossbar Mapping ..... 251
1-99. EDMA Default Synchronization Events ..... 252
1-100. EDMA Multiplexed Synchronization Events ..... 253
1-101. Hardware Spinlock Configuration ..... 259
1-102. Integration Attributes ..... 261
1-103. Clocks and Resets ..... 261
1-104. Hardware Requests ..... 261
1-105. Mailbox Implementation ..... 262
1-106. Local Power Management Features ..... 263
1-107. Interrupt Events ..... 264
1-108. Global Initialization of Surrounding Modules for System Mailbox ..... 266
1-109. Global Initialization of Surrounding Modules for HDVICP2 Mailbox ..... 266
1-110. Global Initialization of Surrounding Modules for HDVICP2-1 Mailbox ..... 267
1-111. Global Initialization of Surrounding Modules for HDVICP2-2 Mailbox ..... 267
1-112. Mailbox Global Initialization ..... 268
1-113. Sending a Message (Polling Method) ..... 268
1-114. Sending a Message (Interrupt Method) ..... 268
1-115. Receiving a Message (Polling Method) ..... 269
1-116. Receiving a Message (Interrupt Method) ..... 269
1-117. Events Servicing in Sending Mode ..... 269
1-118. Events Servicing in Receiving Mode ..... 269
1-119. Mailboxes ..... 270
1-120. Mailbox Registers Mapping Summary ..... 270
1-121. Revision Register (MAILBOX_REVISION) Field Descriptions ..... 271
1-122. System Configuration Register (MAILBOX_SYSCONFIG) Field Descriptions ..... 271
1-123. Message Register (MAILBOX_MESSAGE_m) Field Descriptions ..... 272
1-124. FIFO Status Register (MAILBOX_FIFOSTATUS_m) Field Descriptions ..... 272
1-125. Message Status Register (MAILBOX_MSGSTATUS_m) Field Descriptions ..... 273
1-126. IRQ RAW Status Register (MAILBOX_IRQSTATUS_RAW_u) Field Descriptions ..... 274
1-127. IRQ Clear Status Register (MAILBOX_IRQSTATUS_CLR_u) Field Descriptions ..... 278
1-128. IRQ Enable Set Register (MAILBOX_IRQENABLE_SET_u) Field Descriptions ..... 282
1-129. IRQ Enable Clear Register (MAILBOX_IRQENABLE_CLR_u) Field Descriptions ..... 286
1-130. Integration Attributes ..... 291
1-131. Clocks and Resets ..... 291
1-132. Local Power Management Features ..... 292
1-133. Global Initialization of Surrounding Modules ..... 294
1-134. Spinlock System Bug Recovery ..... 294
1-135. Spin Lock Module Registers Offset ..... 296
1-136. Revision Register (SPINLOCK_REV) Field Descriptions ..... 296
1-137. System Configuration Register (SPINLOCK_SYS_CFG) Field Descriptions ..... 297
1-138. System Status Register (SPINLOCK_SYSSTAT) Field Descriptions ..... 298
1-139. Lock Register (SPINLOCK_LOCK_REG_i) Field Descriptions ..... 299
1-140. Integration Attributes ..... 301
1-141. Clocks and Resets ..... 302
1-142. Hardware Requests ..... 302
1-143. Local Power Management Features ..... 302
1-144. Events ..... 303
1-145. ELM_LOCATION_STATUS_i Value Decoding Table ..... 304


Copyright © 2011–2016, Texas Instruments Incorporated

1-146. ELM Processing Initialization ..... 305
1-147. ELM Processing Completion for Continuous Mode ..... 306
1-148. ELM Processing Completion for Page Mode ..... 306
1-149. Use Case: Continuous Mode ..... 307
1-150. 16-bit NAND Sector Buffer Address Map ..... 309
1-151. Use Case: Page Mode ..... 309
1-152. ELM Registers Mapping Summary ..... 312
1-153. ELM Revision Register (ELM_REVISION) Field Descriptions ..... 313
1-154. ELM System Configuration Register (ELM_SYSCONFIG) Field Descriptions ..... 313
1-155. ELM System Status Register (ELM_SYSSTATUS) Field Descriptions ..... 314
1-156. ELM Interrupt Status Register (ELM_IRQSTATUS) Field Descriptions ..... 315
1-157. ELM Interrupt Enable Register (ELM_IRQENABLE) Field Descriptions ..... 317
1-158. ELM Location Configuration Register (ELM_LOCATION_CONFIG) Field Descriptions ..... 318
1-159. ELM Page Definition Register (ELM_PAGE_CTRL) Field Descriptions ..... 319
1-160. ELM_SYNDROME_FRAGMENT_0_i Register Field Descriptions ..... 320
1-161. ELM_SYNDROME_FRAGMENT_1_i Register Field Descriptions ..... 320
1-162. ELM_SYNDROME_FRAGMENT_2_i Register Field Descriptions ..... 320
1-163. ELM_SYNDROME_FRAGMENT_3_i Register Field Descriptions ..... 321
1-164. ELM_SYNDROME_FRAGMENT_4_i Register Field Descriptions ..... 321
1-165. ELM_SYNDROME_FRAGMENT_5_i Register Field Descriptions ..... 322
1-166. ELM_SYNDROME_FRAGMENT_6_i Register Field Descriptions ..... 322
1-167. ELM_LOCATION_STATUS_i Register Field Descriptions ..... 323
1-168. ELM_ERROR_LOCATION_0-15_i Registers Field Descriptions ..... 324
1-169. MCmd Qualifier Description ..... 325
1-170. MReqInfo Qualifier Description ..... 325
1-171. SResp Qualifier Description ..... 326
1-172. Integration Attributes ..... 330
1-173. Clocks and Resets ..... 330
1-174. Master NIUs ..... 331
1-175. Slave NIUs ..... 332
1-176. L3 Clock Domains and Elements ..... 333
1-177. ConnID Values ..... 336
1-178. L3 Connectivity and Holes Error Routing ..... 340
1-179. Interconnect Flag Mapping ..... 341
1-180. Global Initialization of Surrounding Modules ..... 343
1-181. Subprocess Call Summary for Main Sequence – Error Analysis Mode ..... 345
1-182. Custom Error Identification ..... 345
1-183. Protection Violation Error Identification ..... 345
1-184. L3 Standard Error Identification ..... 346
1-185. FLAGMUX Configuration ..... 346
1-186. HOST Instance Summary ..... 347
1-187. HOST Registers Summary ..... 348
1-188. L3_HOST_STDHOSTHDR_COREREG ..... 349
1-189. L3_HOST_STDHOSTHDR_VERSIONREG ..... 350
1-190. L3_HOST_STDHOSTHDR_MAINCTLREG ..... 350
1-191. L3_HOST_STDERRLOG_SVRTSTDLVL ..... 351
1-192. L3_HOST_STDERRLOG_SVRTCUSTOMLVL ..... 352
1-193. L3_HOST_STDERRLOG_MAIN ..... 352
1-194. L3_HOST_STDERRLOG_HDR ..... 353


Copyright © 2011–2016, Texas Instruments Incorporated

1-195. L3_HOST_STDERRLOG_MSTADDR ..... 354
1-196. L3_HOST_STDERRLOG_SLVADDR ..... 354
1-197. L3_HOST_STDERRLOG_INFO ..... 355
1-198. L3_HOST_STDERRLOG_SLVOFSLSB ..... 355
1-199. L3_HOST_STDERRLOG_SLVOFSMSB ..... 355
1-200. L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR ..... 356
1-201. L3_HOST_STDERRLOG_CUSTOMINFO_INFO ..... 356
1-202. L3_HOST_STDERRLOG_CUSTOMINFO_WR ..... 357
1-203. L3_HOST_STDERRLOG_CUSTOMINFO_ADDR ..... 357
1-204. L3_HOST_STDERRLOG_CUSTOMINFO_DECERR ..... 358
1-205. TARG Instance Summary ..... 358
1-206. TARG Registers Summary ..... 360
1-207. TARG Registers Summary ..... 360
1-208. TARG Registers Summary ..... 361
1-209. TARG Registers Summary ..... 362
1-210. TARG Registers Summary ..... 363
1-211. TARG Registers Summary ..... 363
1-212. TARG Registers Summary ..... 364
1-213. L3_TARG_STDHOSTHDR_COREREG ..... 366
1-214. L3_TARG_STDHOSTHDR_VERSIONREG ..... 367
1-215. L3_TARG_STDHOSTHDR_MAINCTLREG ..... 367
1-216. L3_TARG_STDHOSTHDR_NTTPADDR_0 ..... 368
1-217. L3_TARG_STDERRLOG_SVRTSTDLVL ..... 369
1-218. L3_TARG_STDERRLOG_SVRTCUSTOMLVL ..... 369
1-219. L3_TARG_STDERRLOG_MAIN ..... 370
1-220. L3_TARG_STDERRLOG_HDR ..... 371
1-221. L3_TARG_STDERRLOG_MSTADDR ..... 371
1-222. L3_TARG_STDERRLOG_SLVADDR ..... 372
1-223. L3_TARG_STDERRLOG_INFO ..... 372
1-224. L3_TARG_STDERRLOG_SLVOFSLSB ..... 373
1-225. L3_TARG_STDERRLOG_SLVOFSMSB ..... 373
1-226. L3_TARG_STDERRLOG_CUSTOMINFO_INFO ..... 374
1-227. L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR ..... 374
1-228. L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE ..... 375
1-229. L3_TARG_ADDRSPACESIZELOG ..... 375
1-230. FLAGMUX Instance Summary ..... 376
1-231. FLAGMUX Registers Summary ..... 376
1-232. FLAGMUX Registers Summary ..... 377
1-233. L3_FLAGMUX_STDHOSTHDR_COREREG ..... 378
1-234. L3_FLAGMUX_STDHOSTHDR_VERSIONREG ..... 379
1-235. L3_FLAGMUX_MASK0 ..... 379
1-236. L3_FLAGMUX_REGERR0 ..... 380
1-237. L3_FLAGMUX_MASK1 ..... 380
1-238. L3_FLAGMUX_REGERR1 ..... 380
1-239. Reset Values for FLAGMUX_MASKi ..... 380
1-240. BW_REGULATOR Instance Summary ..... 381
1-241. BW_REGULATOR Registers Summary ..... 382
1-242. BW_REGULATOR Registers Summary ..... 382
1-243. L3_BW_R_STDHOSTHDR_COREREG ..... 383


Copyright © 2011–2016, Texas Instruments Incorporated

1-244. L3_BW_R_STDHOSTHDR_VERSIONREG ..... 384
1-245. L3_BW_R_BANDWIDTH ..... 384
1-246. L3_BW_R_WATERMARK ..... 385
1-247. L3_BW_R_PRESS ..... 385
1-248. L3_BW_R_CLEARHISTORY ..... 385
1-249. STATCOLL Instance Summary ..... 386
1-250. STATCOLL Register Summary ..... 387
1-251. STATCOLL Register Summary ..... 389
1-252. L3_STCOL_STDHOSTHDR_COREREG ..... 391
1-253. L3_STCOL_STDHOSTHDR_VERSIONREG ..... 392
1-254. L3_STCOL_EN ..... 392
1-255. L3_STCOL_SOFTEN ..... 393
1-256. L3_STCOL_TRIGEN ..... 393
1-257. L3_STCOL_REQEVT ..... 393
1-258. L3_STCOL_RSPEVT ..... 394
1-259. L3_STCOL_EVTMUX_SEL_k ..... 395
1-260. L3_STCOL_EVTMUX_SEL_i[] EVTMUX_SELi Width ..... 395
1-261. L3_STCOL_DUMP_IDENTIFIER ..... 395
1-262. L3_STCOL_DUMP_COLLECTTIME ..... 396
1-263. L3_STCOL_DUMP_SLVADDR ..... 396
1-264. L3_STCOL_DUMP_MSTADDR ..... 396
1-265. L3_STCOL_DUMP_SLVOFS ..... 397
1-266. L3_STCOL_DUMP_MANUAL ..... 397
1-267. L3_STCOL_DUMP_SEND ..... 397
1-268. L3_STCOL_FILTER_i_GLOBALEN ..... 398
1-269. L3_STCOL_FILTER_i_EN_m ..... 398
1-270. L3_STCOL_FILTER_i_MASK_m_MSTADDR ..... 399
1-271. L3_STCOL_FILTER_i_MASK_m_RD ..... 399
1-272. L3_STCOL_FILTER_i_MASK_m_WR ..... 400
1-273. L3_STCOL_FILTER_i_MASK_m_ERR ..... 400
1-274. L3_STCOL_FILTER_i_MASK_m_USERINFO ..... 401
1-275. L3_STCOL_FILTER_i_MASK_m_SLVADDR ..... 401
1-276. L3_STCOL_FILTER_i_MASK_m_REQUSERINFO ..... 401
1-277. L3_STCOL_FILTER_i_MASK_m_RSPUSERINFO ..... 402
1-278. L3_STCOL_FILTER_i_MATCH_m_MSTADDR ..... 402
1-279. L3_STCOL_FILTER_i_MATCH_m_SLVADDR ..... 402
1-280. L3_STCOL_FILTER_i_MATCH_m_RD ..... 403
1-281. L3_STCOL_FILTER_i_MATCH_m_WR ..... 404
1-282. L3_STCOL_FILTER_i_MATCH_m_ERR ..... 404
1-283. L3_STCOL_FILTER_i_MATCH_m_USERINFO ..... 405
1-284. L3_STCOL_FILTER_i_MATCH_m_REQUSERINFO ..... 405
1-285. L3_STCOL_FILTER_i_MATCH_m_RSPUSERINFO ..... 405
1-286. L3_STCOL_OP_i_THRESHOLD_MINVAL ..... 406
1-287. L3_STCOL_OP_i_THRESHOLD_MAXVAL ..... 406
1-288. L3_STCOL_OP_i_EVTINFOSEL ..... 406
1-289. L3_STCOL_OP_i_SEL ..... 407
1-290. Integration Attributes ..... 409
1-291. Clocks and Resets ..... 409
1-292. L4 HS TAs ..... 410


Copyright © 2011–2016, Texas Instruments Incorporated

1-293. L4 HS IAs ..... 410
1-294. L4_LS TAs ..... 410
1-295. L4_LS IAs ..... 411
1-296. L4 FWCFG TAs ..... 412
1-297. L4_FWCFG IAs ..... 412
1-298. L4 ConnID Definition ..... 414
1-299. L4_LS AP Registers Default Configuration ..... 416
1-300. L4_HS AP Registers Default Configuration ..... 416
1-301. L4_FWCFG AP Registers Default Configuration ..... 416
1-302. Region Allocations for L4_HS Interconnect ..... 417
1-303. Region Allocations for L4_LS Interconnect ..... 418
1-304. Region Allocations for L4_FWCFG Interconnect ..... 421
1-305. L4 Address and Protection Register Description Overview ..... 422
1-306. L4 CODE Bit Field Definition ..... 423
1-307. L4 Time-Out Link and TA Programming ..... 424
1-308. Global Initialization of Surrounding Modules ..... 426
1-309. Main Sequence – Error Analysis Mode ..... 428
1-310. Subprocess Call Summary for Main Sequence – Error Analysis Mode ..... 428
1-311. Protection Violation Error Identification ..... 429
1-312. Unsupported Command/Address Hole Error Identification ..... 429
1-313. Reset TA and Module ..... 429
1-314. Time-Out Configuration ..... 430
1-315. Firewall Configuration ..... 430
1-316. L4LS Instance Summary ..... 431
1-317. L4HS Instance Summary ..... 432
1-318. IA Registers Mapping Summary ..... 432
1-319. L4_IA_COMPONENT_L ..... 433
1-320. L4_IA_COMPONENT_H ..... 434
1-321. L4_IA_CORE_L ..... 434
1-322. L4_IA_CORE_H ..... 434
1-323. L4_IA_AGENT_CONTROL_L ..... 435
1-324. L4_IA_AGENT_CONTROL_H ..... 435
1-325. L4_IA_AGENT_STATUS_L ..... 436
1-326. L4_IA_AGENT_STATUS_H ..... 436
1-327. L4_IA_ERROR_LOG_L ..... 437
1-328. L4_IA_ERROR_LOG_H ..... 437
1-329. L4_IA_ERROR_LOG_ADDR_L ..... 438
1-330. L4_IA_ERROR_LOG_ADDR_H ..... 438
1-331. L4LS_TA Register Mapping Summary 1 ..... 438
1-332. L4LS_TA Register Mapping Summary 2 ..... 439
1-333. L4LS_TA Register Mapping Summary 3 ..... 439
1-334. L4LS_TA Register Mapping Summary 4 ..... 439
1-335. L4LS_TA Register Mapping Summary 5 ..... 440
1-336. L4LS_TA Register Mapping Summary 6 ..... 440
1-337. L4LS_TA Register Mapping Summary 7 ..... 440
1-338. L4LS_TA Register Mapping Summary 8 ..... 441
1-339. L4LS_TA Register Mapping Summary 9 ..... 441
1-340. L4LS_TA Register Mapping Summary 10 ..... 441
1-341. L4LS_TA Register Mapping Summary 11 ..... 442


Copyright © 2011–2016, Texas Instruments Incorporated

1-342. L4LS_TA Register Mapping Summary 12 ..... 442
1-343. L4LS_TA Register Mapping Summary 13 ..... 442
1-344. L4HS_TA Register Mapping Summary 1 ..... 443
1-345. L4HS_TA Register Mapping Summary 2 ..... 443
1-346. L4HS_TA Register Mapping Summary 3 ..... 443
1-347. L4HS_TA Register Mapping Summary 4 ..... 444
1-348. L4_TA_COMPONENT_H ..... 444
1-349. L4_TA_COMPONENT_L ..... 445
1-350. L4_TA_CORE_L ..... 445
1-351. L4_TA_CORE_H ..... 445
1-352. L4_TA_AGENT_CONTROL_L ..... 446
1-353. L4_TA_AGENT_CONTROL_H ..... 446
1-354. L4_TA_AGENT_STATUS_L ..... 447
1-355. L4_TA_AGENT_STATUS_H ..... 447
1-356. LA Register Mapping Summary ..... 447
1-357. L4_LA_COMPONENT_L ..... 448
1-358. L4_LA_COMPONENT_H ..... 448
1-359. L4_LA_NETWORK_L ..... 449
1-360. L4_LA_NETWORK_H ..... 449
1-361. L4_LA_INITIATOR_INFO_L ..... 449
1-362. Reset value for L4_LA_INITIATOR_INFO_L ..... 450
1-363. L4_LA_INITIATOR_INFO_H ..... 450
1-364. L4_LA_NETWORK_CONTROL_L ..... 451
1-365. L4_LA_NETWORK_CONTROL_H ..... 451
1-366. L4_LA_FLAG_MASK_i_L ..... 452
1-367. L4_LA_FLAG_MASK_i_H ..... 452
1-368. L4_LA_FLAG_STATUS_i_L ..... 452
1-369. L4_LA_FLAG_STATUS_i_H ..... 453
1-370. L4 AP Register Summary ..... 453
1-371. L4_AP_COMPONENT_L ..... 454
1-372. L4_AP_COMPONENT_H ..... 454
1-373. L4_AP_SEGMENT_i_L ..... 454
1-374. L4_AP_SEGMENT_i_H ..... 455
1-375. Reset Value for L4_AP_SEGMENT_i ..... 455
1-376. L4_AP_PROT_GROUP_MEMBERS_k_L ..... 455
1-377. L4_AP_PROT_GROUP_MEMBERS_k_H ..... 456
1-378. L4_AP_PROT_GROUP_ROLES_k_L ..... 456
1-379. L4_AP_PROT_GROUP_ROLES_k_H ..... 456
1-380. L4_AP_REGION_l_L ..... 457
1-381. L4_AP_REGION_l_H ..... 457
1-382. L4_AP_REGION_l Reset Value for L4HS ..... 458
1-383. L4_AP_REGION_l Reset Value for L4 LS ..... 458
2-1. Master Module Standby-Mode Settings ..... 465
2-2. Master Module Standby Status ..... 465
2-3. Module Idle Mode Settings ..... 465
2-4. Slave Module Idle Status ..... 466
2-5. Slave Module Mode Settings in PRCM ..... 466
2-6. Module Clock Enabling Condition ..... 467
2-7. Clock Domain Functional Clock States ..... 468


Copyright © 2011–2016, Texas Instruments Incorporated

2-8. Clock Domain States ..... 468
2-9. Clock Transition Mode Settings ..... 469
2-10. States of a Memory Area in a Power Domain ..... 470
2-11. States of a Logic Area in a Power Domain ..... 470
2-12. Power Domain Control and Status Registers ..... 470
2-13. PRCM Functional Interfaces ..... 472
2-14. PLL Names and Types ..... 476
2-15. Main PLL Clock Source ..... 477
2-16. : Main PLL Supported Configurable Dividers ..... 478
2-17. Clock Multiplexing Used With Main PLLs ..... 478
2-18. USB PLL Supported Dividers ..... 479
2-19. Clock Multiplexing used with USB PLL ..... 479
2-20. Clock Multiplexing Used With SERDES PLL ..... 481
2-21. Video PLL Supported Dividers ..... 483
2-22. Clock Multiplexing Used With Video PLLs ..... 483
2-23. Audio Frequencies ..... 484
2-24. Audio PLL Supported Dividers ..... 485
2-25. Audio PLL Clock Sources ..... 485
2-26. Timer Clock Structure ..... 488
2-27. Clkout Divider Options ..... 490
2-28. AUXOSC Oscillator Settings ..... 490
2-29. DEVOSC Oscillator Settings ..... 491
2-30. Clock Functions ..... 492
2-31. Frequency Factors ..... 493
2-32. Output Clocks in Locked Condition ..... 495
2-33. Output Clocks Before Lock and During Relock Modes ..... 495
2-34. DPLLLJ Frequency Factors ..... 496
2-35. Modes of Operations ..... 496
2-36. Reset Sources ..... 502
2-37. Volage and Power Domains ..... 513
2-38. Power Domain State Table ..... 514
2-39. Device Modules ..... 515
2-40. Supported DVFS OPPs Summary ..... 517
2-41. PRCM Memory Mapped Registers ..... 518
2-42. PLLSS REGISTERS ..... 520
2-43. CONTROL_REVISION Register Field Descriptions ..... 526
2-44. CONTROL_HWINFO Register Field Descriptions ..... 527
2-45. CONTROL_SYSCONFIG Register Field Descriptions ..... 528
2-46. PLLSS_MMR_LOCK Register Field Descriptions ..... 529
2-47. MPUPLL_PWRCTRL Register Field Descriptions ..... 530
2-48. MPUPLL_CLKCTRL Register Field Descriptions ..... 531
2-49. MPUPLL_TENABLE Register Field Descriptions ..... 533
2-50. MPUPLL_TENABLEDIV Register Field Descriptions ..... 534
2-51. MPUPLL_M2NDIV Register Field Descriptions ..... 535
2-52. MPUPLL_MN2DIV Register Field Descriptions ..... 536
2-53. MPUPLL_FRACDIV Register Field Descriptions ..... 537
2-54. MPUPLL_BWCTRL Register Field Descriptions ..... 538
2-55. MPUPLL_FRACCTRL Register Field Descriptions ..... 539
2-56. MPUPLL_STATUS Register Field Descriptions ..... 540


Copyright © 2011–2016, Texas Instruments Incorporated

2-57. MPUPLL_M3DIV Register Field Descriptions ..... 542
2-58. MPUPLL_RAMPCTRL Register Field Descriptions ..... 543
2-59. DSPPLL_PWRCTRL Register Field Descriptions ..... 544
2-60. DSPPLL_CLKCTRL Register Field Descriptions ..... 545
2-61. DSPPLL_TENABLE Register Field Descriptions ..... 547
2-62. DSPPLL_TENABLEDIV Register Field Descriptions ..... 548
2-63. DSPPLL_M2NDIV Register Field Descriptions ..... 549
2-64. DSPPLL_MN2DIV Register Field Descriptions ..... 550
2-65. DSPPLL_FRACDIV Register Field Descriptions ..... 551
2-66. DSPPLL_BWCTRL Register Field Descriptions ..... 552
2-67. DSPPLL_FRACCTRL Register Field Descriptions ..... 553
2-68. DSPPLL_STATUS Register Field Descriptions ..... 554
2-69. SGXPLLL_PWRCTRL Register Field Descriptions ..... 556
2-70. SGXPLL_CLKCTRL Register Field Descriptions ..... 557
2-71. SGXPLLL_TENABLE Register Field Descriptions ..... 559
2-72. SGXPLL_TENABLEDIV Register Field Descriptions ..... 560
2-73. SGXPLL_M2NDIV Register Field Descriptions ..... 561
2-74. SGXPLL_MN2DIV Register Field Descriptions ..... 562
2-75. SGXPLL_FRACDIV Register Field Descriptions ..... 563
2-76. SGXPLL_BWCTRL Register Field Descriptions ..... 564
2-77. SGXPLL_FRACCTRL Register Field Descriptions ..... 565
2-78. SGXPLL_STATUS Register Field Descriptions ..... 566
2-79. HDVICPPLL_PWRCTRL Register Field Descriptions ..... 568
2-80. HDVICPPLL_CLKCTRL Register Field Descriptions ..... 569
2-81. HDVICPPLL_TENABLE Register Field Descriptions ..... 571
2-82. HDVICPPLL_TENABLEDIV Register Field Descriptions ..... 572
2-83. HDVICPPLL_M2NDIV Register Field Descriptions ..... 573
2-84. HDVICPPLL_MN2DIV Register Field Descriptions ..... 574
2-85. HDVICPPLL_FRACDIV Register Field Descriptions ..... 575
2-86. HDVICPPLL_BWCTRL Register Field Descriptions ..... 576
2-87. HDVICPPLL_FRACCTRL Register Field Descriptions ..... 577
2-88. HDVICPPLL_STATUS Register Field Descriptions ..... 578
2-89. L3PLL_PWRCTRL Register Field Descriptions ..... 580
2-90. L3PLL_CLKCTRL Register Field Descriptions ..... 581
2-91. L3PLL_TENABLE Register Field Descriptions ..... 583
2-92. L3PLL_TENABLEDIV Register Field Descriptions ..... 584
2-93. L3PLL_M2NDIV Register Field Descriptions ..... 585
2-94. L3PLL_MN2DIV Register Field Descriptions ..... 586
2-95. L3PLL_FRACDIV Register Field Descriptions ..... 587
2-96. L3PLL_BWCTRL Register Field Descriptions ..... 588
2-97. L3PLL_FRACCTRL Register Field Descriptions ..... 589
2-98. L3PLL_STATUS Register Field Descriptions ..... 590
2-99. ISPPLL_PWRCTRL Register Field Descriptions ..... 592
2-100. ISPPLL_CLKCTRL Register Field Descriptions ..... 593
2-101. ISPPLL_TENABLE Register Field Descriptions ..... 595
2-102. ISPPLL_TENABLEDIV Register Field Descriptions ..... 596
2-103. ISPPLL_M2NDIV Register Field Descriptions ..... 597
2-104. ISPPLL_MN2DIV Register Field Descriptions ..... 598
2-105. ISPPLL_FRACDIV Register Field Descriptions ..... 599


Copyright © 2011–2016, Texas Instruments Incorporated

2-106. ISPPLL_BWCTRL Register Field Descriptions ..... 600
2-107. ISPPLL_FRACCTRL Register Field Descriptions ..... 601
2-108. ISPPLL_STATUS Register Field Descriptions ..... 602
2-109. DSSPLL_PWRCTRL Register Field Descriptions ..... 604
2-110. DSSPLL_CLKCTRL Register Field Descriptions ..... 605
2-111. DSSPLL_TENABLE Register Field Descriptions ..... 607
2-112. DSSPLL_TENABLEDIV Register Field Descriptions ..... 608
2-113. DSSPLL_M2NDIV Register Field Descriptions ..... 609
2-114. DSSPLL_MN2DIV Register Field Descriptions ..... 610
2-115. DSSPLL_FRACDIV Register Field Descriptions ..... 611
2-116. DSSPLL_BWCTRL Register Field Descriptions ..... 612
2-117. DSSPLL_FRACCTRL Register Field Descriptions ..... 613
2-118. DSSPLL_STATUS Register Field Descriptions ..... 614
2-119. VIDEO0PLL_PWRCTRL Register Field Descriptions ..... 616
2-120. VIDEO0PLL_CLKCTRL Register Field Descriptions ..... 617
2-121. VIDEO0PLL_TENABLE Register Field Descriptions ..... 619
2-122. VIDEO0PLL_TENABLEDIV Register Field Descriptions ..... 620
2-123. VIDEO0PLL_M2NDIV Register Field Descriptions ..... 621
2-124. VIDEO0PLL_MN2DIV Register Field Descriptions ..... 622
2-125. VIDEO0PLL_FRACDIV Register Field Descriptions ..... 623
2-126. VIDEO0PLL_BWCTRL Register Field Descriptions ..... 624
2-127. VIDEO0PLL_FRACCTRL Register Field Descriptions ..... 625
2-128. VIDEO0PLL_STATUS Register Field Descriptions ..... 626
2-129. VIDEO1PLL_PWRCTRL Register Field Descriptions ..... 628
2-130. VIDEO1PLL_CLKCTRL Register Field Descriptions ..... 629
2-131. VIDEO1PLL_TENABLE Register Field Descriptions ..... 631
2-132. VIDEO1PLL_TENABLEDIV Register Field Descriptions ..... 632
2-133. VIDEO1PLL_M2NDIV Register Field Descriptions ..... 633
2-134. VIDEO1PLL_MN2DIV Register Field Descriptions ..... 634
2-135. VIDEO1PLL_FRACDIV Register Field Descriptions ..... 635
2-136. VIDEO1PLL_BWCTRL Register Field Descriptions ..... 636
2-137. VIDEO1PLL_FRACCTRL Register Field Descriptions ..... 637
2-138. VIDEO1PLL_STATUS Register Field Descriptions ..... 638
2-139. HDMIPLL_PWRCTRL Register Field Descriptions ..... 640
2-140. HDMIPLL_CLKCTRL Register Field Descriptions ..... 641
2-141. HDMIPLL_TENABLE Register Field Descriptions ..... 643
2-142. HDMIPLL_TENABLEDIV Register Field Descriptions ..... 644
2-143. HDMIPLL_M2NDIV Register Field Descriptions ..... 645
2-144. HDMIPLL_MN2DIV Register Field Descriptions ..... 646
2-145. HDMIPLL_FRACDIV Register Field Descriptions ..... 647
2-146. HDMIPLL_BWCTRL Register Field Descriptions ..... 648
2-147. HDMIPLL_FRACCTRL Register Field Descriptions ..... 649
2-148. HDMIPLL_STATUS Register Field Descriptions ..... 650
2-149. AUDIOPLL_PWRCTRL Register Field Descriptions ..... 652
2-150. AUDIOPLL_CLKCTRL Register Field Descriptions ..... 653
2-151. AUDIOPLL_TENABLE Register Field Descriptions ..... 655
2-152. AUDIOPLL_TENABLEDIV Register Field Descriptions ..... 656
2-153. AUDIOPLL_M2NDIV Register Field Descriptions ..... 657
2-154. AUDIOPLL_MN2DIV Register Field Descriptions ..... 658


Copyright © 2011–2016, Texas Instruments Incorporated

2-155. AUDIOPLL_FRACDIV Register Field Descriptions ..... 659
2-156. AUDIOPLL_BWCTRL Register Field Descriptions ..... 660
2-157. AUDIOPLL_FRACCTRL Register Field Descriptions ..... 661
2-158. AUDIOPLL_STATUS Register Field Descriptions ..... 662
2-159. USBPLL_PWRCTRL Register Field Descriptions ..... 664
2-160. USBPLL_CLKCTRL Register Field Descriptions ..... 665
2-161. USBPLL_TENABLE Register Field Descriptions ..... 667
2-162. USBPLL_TENABLEDIV Register Field Descriptions ..... 668
2-163. USBPLL_M2NDIV Register Field Descriptions ..... 669
2-164. USBPLL_MN2DIV Register Field Descriptions ..... 670
2-165. USBPLL_FRACDIV Register Field Descriptions ..... 671
2-166. USBPLL_BWCTRL Register Field Descriptions ..... 672
2-167. USBPLL_FRACCTRL Register Field Descriptions ..... 673
2-168. USBPLL_STATUS Register Field Descriptions ..... 674
2-169. DDRPLL_PWRCTRL Register Field Descriptions ..... 676
2-170. DDRPLL_CLKCTRL Register Field Descriptions ..... 677
2-171. DDRPLL_TENABLE Register Field Descriptions ..... 679
2-172. DDRPLL_TENABLEDIV Register Field Descriptions ..... 680
2-173. DDRPLL_M2NDIV Register Field Descriptions ..... 681
2-174. DDRPLL_MN2DIV Register Field Descriptions ..... 682
2-175. DDRPLL_FRACDIV Register Field Descriptions ..... 683
2-176. DDRPLL_BWCTRL Register Field Descriptions ..... 684
2-177. DDRPLL_FRACCTRL Register Field Descriptions ..... 685
2-178. DDRPLL_STATUS Register Field Descriptions ..... 686
2-179. OSC_SRC Register Field Descriptions ..... 688
2-180. MPU_CLKSRC Register Field Descriptions ..... 690
2-181. VIDEO_PLL_CLKSRC Register Field Descriptions ..... 691
2-182. MLB_ATL_CLKSRC Register Field Descriptions ..... 692
2-183. McASP345_AUX_CLKSRC Register Field Descriptions ..... 693
2-184. McASP_AHCLK_CLKSRC Register Field Descriptions ..... 694
2-185. McBSP_UART_CLKSRC Register Field Descriptions ..... 696
2-186. HDMI_I2S_CLKSRC Register Field Descriptions ..... 697
2-187. DMTIMER_CLKSRC Register Field Descriptions ..... 698
2-188. CLKOUT_MUX Register Field Descriptions ..... 700
2-189. RMII_REFCLK_SRC Register Field Descriptions ..... 701
2-190. SYSCLK18_CLKSRC Register Field Descriptions ..... 702
2-191. WDT0_CLKSRC Register Field Descriptions ..... 703
2-192. DMTIMER_CLK_CHANGE Register Field Descriptions ..... 704
2-193. DEEPSLEEP_CTRL Register Field Descriptions ..... 706
2-194. DEEPSLEEP_STATUS Register Field Descriptions ..... 707
2-195. PRM_DEVICE REGISTERS ..... 707
2-196. PRM_RSTCTRL Register Field Descriptions ..... 708
2-197. PRM_RSTTIME Register Field Descriptions ..... 709
2-198. PRM_RSTST Register Field Descriptions ..... 710
2-199. CM_DEVICE REGISTERS ..... 711
2-200. CM_CLKOUT_CTRL Register Field Descriptions ..... 712
2-201. OCP_SOCKET_PRM REGISTERS ..... 712
2-202. REVISION_PRM Register Field Descriptions ..... 713
2-203. CM_DPLL REGISTERS ..... 713


Copyright © 2011–2016, Texas Instruments Incorporated

2-204. CM_SYSCLK3_CLKSEL Register Field Descriptions ..... 715
2-205. CM_SYSCLK10_CLKSEL Register Field Descriptions ..... 716
2-206. CM_VPB3_CLKSEL Register Field Descriptions ..... 717
2-207. CM_VPC1_CLKSEL Register Field Descriptions ..... 718
2-208. CM_VPD1_CLKSEL Register Field Descriptions ..... 719
2-209. CM_SYSCLK19_CLKSEL Register Field Descriptions ..... 720
2-210. CM_SYSCLK20_CLKSEL Register Field Descriptions ..... 721
2-211. CM_SYSCLK21_CLKSEL Register Field Descriptions ..... 722
2-212. CM_SYSCLK22_CLKSEL Register Field Descriptions ..... 723
2-213. CM_APA_CLKSEL Register Field Descriptions ..... 724
2-214. CM_SYSCLK14_CLKSEL Register Field Descriptions ..... 725
2-215. CM_SYSCLK16_CLKSEL Register Field Descriptions ..... 726
2-216. CM_SYSCLK18_CLKSEL Register Field Descriptions ..... 727
2-217. CM_AUDIOCLK_MCASP0_CLKSEL Register Field Descriptions ..... 728
2-218. CM_AUDIOCLK_MCASP1_CLKSEL Register Field Descriptions ..... 729
2-219. CM_AUDIOCLK_MCASP2_CLKSEL Register Field Descriptions ..... 730
2-220. CM_AUDIOCLK_MCBSP_CLKSEL Register Field Descriptions ..... 731
2-221. CM_HDMI_CLKSEL Register Field Descriptions ..... 732
2-222. CM_SYSCLK23_CLKSEL Register Field Descriptions ..... 733
2-223. CM_ACTIVE REGISTERS ..... 733
2-224. CM_GEM_CLKSTCTRL Register Field Descriptions ..... 734
2-225. CM_ACTIVE_GEM_CLKCTRL Register Field Descriptions ..... 735
2-226. CM_DEFAULT REGISTERS ..... 736
2-227. CM_DEFAULT_PCI_CLKSTCTRL Register Field Descriptions ..... 737
2-228. CM_DEFAULT_DUCATI_CLKSTCTRL Register Field Descriptions ..... 738
2-229. CM_DEFAULT_DMM_CLKCTRL Register Field Descriptions ..... 739
2-230. CM_DEFAULT_USB_CLKCTRL Register Field Descriptions ..... 740
2-231. CM_DEFAULT_SATA_CLKCTRL Register Field Descriptions ..... 741
2-232. CM_DEFAULT_DUCATI_CLKCTRL Register Field Descriptions ..... 742
2-233. CM_DEFAULT_PCI_CLKCTRL Register Field Descriptions ..... 743
2-234. CM_HDVICP REGISTERS ..... 744
2-235. CM_HDVICP_CLKSTCTRL Register Field Descriptions ..... 745
2-236. CM_HDVICP_CLKCTRL Register Field Descriptions ..... 746
2-237. CM_HDVICP_SL2_CLKCTRL Register Field Descriptions ..... 747
2-238. CM_ISP REGISTERS ..... 747
2-239. CM_ISP_CLKSTCTRL Register Field Descriptions ..... 749
2-240. CM_ISP_ISP_CLKCTRL Register Field Descriptions ..... 750
2-241. CM_ISP_FDIF_CLKCTRL Register Field Descriptions ..... 751
2-242. CM_DSS REGISTERS ..... 751
2-243. CM_DSS_CLKSTCTRL Register Field Descriptions ..... 752
2-244. CM_DSS_DSS_CLKCTRL Register Field Descriptions ..... 753
2-245. CM_DSS_HDMI_CLKCTRL Register Field Descriptions ..... 754
2-246. CM_SGX REGISTERS ..... 754
2-247. CM_SGX_CLKSTCTRL Register Field Descriptions ..... 755
2-248. CM_SGX_SGX_CLKCTRL Register Field Descriptions ..... 756
2-249. PRM_ACTIVE REGISTERS ..... 757
2-250. PM_ACTIVE_PWRSTCTRL Register Field Descriptions ..... 758
2-251. PM_ACTIVE_PWRSTST Register Field Descriptions ..... 759
2-252. RM_ACTIVE_RSTCTRL Register Field Descriptions ..... 760


Copyright © 2011–2016, Texas Instruments Incorporated

2-253. RM_ACTIVE_RSTST Register Field Descriptions ..... 761
2-254. PRM_DEFAULT REGISTERS ..... 761
2-255. PM_DEFAULT_PWRSTCTRL Register Field Descriptions ..... 763
2-256. RM_DEFAULT_RSTCTRL Register Field Descriptions ..... 764
2-257. RM_DEFAULT_RSTST Register Field Descriptions ..... 765
2-258. PRM_HDVICP REGISTERS ..... 766
2-259. PM_HDVICP_PWRSTCTRL Register Field Descriptions ..... 767
2-260. PM_HDVICP_PWRSTST Register Field Descriptions ..... 768
2-261. RM_HDVICP_RSTCTRL Register Field Descriptions ..... 769
2-262. RM_HDVICP_RSTST Register Field Descriptions ..... 770
2-263. PRM_ISP REGISTERS ..... 771
2-264. PM_ISP_PWRSTCTRL Register Field Descriptions ..... 772
2-265. PM_ISP_PWRSTST Register Field Descriptions ..... 773
2-266. RM_ISP_RSTCTRL Register Field Descriptions ..... 774
2-267. RM_ISP_RSTST Register Field Descriptions ..... 775
2-268. PRM_DSS REGISTERS ..... 775
2-269. PM_DSS_PWRSTCTRL Register Field Descriptions ..... 777
2-270. PM_DSS_PWRSTST Register Field Descriptions ..... 778
2-271. RM_DSS_RSTCTRL Register Field Descriptions ..... 779
2-272. RM_DSS_RSTST Register Field Descriptions ..... 780
2-273. PRM_SGX REGISTERS ..... 780
2-274. PM_SGX_PWRSTCTRL Register Field Descriptions ..... 782
2-275. RM_SGX_RSTCTRL Register Field Descriptions ..... 783
2-276. PM_SGX_PWRSTST Register Field Descriptions ..... 784
2-277. RM_SGX_RSTST Register Field Descriptions ..... 785
2-278. CM_ALWON REGISTERS ..... 785
2-279. CM_ALWON_L3_SLOW_CLKSTCTRL Register Field Descriptions ..... 788
2-280. CM_ETHERNET_CLKSTCTRL Register Field Descriptions ..... 791
2-281. CM_ALWON_L3_MED_CLKSTCTRL Register Field Descriptions ..... 792
2-282. CM_MMU_CLKSTCTRL Register Field Descriptions ..... 793
2-283. CM_MMUCFG_CLKSTCTRL Register Field Descriptions ..... 794
2-284. CM_ALWON_OCMC_0_CLKSTCTRL Register Field Descriptions ..... 795
2-285. CM_ALWON_VCP_CLKSTCTRL Register Field Descriptions ..... 796
2-286. CM_ALWON_MPU_CLKSTCTRL Register Field Descriptions ..... 797
2-287. CM_ALWON_SYSCLK4_CLKSTCTRL Register Field Descriptions ..... 798
2-288. CM_ALWON_SYSCLK5_CLKSTCTRL Register Field Descriptions ..... 799
2-289. CM_ALWON_SYSCLK6_CLKSTCTRL Register Field Descriptions ..... 800
2-290. CM_ALWON_RTC_CLKSTCTRL Register Field Descriptions ..... 801
2-291. CM_ALWON_L3_FAST_CLKSTCTRL Register Field Descriptions ..... 802
2-292. CM_ALWON_MCASP0_CLKCTRL Register Field Descriptions ..... 803
2-293. CM_ALWON_MCASP1_CLKCTRL Register Field Descriptions ..... 804
2-294. CM_ALWON_MCASP2_CLKCTRL Register Field Descriptions ..... 805
2-295. CM_ALWON_MCBSP_CLKCTRL Register Field Descriptions ..... 806
2-296. CM_ALWON_UART_0_CLKCTRL Register Field Descriptions ..... 807
2-297. CM_ALWON_UART_1_CLKCTRL Register Field Descriptions ..... 808
2-298. CM_ALWON_UART_2_CLKCTRL Register Field Descriptions ..... 809
2-299. CM_ALWON_GPIO_0_CLKCTRL Register Field Descriptions ..... 810
2-300. CM_ALWON_GPIO_1_CLKCTRL Register Field Descriptions ..... 811
2-301. CM_ALWON_I2C_0_CLKCTRL Register Field Descriptions ..... 812


Copyright © 2011–2016, Texas Instruments Incorporated

2-302. CM_ALWON_I2C_1_CLKCTRL Register Field Descriptions ..... 813
2-303. CM_ALWON_MCASP_3_4_5_CLKCTRL Register Field Descriptions ..... 814
2-304. CM_ALWON_ATL_CLKCTRL Register Field Descriptions ..... 815
2-305. CM_ALWON_MLB_CLKCTRL Register Field Descriptions ..... 816
2-306. CM_ALWON_PATA_CLKCTRL Register Field Descriptions ..... 817
2-307. CM_ALWON_UART_3_CLKCTRL Register Field Descriptions ..... 818
2-308. CM_ALWON_UART_4_CLKCTRL Register Field Descriptions ..... 819
2-309. CM_ALWON_UART_5_CLKCTRL Register Field Descriptions ..... 820
2-310. CM_ALWON_WDTIMER_CLKCTRL Register Field Descriptions ..... 821
2-311. CM_ALWON_SPI_CLKCTRL Register Field Descriptions ..... 822
2-312. CM_ALWON_MAILBOX_CLKCTRL Register Field Descriptions ..... 823
2-313. CM_ALWON_SPINBOX_CLKCTRL Register Field Descriptions ..... 824
2-314. CM_ALWON_MMUDATA_CLKCTRL Register Field Descriptions ..... 825
2-315. CM_ALWON_MMUCFG_CLKCTRL Register Field Descriptions ..... 826
2-316. CM_ALWON_OCMC_0_CLKCTRL Register Field Descriptions ..... 827
2-317. CM_ALWON_VCP_CLKCTRL Register Field Descriptions ..... 828
2-318. CM_ALWON_CONTROL_CLKCTRL Register Field Descriptions ..... 829
2-319. CM_ALWON_GPMC_CLKCTRL Register Field Descriptions ..... 830
2-320. CM_ALWON_ETHERNET_0_CLKCTRL Register Field Descriptions ..... 831
2-321. CM_ALWON_MPU_CLKCTRL Register Field Descriptions ..... 832
2-322. CM_ALWON_DEBUGSS_CLKCTRL Register Field Descriptions ..... 833
2-323. CM_ALWON_L3_CLKCTRL Register Field Descriptions ..... 835
2-324. CM_ALWON_L4HS_CLKCTRL Register Field Descriptions ..... 836
2-325. CM_ALWON_L4LS_CLKCTRL Register Field Descriptions ..... 837
2-326. CM_ALWON_RTC_CLKCTRL Register Field Descriptions ..... 838
2-327. CM_ALWON_TPCC_CLKCTRL Register Field Descriptions ..... 839
2-328. CM_ALWON_TPTC0_CLKCTRL Register Field Descriptions ..... 840
2-329. CM_ALWON_TPTC1_CLKCTRL Register Field Descriptions ..... 841
2-330. CM_ALWON_TPTC2_CLKCTRL Register Field Descriptions ..... 842
2-331. CM_ALWON_TPTC3_CLKCTRL Register Field Descriptions ..... 843
2-332. CM_ALWON_DCAN_0_1_CLKCTRL Register Field Descriptions ..... 844
2-333. CM_ALWON_MMCHS_0_CLKCTRL Register Field Descriptions ..... 845
2-334. CM_ALWON_MMCHS_1_CLKCTRL Register Field Descriptions ..... 846
2-335. CM_ALWON_MMCHS_2_CLKCTRL Register Field Descriptions ..... 847
2-336. CM_ALWON_CUST_EFUSE_CLKCTRL Register Field Descriptions ..... 848
2-337. PRM_ALWON REGISTERS ..... 849
2-338. RM_ALWON_RSTST Register Field Descriptions ..... 850
3-1. MMR_LOCKx Registers ..... 852
3-2. List of PINCNTLx Registers, reset affected by RESET_ISO. ..... 855
3-3. CONTROL_MODULE REGISTERS ..... 856
3-4. CONTROL_REVISION Register Field Descriptions ..... 862
3-5. CONTROL_HWINFO Register Field Descriptions ..... 863
3-6. CONTROL_SYSCONFIG Register Field Descriptions ..... 864
3-7. CONTROL_STATUS Register Field Descriptions ..... 865
3-8. BOOTSTAT Register Field Descriptions ..... 867
3-9. DSPBOOTADDR Register Field Descriptions ..... 868
3-10. MMR_LOCK0 Register Field Descriptions ..... 869
3-11. MMR_LOCK1 Register Field Descriptions ..... 870
3-12. MMR_LOCK2 Register Field Descriptions ..... 871


Copyright © 2011–2016, Texas Instruments Incorporated

3-13. MMR_LOCK3 Register Field Descriptions ..... 872
3-14. MMR_LOCK4 Register Field Descriptions ..... 873
3-15. CONTROL_SEC_CTRL Register Field Descriptions ..... 874
3-16. DEVOSC Register Field Descriptions ..... 875
3-17. AUXOSC Register Field Descriptions ..... 876
3-18. PCIE_CFG Register Field Descriptions ..... 877
3-19. DEVICE_ID Register Field Descriptions ..... 878
3-20. DEV_FEATURE Register Field Descriptions ..... 879
3-21. INIT_PRIORITY_0 Register Field Descriptions ..... 881
3-22. INIT_PRIORITY_1 Register Field Descriptions ..... 882
3-23. MMU_CFG Register Field Descriptions ..... 883
3-24. TPTC_CFG Register Field Descriptions ..... 884
3-25. DSP_IDLE_CFG Register Field Descriptions ..... 885
3-26. USB_CTRL0 Register Field Descriptions ..... 886
3-27. USB_STS0 Register Field Descriptions ..... 888
3-28. USB_CTRL1 Register Field Descriptions ..... 889
3-29. USB_STS1 Register Field Descriptions ..... 891
3-30. MAC_ID0_LO Register Field Descriptions ..... 892
3-31. MAC_ID0_HI Register Field Descriptions ..... 893
3-32. MAC_ID1_LO Register Field Descriptions ..... 894
3-33. MAC_ID1_HI Register Field Descriptions ..... 895
3-34. SW_REVISION Register Field Descriptions ..... 896
3-35. DCAN_RAMINIT Register Field Descriptions ..... 897
3-36. AUD_CTRL Register Field Descriptions ..... 898
3-37. GMII_SEL Register Field Descriptions ..... 899
3-38. OCMEM_PWRDN Register Field Descriptions ..... 900
3-39. MEDIA_CONTROLLER_MEM_PWRDN Register Field Descriptions ..... 901
3-40. SD_DAC_CTRL Register Field Descriptions ..... 902
3-41. SD_DAC0_CAL Register Field Descriptions ..... 904
3-42. SD_DAC1_CAL Register Field Descriptions ..... 905
3-43. SD_DAC0_REGCTRL Register Field Descriptions ..... 906
3-44. SD_DAC0_REGSTATUS Register Field Descriptions ..... 907
3-45. SD_DAC1_REGCTRL Register Field Descriptions ..... 908
3-46. SD_DAC1_REGSTATUS Register Field Descriptions ..... 909
3-47. EMIF_CLK_GATE Register Field Descriptions ..... 910
3-48. SMRT_CTRL Register Field Descriptions ..... 911
3-49. ARM_CORTEX_A8_HW_DBG_SEL Register Field Descriptions ..... 912
3-50. ARM_CORTEX_A8_HW_DBG_INFO Register Field Descriptions ..... 913
3-51. PRCM_DEBUG_ALWON_DEFAULT Register Field Descriptions ..... 914
3-52. PRCM_DEBUG_PD_DOMAIN_STATUS Register Field Descriptions ..... 916
3-53. PCIE_PLLCFG0 Register Field Descriptions ..... 919
3-54. PCIE_PLLCFG1 Register Field Descriptions ..... 921
3-55. PCIE_PLLCFG2 Register Field Descriptions ..... 922
3-56. PCIE_PLLCFG3 Register Field Descriptions ..... 923
3-57. PCIE_PLLCFG4 Register Field Descriptions ..... 925
3-58. PCIE_PLLSTATUS Register Field Descriptions ..... 927
3-59. PCIE_RXSTATUS Register Field Descriptions ..... 928
3-60. PCIE_TXSTATUS Register Field Descriptions ..... 929
3-61. SATA_PLLCFG0 Register Field Descriptions ..... 930


Copyright © 2011–2016, Texas Instruments Incorporated

3-62. SATA_PLLCFG1 Register Field Descriptions ..... 931
3-63. SATA_PLLCFG2 Register Field Descriptions ..... 932
3-64. SATA_PLLCFG3 Register Field Descriptions ..... 933
3-65. SATA_PLLCFG4 Register Field Descriptions ..... 936
3-66. SATA_TXSTATUS Register Field Descriptions ..... 938
3-67. SATA_TESTCFG Register Field Descriptions ..... 939
3-68. VDD_MPU_OPP_050 Register Field Descriptions ..... 940
3-69. VDD_MPU_OPP_100 Register Field Descriptions ..... 941
3-70. VDD_MPU_OPP_120 Register Field Descriptions ..... 942
3-71. VDD_MPU_OPP_166 Register Field Descriptions ..... 943
3-72. VDD_C674x_DSP_OPP_050 Register Field Descriptions ..... 944
3-73. VDD_C674x_DSP_OPP_100 Register Field Descriptions ..... 945
3-74. VDD_C674x_DSP_OPP_120 Register Field Descriptions ..... 946
3-75. VDD_C674x_DSP_OPP_166 Register Field Descriptions ..... 947
3-76. VDD_HDVICP_OPP_050 Register Field Descriptions ..... 948
3-77. VDD_HDVICP_OPP_100 Register Field Descriptions ..... 949
3-78. VDD_HDVICP_OPP_120 Register Field Descriptions ..... 950
3-79. VDD_HDVICP_OPP_166 Register Field Descriptions ..... 951
3-80. VDD_CORE_OPP_050 Register Field Descriptions ..... 952
3-81. VDD_CORE_OPP_100 Register Field Descriptions ..... 953
3-82. VDD_CORE_OPP_120 Register Field Descriptions ..... 954
3-83. VDD_CORE_OPP_166 Register Field Descriptions ..... 955
3-84. BB_SCALE Register Field Descriptions ..... 956
3-85. USB_VID_PID Register Field Descriptions ..... 957
3-86. PCIE_VID_PID Register Field Descriptions ..... 958
3-87. PINCNTL[270:1] Register Field Descriptions ..... 959
3-88. MLBP_SIG_IO_CTRL Register Field Descriptions ..... 960
3-89. MLBP_DAT_IO_CTRL Register Field Descriptions ..... 961
3-90. SERDES_REFCLK_CTL Register Field Descriptions ..... 962
3-91. DSP_INTMUX_15_18 Register Field Descriptions ..... 963
3-92. DSP_INTMUX_19_22 Register Field Descriptions ..... 964
3-93. DSP_INTMUX_23_26 Register Field Descriptions ..... 965
3-94. DSP_INTMUX_27_30 Register Field Descriptions ..... 966
3-95. DSP_INTMUX_31_34 Register Field Descriptions ..... 967
3-96. DSP_INTMUX_35_38 Register Field Descriptions ..... 968
3-97. DSP_INTMUX_39_42 Register Field Descriptions ..... 969
3-98. DSP_INTMUX_43_46 Register Field Descriptions ..... 970
3-99. DSP_INTMUX_47_50 Register Field Descriptions ..... 971
3-100. DSP_INTMUX_51_54 Register Field Descriptions ..... 972
3-101. DSP_INTMUX_55_58 Register Field Descriptions ..... 973
3-102. DSP_INTMUX_59_62 Register Field Descriptions ..... 974
3-103. DSP_INTMUX_63_66 Register Field Descriptions ..... 975
3-104. DSP_INTMUX_67_70 Register Field Descriptions ..... 976
3-105. DSP_INTMUX_71_74 Register Field Descriptions ..... 977
3-106. DSP_INTMUX_75_78 Register Field Descriptions ..... 978
3-107. DSP_INTMUX_79_82 Register Field Descriptions ..... 979
3-108. DSP_INTMUX_83_86 Register Field Descriptions ..... 980
3-109. DSP_INTMUX_87_90 Register Field Descriptions ..... 981
3-110. DSP_INTMUX_91_94 Register Field Descriptions ..... 982


Copyright © 2011–2016, Texas Instruments Incorporated

3-111. DSP_INTMUX_95 Register Field Descriptions ..... 983
3-112. Media_Controller_INTMUX_0_3 Register Field Descriptions ..... 984
3-113. Media_Controller_INTMUX_4_7 Register Field Descriptions ..... 985
3-114. Media_Controller_INTMUX_8_11 Register Field Descriptions ..... 986
3-115. Media_Controller_INTMUX_12_15 Register Field Descriptions ..... 987
3-116. Media_Controller_INTMUX_16_19 Register Field Descriptions ..... 988
3-117. Media_Controller_INTMUX_20_23 Register Field Descriptions ..... 989
3-118. Media_Controller_INTMUX_24_27 Register Field Descriptions ..... 990
3-119. Media_Controller_INTMUX_28_31 Register Field Descriptions ..... 991
3-120. Media_Controller_INTMUX_32_35 Register Field Descriptions ..... 992
3-121. Media_Controller_INTMUX_36_39 Register Field Descriptions ..... 993
3-122. Media_Controller_INTMUX_40_43 Register Field Descriptions ..... 994
3-123. Media_Controller_INTMUX_44_47 Register Field Descriptions ..... 995
3-124. Media_Controller_INTMUX_48_51 Register Field Descriptions ..... 996
3-125. Media_Controller_INTMUX_52_55 Register Field Descriptions ..... 997
3-126. Media_Controller_INTMUX_56 Register Field Descriptions ..... 998
3-127. EDMA3CC_EVTMUX_0_3 Register Field Descriptions ..... 999
3-128. EDMA3CC_EVTMUX_4_7 Register Field Descriptions ..... 1000
3-129. EDMA3CC_EVTMUX_8_11 Register Field Descriptions ..... 1001
3-130. EDMA3CC_EVTMUX_12_15 Register Field Descriptions ..... 1002
3-131. EDMA3CC_EVTMUX_16_19 Register Field Descriptions ..... 1003
3-132. EDMA3CC_EVTMUX_20_23 Register Field Descriptions ..... 1004
3-133. EDMA3CC_EVTMUX_24_27 Register Field Descriptions ..... 1005
3-134. EDMA3CC_EVTMUX_28_31 Register Field Descriptions ..... 1006
3-135. EDMA3CC_EVTMUX_32_35 Register Field Descriptions ..... 1007
3-136. EDMA3CC_EVTMUX_36_39 Register Field Descriptions ..... 1008
3-137. EDMA3CC_EVTMUX_40_43 Register Field Descriptions ..... 1009
3-138. EDMA3CC_EVTMUX_44_47 Register Field Descriptions ..... 1010
3-139. EDMA3CC_EVTMUX_48_51 Register Field Descriptions ..... 1011
3-140. EDMA3CC_EVTMUX_52_55 Register Field Descriptions ..... 1012
3-141. EDMA3CC_EVTMUX_56_59 Register Field Descriptions ..... 1013
3-142. EDMA3CC_EVTMUX_60_63 Register Field Descriptions ..... 1014
3-143. GPIO_MUX Register Field Descriptions ..... 1015
3-144. RESET_ISO Register Field Descriptions ..... 1016
4-1. Acronyms and Abbreviations ..... 1018
4-2. Naming Conventions ..... 1019
4-3. ROM Exception Vectors ..... 1022
4-4. Dead Loops ..... 1022
4-5. RAM Exception Vectors ..... 1024
4-6. Tracing Data ..... 1025
4-7. ROM Code Default Clock Settings ..... 1027
4-8. BTMode[4-0] Configuration Pins ..... 1029
4-9. XIP Timings Parameters ..... 1033
4-10. Pins Used for NOR Boot ..... 1033
4-11. Special SYSBOOT Pins for NOR Boot ..... 1035
4-12. NAND Timings Parameters ..... 1037
4-13. ONFI Parameters Page Description ..... 1037
4-14. Supported NAND Devices ..... 1037
4-15. 4th NAND ID Data Byte ..... 1039


Copyright © 2011–2016, Texas Instruments Incorporated

4-16. Pins used for NANDI2C boot for I2C EEPROM access ..... 1039
4-17. NAND Geometry Information on I2C EEPROM ..... 1039
4-18. Pins Used for NAND Boot ..... 1045
4-19. Special SYSBOOT pins for NAND boot ..... 1046
4-20. Master Boot Record Structure ..... 1051
4-21. Partition Entry ..... 1051
4-22. Partition Types ..... 1051
4-23. FAT Boot Sector ..... 1053
4-24. FAT Directory Entry ..... 1057
4-25. FAT Entry Description ..... 1058
4-26. Pins Used for MMC Boot ..... 1058
4-27. Pins Used for SPI Boot ..... 1059
4-28. Blocks and Sectors Searched on non-XIP Memories ..... 1059
4-29. Pins Used for EMAC Boot ..... 1061
4-30. Pins Used for Boot in RGMII Mode ..... 1062
4-31. Pins Used for Boot in RMII Mode ..... 1062
4-32. Sysboot Pins ..... 1062
4-33. PCIe 32 BAR Window Size Configuration ..... 1063
4-34. PCIe 64 BAR Window Size Configuration ..... 1063
4-35. PCIe BAR Window Base Address and Offset Configuration ..... 1064
4-36. Pins Used for UART Boot ..... 1065
4-37. Image Formats ..... 1066
4-38. TOC Item Fields ..... 1066
4-39. Filenames in TOC ..... 1067
4-40. Booting Parameters Structure ..... 1067
4-41. Write Auxiliary Control Register (WRITE_ACR) ..... 1068
4-42. Write Nonsecure Access Control Register (WRITE_NACR) ..... 1069
4-43. Write L2 Cache Auxiliary Control Register (WRITE_CACR) ..... 1069
4-44. Tracing Vectors ..... 1069
5-1. Initialization of a Transmit Object ..... 1088
5-2. Initialization of a single Receive Object for Data Frames ..... 1088
5-3. Initialization of a Single Receive Object for Remote Frames ..... 1089
5-4. Parameters of the CAN Bit Time ..... 1096
5-5. Structure of a Message Object ..... 1106
5-6. Field Descriptions ..... 1106
5-7. Message RAM addressing in Debug/Suspend and RDA Mode ..... 1108
5-8. Message RAM Representation in Debug/Suspend Mode ..... 1109
5-9. Message RAM Representation in RAM Direct Access Mode ..... 1110
5-10. DCAN Control Register Summary Table ..... 1111
5-11. CAN Control Register (DCAN CTL) Field Descriptions ..... 1112
5-12. Error and Status Register (DCAN ES) Field Descriptions ..... 1114
5-13. Parity Error End of Interrupt Register (PARITYERR_EOI) Field Descriptions ..... 1116
5-14. Error Counter Register (DCAN ERRC) Field Descriptions ..... 1116
5-15. Bit Timing Register (DCAN BTR) Field Descriptions ..... 1117
5-16. Interrupt Register (DCAN INT) Field Descriptions ..... 1118
5-17. Test Register (DCAN TEST) Field Descriptions ..... 1119
5-18. Parity Error Code Register (DCAN PERR) Field Descriptions ..... 1120
5-19. Auto-Bus-On Time Register (DCAN ABOTR) Field Descriptions ..... 1121
5-20. Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78) ..... 1123



Copyright © 2011–2016, Texas Instruments Incorporated

5-21. Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78) Field Descriptions ..... 1123
5-22. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) ..... 1125
5-23. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) Field Descriptions ..... 1125
5-24. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) ..... 1127
5-25. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) Field Descriptions ..... 1127
5-26. Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78) ..... 1129
5-27. Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78) Field Descriptions ..... 1129
5-28. Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78) ..... 1130
5-29. Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78) Field Descriptions ..... 1130
5-30. IF1/IF2 Command Registers (DCAN IF1CMD, DCAN IF2CMD) Field Descriptions ..... 1132
5-31. IF1/IF2 Mask Registers (DCAN IF1MSK, DCAN IF2MSK) Field Descriptions ..... 1134
5-32. IF1/IF2 Arbitration Registers (DCAN IF1ARB, DCAN IF2ARB) Field Descriptions ..... 1135
5-33. IF1/IF2 Message Control Registers (DCAN IF1MCTL, DCAN IF2MCTL) Field Descriptions ..... 1137
5-34. IF3 Observation Register (DCAN IF3OBS) Field Descriptions ..... 1140
5-35. IF3 Mask Register (DCAN IF3MSK) Field Descriptions ..... 1142
5-36. IF3 Arbitration Register (DCAN IF3ARB) Field Descriptions ..... 1143
5-37. IF3 Message Control Register (DCAN IF3MCTL) Field Descriptions ..... 1144
5-38. Update Enable Registers (DCAN IF3UPD12 to IF3UPD78)) ..... 1147
5-39. Update Enable Registers (DCAN IF3UPD12 to IF3UPD78) Field Descriptions ..... 1147
5-40. CAN TX I/O Control Register (DCAN TIOC) Field Descriptions ..... 1148
5-41. CAN RX IO Control Register (DCAN RIOC) Field Descriptions ..... 1150
6-1. Stride for Well-formed Tiled Mode 2D Block Requests ..... 1176
6-2. TILER Modes ..... 1180
6-3. Tiled Mode Container Characteristics ..... 1181
6-4. Case 1 Memory Controllers ..... 1202
6-5. Controller Configuration ..... 1202
6-6. Case 2 Memory Controllers ..... 1203
6-7. Controller Configuration ..... 1203
6-8. Section Mapping Option 1 ..... 1205
6-9. Section Mapping Option 2 ..... 1205
6-10. DMM/TILER Registers ..... 1206
6-11. DMM_REVISION Register Field Descriptions ..... 1206
6-12. DMM_SYSCONFIG Register Field Descriptions ..... 1207
6-13. DMM_LISA_LOCK Register Field Descriptions ..... 1207
6-14. DMM_LISA_MAP Registers Field Descriptions ..... 1208
6-15. DMM_TILER_OR Registers Field Descriptions ..... 1209
6-16. DMM_PAT_CONFIG Register Field Descriptions ..... 1209
6-17. DMM_PAT_VIEW Registers Field Descriptions ..... 1210
6-18. DMM_PAT_VIEW_MAP Registers Field Descriptions ..... 1211
6-19. DMM_PAT_VIEW_MAP_BASE Register Field Descriptions ..... 1212
6-20. DMM_PAT_IRQ_EOI Register Field Descriptions ..... 1212
6-21. DMM_PAT_IRQSTATUS_RAW Register Field Descriptions ..... 1214
6-22. DMM_PAT_IRQSTATUS Register Field Descriptions ..... 1216
6-23. DMM_PAT_IRQENABLE_SET Register Field Descriptions ..... 1218
6-24. DMM_PAT_IRQENABLE_CLR Register Field Descriptions ..... 1220
6-25. DMM_PAT_STATUS Registers Field Descriptions ..... 1222
6-26. DMM_PAT_DESCR Registers Field Descriptions ..... 1223
6-27. DMM_PAT_AREA Registers Field Descriptions ..... 1223
6-28. DMM_PAT_CTRL Registers Field Descriptions ..... 1224


Copyright © 2011–2016, Texas Instruments Incorporated

6-29. DMM_PAT_DATA Registers Field Descriptions ..... 1224
6-30. DMM_PEG_PRIO Registers Field Descriptions ..... 1225
6-31. DMM_PEG_PRIO_PAT Register Field Descriptions ..... 1225
7-1. DDR2 and DDR3 Memory Controller Signal Descriptions ..... 1229
7-2. Digital Filter Configuration ..... 1233
7-3. IBANK, RSIZE and PAGESIZE Fields Information ..... 1234
7-4. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=0 and EBANK_POS=0 ..... 1235
7-5. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=1 and EBANK_POS=0 ..... 1236
7-6. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=2 and EBANK_POS=0 ..... 1236
7-7. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=3 and EBANK_POS=0 ..... 1236
7-8. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=0 and EBANK_POS=1 ..... 1237
7-9. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=1 and EBANK_POS = 1 ..... 1237
7-10. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=2 and EBANK_POS = 1 ..... 1237
7-11. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=3 and EBANK_POS=1 ..... 1238
7-12. DDR2/DDR3 Memory Controller Commands Priority ..... 1239
7-13. Refresh Modes ..... 1241
7-14. Performance Counter Filter Configuration ..... 1242
7-15. Output Impedance Settings ..... 1246
7-16. Programmable Slew-rate Settings ..... 1247
7-17. ODT Settings ..... 1247
7-18. SDRCR Configuration ..... 1251
7-19. DDR2-800 Refresh Rate Specification ..... 1251
7-20. Refresh Rate Calculated Value ..... 1251
7-21. SDRTIM1 Configuration ..... 1252
7-22. SDRTIM2 Configuration ..... 1252
7-23. SDRTIM3 Configuration ..... 1252
7-24. DDR2/DDR3 Memory Controller Control Register (DDRPHYCR) ..... 1253
7-25. Configuring DDRn_IO_CTRL ..... 1253
7-26. SDRCR Configuration ..... 1254
7-27. Periodic Refresh Interval Specification ..... 1254
7-28. SDRRCR Configuration ..... 1254
7-29. SDRTIM1 Configuration ..... 1256
7-30. SDRTIM2 Configuration ..... 1257
7-31. SDRTIM3 Configuration ..... 1257
7-32. PHY Control Register (DDRPHYCR) ..... 1258
7-33. ZQCR Configuration ..... 1258
7-34. DDR2/DDR3 IO_CTRL Configuration ..... 1259
7-35. DDR2/DDR3 Memory Controller Registers ..... 1261
7-36. SDRAM Status Register (SDRSTAT) Field Descriptions ..... 1262
7-37. SDRAM Configuration Register (SDRCR) Field Descriptions ..... 1263
7-38. SDRAM Configuration Register 2 (SDRCR2) Field Descriptions ..... 1266
7-39. SDRAM Refresh Control Register (SDRRCR) Field Descriptions ..... 1267
7-40. SDRAM Refresh Control Shadow Register (SDRRCSR) Field Descriptions ..... 1268
7-41. SDRAM Timing 1 Register (SDRTIM1) Field Descriptions ..... 1269
7-42. SDRAM Timing 1 Shadow Register (SDRTIM1SR) Field Descriptions ..... 1270
7-43. SDRAM Timing 2 Register (SDRTIM2) Field Descriptions ..... 1271
7-44. SDRAM Timing 2 Shadow Register (SDRTIM2SR) Field Descriptions ..... 1272
7-45. SDRAM Timing 3 Register (SDRTIM3) Field Descriptions ..... 1273
7-46. SDRAM Timing 3 Shadow Register (SDRTIM3SR) Field Descriptions ..... 1274


Copyright © 2011–2016, Texas Instruments Incorporated

7-47. Power Management Control Register (PMCR) Field Descriptions ..... 1275
7-48. Power Management Control Shadow Register (PMCSR) Field Descriptions ..... 1277
7-49. Peripheral Bus Burst Priority Register (PBBPR) Field Descriptions ..... 1278
7-50. Performance Counter 1 Register (PERF_CNT_1) Field Descriptions ..... 1278
7-51. Performance Counter 2 Register (PERF_CNT_2) Field Descriptions ..... 1279
7-52. Performance Counter Config Register (PERF_CNT_CFG) Field Descriptions ..... 1279
7-53. Performance Counter Master Region Select Register (PERF_CNT_SEL) Field Des criptions ..... 1280
7-54. End of Interrupt Register (EOI) Field Descriptions ..... 1280
7-55. System OCP Interrup RAW Status Register (SOIRSR) Field Descriptions ..... 1281
7-56. System OCP Interrupt Status Register (SOISR) Field Descriptions ..... 1281
7-57. System OCP Interrupt Enable Set Register (SOIESR) Field Descriptions ..... 1282
7-58. System OCP Interrupt Enable Clear Register (SOIECR) Field Descriptions ..... 1282
7-59. SDRAM Output Impedance Calibration Configuration Register (ZQCR) Field Descriptions ..... 1283
7-60. DDR PHY Control Register (DDRPHYCR) Field Descriptions ..... 1284
7-61. DDR PHY Control Shadow Register (DDRPHYCSR) Field Descriptions ..... 1285
7-62. Priority to Class of Service Mapping Register (PRI_COS_MAP) Field Descriptions ..... 1286
7-63. Connection ID to Class of Service 1 Mapping Register(CONNID_COS_1_MAP) Field Descriptions ..... 1287
7-64. Connection ID to Class of Service 2 Mapping Register (CONNID_COS_2_MAP) Field Descriptions ..... 1288
7-65. Read Write Execution Threshold Register (RD_WR_EXEC_THRSH) Field Descriptions ..... 1289
7-66. Memory-Mapped Registers for DDR2 and DDR3 PHY ..... 1290
7-67. DDR PHY Command 0/1/2 Address/Command Slave Ratio Register (CMD0/1/2_REG_PHY_CTRL_SLAVE_RATIO_0) Field Descriptions ..... 1291
7-68. DDR PHY Command 0/1/2 Address/Command DLL Lock Difference Register ( CMD0/1/2_REG_PHY_DLL_LOCK_DIFF_0) Field Descriptions ..... 1292
7-69. DDR PHY Command 0/1/2 Invert Clockout Selection Register( CMD0/1/2_REG_PHY_INVERT_CLKOUT_0) Field Descriptions ..... 1292
7-70. DDR PHY Data Macro 0/1/2/3 Read DQS Slave Ratio Register (DATA0/1/2/3_REG_PHY_RD_DQS_SLAVE_RATIO_0) Field Descriptions ..... 1293
7-71. DDR PHY Data Macro 0/1/2/3 Write DQS Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DQS_SLAVE_RATIO_0) ..... 1294
7-72. DDR PHY Data Macro 0/1/2/3 Write DQS Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DQS_SLAVE_RATIO_0) Field Descriptions ..... 1294
7-73. DDR PHY Data Macro 0/1/2/3 DQS Gate Slave Ratio Register (DATA0/1/2/3_REG_PHY_FIFO_WE_SLAVE_RATIO_0) Field Descriptions ..... 1294
7-74. DDR PHY Data Macro 0/1/2/3 Write Data Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DATA_SLAVE_RATIO_0) Field Descriptions ..... 1295
7-75. DDR PHY Data Macro 0/1/2/3 Delay Selection Register (DATA0/1/2/3_REG_PHY_USE_RANK0_DELAYS) Field Descriptions ..... 1296
7-76. DDR PHY Data Macro 0/1/2/3 DLL Lock Difference Register (DATA0/1/2/3_REG_PHY_DLL_LOCK_DIFF_0) Field Descriptions ..... 1296
7-77. DDR Related Control Module Registers ..... 1297
7-78. EMIF0/1 Clock Gate Control Register Field Descriptions ..... 1297
7-79. DDR Memory Controller0_IO Control Register (DDR0_IO_CTRL) Field Descriptions ..... 1298
7-80. DDR Memory Controller1_IO Control Register (DDR1_IO_CTRL) Field Descriptions ..... 1299
7-81. DDR VTP Control Register (DDR_VTP_CTRL_0) Field Descriptions ..... 1300
7-82. DDR VTP Control Registerv 1 (DDR_VTP_CTRL_1) Field Descriptions ..... 1301
7-83. Master Connection IDs ..... 1302
8-1. EDMA3 Parameter RAM Contents ..... 1313
8-2. EDMA3 Channel Parameter Description ..... 1314
8-3. Channel Options Parameters (OPT) Field Descriptions ..... 1315
8-4. Dummy and Null Transfer Request ..... 1319


Copyright © 2011–2016, Texas Instruments Incorporated

8-5. Parameter Updates in EDMA3CC (for Non-Null, Non-Dummy PaRAM Set) ..... 1320
8-6. Expected Number of Transfers for Non-Null Transfer ..... 1326
8-7. Shadow Region Registers ..... 1330
8-8. Chain Event Triggers ..... 1332
8-9. EDMA3 Transfer Completion Interrupts ..... 1332
8-10. EDMA3 Error Interrupts ..... 1332
8-11. Transfer Complete Code (TCC) to EDMA3CC Interrupt Mapping ..... 1333
8-12. Number of Interrupts ..... 1333
8-13. Allowed Accesses ..... 1338
8-14. MPPA Registers to Region Assignment ..... 1338
8-15. Example Access Denied ..... 1339
8-16. Example Access Allowed ..... 1340
8-17. Read/Write Command Optimization Rules ..... 1344
8-18. EDMA3 Transfer Controller Configurations ..... 1346
8-19. EDMACC Registers ..... 1369
8-20. Peripheral ID Register (PID) Field Descriptions ..... 1372
8-21. EDMA3CC Configuration Register (CCCFG) Field Descriptions ..... 1373
8-22. DMA Channel Map *n* Registers (DCHMAP*n*) Field Descriptions ..... 1375
8-23. QDMA Channel Map *n* Registers (QCHMAP*n*) Field Descriptions ..... 1376
8-24. DMA Channel Queue *n* Number Registers (DMAQNUM*n*) Field Descriptions ..... 1377
8-25. Bits in DMAQNUM*n* ..... 1377
8-26. QDMA Channel Queue Number Register (QDMAQNUM) Field Descriptions ..... 1378
8-27. Event Missed Register (EMR) Field Descriptions ..... 1379
8-28. Event Missed Register High (EMRH) Field Descriptions ..... 1379
8-29. Event Missed Clear Register (EMCR) Field Descriptions ..... 1380
8-30. Event Missed Clear Register High (EMCRH) Field Descriptions ..... 1380
8-31. QDMA Event Missed Register (QEMR) Field Descriptions ..... 1381
8-32. QDMA Event Missed Clear Register (QEMCR) Field Descriptions ..... 1382
8-33. EDMA3CC Error Register (CCERR) Field Descriptions ..... 1383
8-34. EDMA3CC Error Clear Register (CCERRCLR) Field Descriptions ..... 1384
8-35. Error Evaluation Register (EEVAL) Field Descriptions ..... 1385
8-36. DMA Region Access Enable Registers for Region M (DRAE*m*/DRAEH*m*) Field Descriptions ..... 1386
8-37. QDMA Region Access Enable for Region M (QRAE*m*) Field Descriptions ..... 1387
8-38. Event Queue Entry Registers (Q*x*E*y*) Field Descriptions ..... 1388
8-39. Queue Status Register *n* (QSTAT*n*) Field Descriptions ..... 1389
8-40. Queue Watermark Threshold A Register (QWMTHRA) Field Descriptions ..... 1390
8-41. EDMA3CC Status Register (CCSTAT) Field Descriptions ..... 1391
8-42. Memory Protection Fault Address Register (MPFAR) Field Descriptions ..... 1393
8-43. Memory Protection Fault Status Register (MPFSR) Field Descriptions ..... 1394
8-44. Memory Protection Fault Command Register (MPFCR) Field Descriptions ..... 1395
8-45. Memory Protection Page Attribute Register (MPPA*n*) Field Descriptions ..... 1396
8-46. Event Register (ER) Field Descriptions ..... 1398
8-47. Event Register High (ERH) Field Descriptions ..... 1398
8-48. Event Clear Register (ECR) Field Descriptions ..... 1399
8-49. Event Clear Register High (ECRH) Field Descriptions ..... 1399
8-50. Event Set Register (ESR) Field Descriptions ..... 1400
8-51. Event Set Register High (ESRH) Field Descriptions ..... 1401
8-52. Chained Event Register (CER) Field Descriptions ..... 1402
8-53. Chained Event Register High (CERH) Field Descriptions ..... 1403


Copyright © 2011–2016, Texas Instruments Incorporated

8-54. Event Enable Register (EER) Field Descriptions ..... 1404
8-55. Event Enable Register High (EERH) Field Descriptions ..... 1404
8-56. Event Enable Clear Register (EECR) Field Descriptions ..... 1405
8-57. Event Enable Clear Register High (EECRH) Field Descriptions ..... 1405
8-58. Event Enable Set Register (EESR) Field Descriptions ..... 1406
8-59. Event Enable Set Register High (EESRH) Field Descriptions ..... 1406
8-60. Secondary Event Register (SER) Field Descriptions ..... 1407
8-61. Secondary Event Register High (SERH) Field Descriptions ..... 1407
8-62. Secondary Event Clear Register (SECR) Field Descriptions ..... 1408
8-63. Secondary Event Clear Register High (SECRH) Field Descriptions ..... 1408
8-64. Interrupt Enable Register (IER) Field Descriptions ..... 1409
8-65. Interrupt Enable Register High (IERH) Field Descriptions ..... 1409
8-66. Interrupt Enable Clear Register (IECR) Field Descriptions ..... 1410
8-67. Interrupt Enable Clear Register High (IECRH) Field Descriptions ..... 1410
8-68. Interrupt Enable Set Register (IESR) Field Descriptions ..... 1411
8-69. Interrupt Enable Set Register High (IESRH) Field Descriptions ..... 1411
8-70. Interrupt Pending Register (IPR) Field Descriptions ..... 1412
8-71. Interrupt Pending Register High (IPRH) Field Descriptions ..... 1412
8-72. Interrupt Clear Register (ICR) Field Descriptions ..... 1413
8-73. Interrupt Clear Register High (ICRH) Field Descriptions ..... 1413
8-74. Interrupt Evaluate Register (IEVAL) Field Descriptions ..... 1414
8-75. QDMA Event Register (QER) Field Descriptions ..... 1415
8-76. QDMA Event Enable Register (QEER) Field Descriptions ..... 1416
8-77. QDMA Event Enable Clear Register (QEECR) Field Descriptions ..... 1417
8-78. QDMA Event Enable Set Register (QEESR) Field Descriptions ..... 1418
8-79. QDMA Secondary Event Register (QSER) Field Descriptions ..... 1419
8-80. QDMA Secondary Event Clear Register (QSECR) Field Descriptions ..... 1420
8-81. EDMA3TC Registers ..... 1421
8-82. Peripheral ID Register (PID) Field Descriptions ..... 1422
8-83. EDMA3TC Configuration Register (TCCFG) Field Descriptions ..... 1423
8-84. EDMA3TC Channel Status Register (TCSTAT) Field Descriptions ..... 1424
8-85. Error Register (ERRSTAT) Field Descriptions ..... 1426
8-86. Error Enable Register (ERREN) Field Descriptions ..... 1427
8-87. Error Clear Register (ERRCLR) Field Descriptions ..... 1428
8-88. Error Details Register (ERRDET) Field Descriptions ..... 1429
8-89. Error Interrupt Command Register (ERRCMD) Field Descriptions ..... 1430
8-90. Read Rate Register (RDRATE) Field Descriptions ..... 1431
8-91. Source Active Options Register (SAOPT) Field Descriptions ..... 1432
8-92. Source Active Source Address Register (SASRC) Field Descriptions ..... 1434
8-93. Source Active Count Register (SACNT) Field Descriptions ..... 1434
8-94. Source Active Destination Address Register (SADST) Field Descriptions ..... 1435
8-95. Source Active Source B-Dimension Index Register (SABIDX) Field Descriptions ..... 1435
8-96. Source Active Memory Protection Proxy Register (SAMPPRXY) Field Descriptions ..... 1436
8-97. Source Active Count Reload Register (SACNTRLD) Field Descriptions ..... 1437
8-98. Source Active Source Address B-Reference Register (SASRCBREF) Field Descriptions ..... 1437
8-99. Source Active Destination Address B-Reference Register (SADSTBREF) Field Descriptions ..... 1438
8-100. Destination FIFO Options Register (DFOPT*n*) Field Descriptions ..... 1439
8-101. Destination FIFO Source Address Register (DFSRC*n*) Field Descriptions ..... 1441
8-102. Destination FIFO Count Register (DFCNT*n*) Field Descriptions ..... 1441


Copyright © 2011–2016, Texas Instruments Incorporated

8-103. Destination FIFO Destination Address Register (DFDST*n*) Field Descriptions ..... 1442
8-104. Destination FIFO B-Index Register (DFBIDX*n*) Field Descriptions ..... 1442
8-105. Destination FIFO Memory Protection Proxy Register (DFMPPRXY*n*) Field Descriptions ..... 1443
8-106. Destination FIFO Count Reload Register (DFCNTRLD*n*) Field Descriptions ..... 1444
8-107. Destination FIFO Source Address B-Reference Register (DFSRCBREF*n*) Field Descriptions ..... 1444
8-108. Destination FIFO Destination Address B-Reference Register (DFDSTBREF*n*) Field Descriptions ..... 1445
8-109. Debug List ..... 1446
9-1. G/MII Signal Description ..... 1452
9-2. RMII Signal Description ..... 1453
9-3. RGMII Signal Description ..... 1453
9-4. Learned Address Control Bits ..... 1460
9-5. Free (Unused) Address Table Entry Bit Values ..... 1460
9-6. Multicast Address Table Entry Bit Values ..... 1460
9-7. VLAN/Multicast Address Table Entry Bit Values ..... 1461
9-8. Unicast Address Table Entry Bit Values ..... 1462
9-9. OUI Unicast Address Table Entry Bit Values ..... 1463
9-10. Unicast Address Table Entry Bit Values ..... 1463
9-11. VLAN Table Entry ..... 1464
9-12. Example of TX Configuration ..... 1469
9-13. Example of RX Configuration ..... 1469
9-14. Example of Rate-limit Configurations ..... 1469
9-15. G/MII Interface Signal Descriptions in GIG (1000 Mbps) Mode ..... 1472
9-16. G/MII Interface Signal Descriptions in MII (100/10 Mbps) Mode ..... 1473
9-17. RMII Signal Descriptions ..... 1474
9-18. Link Speed ..... 1475
9-19. RGMII Signal Descriptions ..... 1476
9-20. Embedded Memories ..... 1477
9-21. Switch Latency ..... 1480
9-22. Emulation Control Input ..... 1481
9-23. Values of Message Type Field ..... 1488
9-24. MDIO Read Frame Format ..... 1495
9-25. MDIO Write Frame Format ..... 1495
9-26. CPSW_3G Module Registers ..... 1509
9-27. Register Bit Definitions ..... 1513
9-28. CPSW ID Version Register (CPSW_ID_VER) Field Descriptions ..... 1514
9-29. CPSW Control Register (CPSW_CONTROL) Field Descriptions ..... 1515
9-30. CPSW Software Reset Register (CPSW_SOFT_RESET) Field Descriptions ..... 1516
9-31. CPSW Statistics Port Enable Register (CPSW_STAT_PORT_EN) Field Descriptions ..... 1516
9-32. CPSW Transmit Priority Type Register (CPSW_PTYPE) Field Descriptions ..... 1517
9-33. CPSW Software Idle Register (CPSW_SOFT_IDLE) Field Descriptions ..... 1518
9-34. CPSW Throughput Rate Register (CPSW_THRU_RATE) Field Descriptions ..... 1518
9-35. CPSW CPGMAC_SL Short Gap Threshold Register (CPSW_GAP_THRESH) Field Descriptions ..... 1519
9-36. CPSW Transmit Start Words Register (CPSW_TX_START_WDS) Field Descriptions ..... 1519
9-37. CPSW Flow Control Register (CPSW_FLOW_CONTROL) Field Descriptions ..... 1520
9-38. CPSW Port 0 Maximum FIFO Blocks Register (P0_MAX_BLKS) Field Descriptions ..... 1520
9-39. CPSW Port 0 FIFO Block Usage Count (Read Only) Register (P0_BLK_CNT) Field Descriptions ..... 1521
9-40. CPSW Port 0 Transmit FIFO Control Register (P0_TX_IN_CTL) Field Descriptions ..... 1521
9-41. CPSW Port 0 VLAN Register (P0_PORT_VLAN) Field Descriptions ..... 1522
9-42. CPSW Port 0 TX Header Priority to Switch Priority Mapping Register (P0_TX_PRI_MAP) Field


Copyright © 2011–2016, Texas Instruments Incorporated

Descriptions ..... 1522
9-43. CPSW CPDMA TX (Port 0 RX) Packet Priority to Header Priority Mapping Register (CPDMA_TX_PRI_MAP) Field Descriptions ..... 1523
9-44. CPSW CPDMA RX (Port 0 TX) Switch Priority to DMA Channel Mapping Register (CPDMA_RX_CH_MAP) Field Descriptions ..... 1524
9-45. CPSW Port 1 Maximum FIFO Blocks Register (P1_MAX_BLKS) Field Descriptions ..... 1525
9-46. CPSW Port 1 FIFO Block Usage Count (Read Only) Register (P1_BLK_CNT) Field Descriptions ..... 1525
9-47. CPSW Port 1 Transmit FIFO Control Register (P1_TX_IN_CTL) Field Descriptions ..... 1526
9-48. CPSW Port 1 VLAN Register (P1_PORT_VLAN) Field Descriptions ..... 1526
9-49. CPSW Port 1 TX Header Priority to Switch Pri Mapping Register (P1_TX_PRI_MAP) Field Descriptions . 1527
9-50. CPSW_3G Port 1 Time Sync Control Register (P1_TS_CTL) Field Descriptions ..... 1528
9-51. CPSW_3G Port 1 Time Sync Sequence ID and LTYPE Register (P1_TS_SEQ_LTYPE) Field Descriptions ..... 1529
9-52. CPSW_3G Port 1 Time Sync VLAN2 and VLAN2 Register (P1_TS_VLAN) Field Descriptions ..... 1529
9-53. CPSW CPGMAC_SL1 Source Address Low Register (SL1_SA_LO) Field Descriptions ..... 1530
9-54. CPSW CPGMAC_SL1 Source Address High Register (SL1_SA_HI) Field Descriptions ..... 1530
9-55. CPSW Port 1 Transmit Queue Send Percentages Register (P1_SEND_PERCENT) Field Descriptions ... 1531
9-56. CPSW Port 2 Maximum FIFO Blocks Register (P2_MAX_BLKS) Field Descriptions ..... 1532
9-57. CPSW Port 2 FIFO Block Usage Count (Read Only) Register (P2_BLK_CNT) Field Descriptions ..... 1532
9-58. CPSW Port 2 Transmit FIFO Control Register (P2_TX_IN_CTL) Field Descriptions ..... 1533
9-59. CPSW Port 2 VLAN Register (P2_PORT_VLAN) Field Descriptions ..... 1533
9-60. CPSW Port 2 TX Header Priority to Switch Priority Mapping Register (P2_TX_PRI_MAP) Field Descriptions ..... 1534
9-61. CPSW_3GF Port 2 Time Sync Control Register (P2_TS_CTL) Field Descriptions ..... 1535
9-62. Port 2 Time Sync Sequence ID and LTYPE Register (P2_TS_SEQ_LTYPE) Field Descriptions ..... 1536
9-63. CPSW_3GF Port 2 Time Sync VLAN2 and VLAN2 Register (P2_TS_VLAN) Field Descriptions ..... 1536
9-64. CPSW CPGMAC_SL2 Source Address Low Register (SL2_SA_LO) Field Descriptions ..... 1537
9-65. CPGMAC_SL2 Source Address High Register (SL2_SA_HI) Field Descriptions ..... 1537
9-66. Port 2 Transmit Queue Send Percentages Register (P2_SEND_PERCENT) Field Descriptions ..... 1538
9-67. CPDMA_REGS TX Identification and Version Register (TX_IDVER) Field Descriptions ..... 1539
9-68. CPDMA_REGS TX Control Register (TX_CONTROL) Field Descriptions ..... 1539
9-69. CPDMA_REGS TX Teardown Register (TX_TEARDOWN) Field Descriptions ..... 1539
9-70. CPDMA_REGS RX Identification and Version Register (RX_IDVER) Field Descriptions ..... 1540
9-71. CPDMA_REGS RX Control Register (RX_CONTROL) Field Descriptions ..... 1540
9-72. CPDMA_REGS RX Teardown Register (RX_TEARDOWN) Field Descriptions ..... 1540
9-73. CPDMA_REGS Software Reset Register (SOFT_RESET) Field Descriptions ..... 1541
9-74. CPDMA_REGS Control Register (DMACONTROL) Field Descriptions ..... 1541
9-75. CPDMA_REGS Status Register (DMASTATUS) Field Descriptions ..... 1543
9-76. CPDMA_REGS Receive Buffer Offset Register (RX_BUFFER_OFFSET) Field Descriptions ..... 1544
9-77. CPDMA_REGS Emulation Control Register (EMCONTROL) Field Descriptions ..... 1544
9-78. CPDMA_REGS Transmit Priority Rate Registers Field Descriptions ..... 1544
9-79. CPDMA_INT TX Interrupt Status (raw value) Register (TX_INTSTAT_RAW) Field Descriptions ..... 1545
9-80. CPDMA_INT TX Interrupt Status (masked value) Register (TX_INTSTAT_MASKED) Field Descriptions . 1545
9-81. CPDMA_INT TX Interrupt Mask Set Register (TX _INTMASK_SET) Field Descriptions ..... 1546
9-82. CPDMA_INT TX Interrupt Mask Clear Register (TX_INTMASK_CLEAR) Field Descriptions ..... 1546
9-83. CPDMA_INT Input Vector (CPDMA_IN_VECTOR) Field Descriptions ..... 1547
9-84. CPDMA_INT End Of Interrupt Vector (CPDMA_EOI_VECTOR) Field Descriptions ..... 1547
9-85. CPDMA_INT RX Interrupt Status (raw value) Register (RX_INTSTAT_RAW) Field Descriptions ..... 1548
9-86. CPDMA_INT RX Interrupt Status (masked value) Register (RX_INTSTAT_MASKED) Field Descriptions . 1549
9-87. CPDMA_INT RX Interrupt Mask Set Register (RX_INTMASK_SET) Field Descriptions ..... 1550


Copyright © 2011–2016, Texas Instruments Incorporated

9-88. CPDMA_INT RX Interrupt Mask Clear Register (RX_INTMASK_CLEAR) Field Descriptions ..... 1551
9-89. CPDMA_INT DMA Interrupt Status (raw value) Register (DMA_INTSTAT_RAW) Field Descriptions ..... 1552
9-90. CPDMA_INT DMA Interrupt Status (masked value) Register (DMA_INTSTAT_MASKED) Field Descriptions ..... 1552
9-91. CPDMA_INT DMA Interrupt Mask Set Register (DMA_INTMASK_SET) Field Descriptions ..... 1553
9-92. CPDMA_INT DMA Interrupt Mask Clear Register (DMA_INTMASK_CLEAR) Field Descriptions ..... 1553
9-93. CPDMA_INT RX Channel Threshold Pending Channel Registers Field Descriptions ..... 1554
9-94. CPDMA_INT RX Channel Free Buffer Count Registers Field Descriptions ..... 1554
9-95. CPDMA_STATERAM TX Channel Head Descriptor Pointers Field Descriptions ..... 1554
9-96. CPDMA_STATERAM RX Channel Head Descriptor Pointers Field Descriptions ..... 1555
9-97. CPDMA_STATERAM TX Channel Completion Pointer Registers Field Descriptions ..... 1555
9-98. CPDMA_STATERAM RX Channel Completion Pointer Registers Field Descriptions ..... 1555
9-99. Identification and Version Register (CPTS_IDVER) Field Descriptions ..... 1565
9-100. Time Sync Control Register (CPTS_CONTROL) Field Descriptions ..... 1565
9-101. Reference Clock Select Register (CPTS_RFTCLK_SEL) Field Descriptions ..... 1566
9-102. Time Stamp Event Push Register (CPTS_TS_PUSH) Field Descriptions ..... 1566
9-103. Time Stamp Load Value Register (CPTS_TS_LOAD_VAL) Field Descriptions ..... 1566
9-104. Time Stamp Load Enable Register (CPTS_TS_LOAD_EN) Field Descriptions ..... 1567
9-105. Time Sync Interrupt Status Register Raw Register (CPTS_INTSTAT_RAW) Field Descriptions ..... 1567
9-106. Time Sync Interrupt Status Register Masked Register (CPTS_INTSTAT_MASKED) Field Descriptions ..... 1567
9-107. Time Sync Interrupt Enable Register (CPTS_INT_ENABLE) Field Descriptions ..... 1568
9-108. Event Interrupt Pop Register (CPTS_EVENT_POP) Field Descriptions ..... 1568
9-109. Event Low Register (CPTS_EVENT_LOW) Field Descriptions ..... 1568
9-110. Event High Register (CPTS_EVENT_HIGH) Field Descriptions ..... 1569
9-111. Address Lookup Engine ID/Version Register (ALE_IDVER) Field Descriptions ..... 1569
9-112. ALE Control Register (ALE_CONTROL) Field Descriptions ..... 1570
9-113. ALE Prescale Register (ALE_PRESCALE) Field Descriptions ..... 1571
9-114. ALE Unknown VLAN Register (ALE_UNKNOWN_VLAN) Field Descriptions ..... 1572
9-115. ALE Table Control Register (ALE_TBLCTL) Field Descriptions ..... 1572
9-116. ALE Table Word 2 Register (ALE_TBLW2) Field Descriptions ..... 1573
9-117. ALE Table Word 1 Register (ALE_TBLW1) Field Descriptions ..... 1573
9-118. ALE Table Word 0 Register (ALE_TBLW0) Field Descriptions ..... 1573
9-119. ALE Port Control Registers Field Descriptions ..... 1574
9-120. CPGMAC_SL1 IDVER Register (SL1_IDVER) Field Descriptions ..... 1575
9-121. CPGMAC_SL1 MAC Control Register (SL1_MACCONTROL) Field Descriptions ..... 1575
9-122. CPGMAC_SL1 MAC Status Register (SL1_MACSTATUS) Field Descriptions ..... 1577
9-123. CPGMAC_SL1 Software Reset Register (SL1_SOFT_RESET) Field Descriptions ..... 1578
9-124. CPGMAC_SL1 RX Maximum Length Register (SL1_RX_MAXLEN) Field Descriptions ..... 1578
9-125. CPGMAC_SL1 Backoff Random Number Generator Test Register (SL1_BOFFTEST) Field Descriptions 1579
9-126. CPGMAC_SL1 RX Pause Timer Register (SL1_RX_PAUSE) Field Descriptions ..... 1579
9-127. CPGMAC_SL1 TX Pause Timer Register (SL1_TX_PAUSE) Field Descriptions ..... 1580
9-128. CPGMAC_SL1 Emulation Control Register (SL1_EMCONTROL) Field Descriptions ..... 1580
9-129. CPGMAC_SL1 RX Packet Priority to Header Priority Mapping Register (SL1_Pri_Map) Field Descriptions ..... 1581
9-130. CPGMAC_SL2 IDVER Register (SL2_IDVER) Field Descriptions ..... 1582
9-131. CPGMAC_SL2 MAC Control Register (SL2_MACCONTROL) Field Descriptions ..... 1582
9-132. CPGMAC_SL2 MAC Status Register (SL2_MACSTATUS) Field Descriptions ..... 1585
9-133. CPGMAC_SL2 Software Reset Register (SL2_SOFT_RESET) Field Descriptions ..... 1585
9-134. CPGMAC_SL2 RX Maximum Length Register (SL2_RX_MAXLEN) Field Descriptions ..... 1586


Copyright © 2011–2016, Texas Instruments Incorporated

9-135. CPGMAC_SL2 Backoff Random Number Generator Test Register (SL2_BOFFTEST) Field Descriptions 1586
9-136. CPGMAC_SL2 RX Pause Timer Register (SL2_RX_PAUSE) Field Descriptions 1587
9-137. CPGMAC_SL2 TX Pause Timer Register (SL2_TX_PAUSE) Field Descriptions 1587
9-138. CPGMAC_SL2 Emulation Control Register (SL2_EMCONTROL) Field Descriptions 1588
9-139. CPGMAC_SL2 RX Packet Priority to Header Priority Mapping Register (SL2_RX_PRI_MAP) Field Descriptions 1589
9-140. MDIO Module Registers 1590
9-141. MDIO Version Register (VERSION) Field Descriptions 1590
9-142. MDIO Control Register (CONTROL) Field Descriptions 1591
9-143. PHY Alive Status Register (ALIVE) Field Descriptions 1592
9-144. PHY Link Status Register (LINK) Field Descriptions 1592
9-145. MDIO Link Status Change Interrupt (raw value) Register (LINKINTRAW) Field Descriptions 1593
9-146. MDIO Link Status Change Interrupt (masked value) Register (LINKINTMASKED) Field Descriptions 1593
9-147. MDIO User Command Complete Interrupt (raw value) Register (USERINTRAW) Field Descriptions 1594
9-148. MDIO User Command Complete Interrupt (masked value) Register (USERINTMASKED) Field Descriptions 1594
9-149. MDIO User Command Complete Interrupt Mask Set Register (USERINTMASKSET) Field Descriptions 1595
9-150. MDIO User Command Complete Interrupt Mask Clear Register (USERINTMASKCLEAR) Field Descriptions 1595
9-151. MDIO User Access Register 0 (USERACCESS0) Field Descriptions 1596
9-152. MDIO User PHY Select Register 0 (USERPHYSEL0) Field Descriptions 1596
9-153. MDIO User Access Register 1 (USERACCESS1) Field Descriptions 1597
9-154. MDIO User PHY Select Register 1 (USERPHYSEL1) Field Descriptions 1597
9-155. CPSW Subsytem Module Registers 1598
9-156. Subsystem ID Version Register (IDVER) Field Descriptions 1598
9-157. Subsystem Software Reset Register (SOFT_RESET) Field Descriptions 1599
9-158. Subsystem Control Register (CONTROL) Field Descriptions 1599
9-159. Subsystem Interrupt Control Register (INT_CONTROL) Field Descriptions 1600
9-160. Subsystem Receive Threshold Interrupt Enable Register (RX_THRESH_EN) Field Descriptions 1600
9-161. Subsystem Receive Interrupt Enable Register (RX_EN) Field Descriptions 1601
9-162. Subsystem Transmit Interrupt Enable Register (TX_EN) Field Descriptions 1601
9-163. Subsystem Miscellaneous Interrupt Enable Register (MISC_EN) Field Descriptions 1601
9-164. Subsystem Receive Threshold Masked Interrupt Status Register (RX_THRESH_STAT) Field Descriptions 1602
9-165. Subsystem Receive Masked Interrupt Status Register (RX_STAT) Field Descriptions 1602
9-166. Subsystem Transmit Masked Interrupt Status Register (TX_STAT) Field Descriptions 1602
9-167. Subsystem Miscellaneous Masked Interrupt Status Register (MISC_STAT) Field Descriptions 1603
9-168. Subsystem Receive Interrupts per Millisecond Register (RX_IMAX) Field Descriptions 1603
9-169. Subsystem Transmit Interrupts per Millisecond Register (TX_IMAX) Field Descriptions 1603
9-170. RGMII Control Read Register (RGMII_CTL) Field Descriptions 1604
10-1. GPIO Registers 1615
10-2. GPIO_REVISION Register Field Descriptions 1616
10-3. GPIO_SYSCONFIG Register Field Descriptions 1617
10-4. GPIO_EOI Register Field Descriptions 1618
10-5. GPIO_IRQSTATUS_RAW_0 Register Field Descriptions 1619
10-6. GPIO_IRQSTATUS_RAW_1 Register Field Descriptions 1619
10-7. GPIO_IRQSTATUS_0 Register Field Descriptions 1620
10-8. GPIO_IRQSTATUS_1 Register Field Descriptions 1620
10-9. GPIO_IRQENABLE_SET_0 Register Field Descriptions 1621
10-10. GPIO_IRQENABLE_SET_1 Register Field Descriptions 1621


Copyright © 2011–2016, Texas Instruments Incorporated

10-11. GPIO_IRQENABLE_CLR_0 Register Field Descriptions ..... 1622
10-12. GPIO_IRQENABLE_CLR_1 Register Field Descriptions ..... 1622
10-13. GPIO_SYSSTATUS Register Field Descriptions ..... 1623
10-14. GPIO_CTRL Register Field Descriptions ..... 1624
10-15. GPIO_OE Register Field Descriptions ..... 1624
10-16. GPIO_DATAIN Register Field Descriptions ..... 1625
10-17. GPIO_DATAOUT Register Field Descriptions ..... 1625
10-18. GPIO_LEVELDETECT0 Register Field Descriptions ..... 1626
10-19. GPIO_LEVELDETECT1 Register Field Descriptions ..... 1626
10-20. GPIO_RISINGDETECT Register Field Descriptions ..... 1627
10-21. GPIO_FALLINGDETECT Register Field Descriptions ..... 1627
10-22. GPIO_DEBOUNCENABLE Register Field Descriptions ..... 1628
10-23. GPIO_DEBOUNCINGTIME Register Field Descriptions ..... 1628
10-24. GPIO_CLEARDATAOUT Register Field Descriptions ..... 1629
10-25. GPIO_SETDATAOUT Register Field Descriptions ..... 1629
11-1. GPMC I/O Description ..... 1633
11-2. GPMC Pin Multiplexing Options ..... 1633
11-3. GPMC Integration Attributes ..... 1637
11-4. GPMC Clocks and Resets ..... 1638
11-5. GPMC Hardware Requests ..... 1638
11-6. GPMC Clocks ..... 1639
11-7. GPMC_CONFIG1_i Configuration ..... 1639
11-8. GPMC Local Power Management Features ..... 1639
11-9. GPMC Interrupt Events ..... 1640
11-10. Idle Cycle Insertion Configuration ..... 1651
11-11. Chip-Select Configuration for NAND Interfacing ..... 1680
11-12. ECC Enable Settings ..... 1689
11-13. Flattened BCH Codeword Mapping (512 Bytes + 104 Bits) ..... 1694
11-14. Aligned Message Byte Mapping in 8-bit NAND ..... 1694
11-15. Aligned Message Byte Mapping in 16-bit NAND ..... 1695
11-16. Aligned Nibble Mapping of Message in 8-bit NAND ..... 1695
11-17. Misaligned Nibble Mapping of Message in 8-bit NAND ..... 1695
11-18. Aligned Nibble Mapping of Message in 16-bit NAND ..... 1695
11-19. Misaligned Nibble Mapping of Message in 16-bit NAND (1 Unused Nibble) ..... 1696
11-20. Misaligned Nibble Mapping of Message in 16-bit NAND (2 Unused Nibble) ..... 1696
11-21. Misaligned Nibble Mapping of Message in 16-bit NAND (3 Unused Nibble) ..... 1696
11-22. Prefetch Mode Configuration ..... 1707
11-23. GPMC Configuration in NOR Mode ..... 1712
11-24. GPMC Configuration in NAND Mode ..... 1712
11-25. Reset GPMC ..... 1714
11-26. NOR Memory Type ..... 1714
11-27. NOR Chip-Select Configuration ..... 1714
11-28. NOR Timings Configuration ..... 1714
11-29. WAIT Pin Configuration ..... 1715
11-30. Enable Chip-Select ..... 1715
11-31. NAND Memory Type ..... 1715
11-32. NAND Chip-Select Configuration ..... 1715
11-33. Asynchronous Read and Write Operations ..... 1715
11-34. ECC Engine ..... 1716


Copyright © 2011–2016, Texas Instruments Incorporated

11-35. Prefetch and Write-Posting Engine ..... 1716
11-36. WAIT Pin Configuration ..... 1716
11-37. Enable Chip-Select ..... 1716
11-38. Mode Parameters Check List Table ..... 1717
11-39. Access Type Parameters Check List Table ..... 1717
11-40. Timing Parameters ..... 1719
11-41. NAND Formulas Description Table ..... 1721
11-42. Synchronous NOR Formulas Description Table ..... 1722
11-43. Asynchronous NOR Formulas Description Table ..... 1729
11-44. GPMC Signals ..... 1731
11-45. Useful Timing Parameters on the Memory Side ..... 1733
11-46. Calculating GPMC Timing Parameters ..... 1734
11-47. AC Characteristics for Asynchronous Read Access ..... 1735
11-48. GPMC Timing Parameters for Asynchronous Read Access ..... 1736
11-49. AC Characteristics for Asynchronous Single Write (Memory Side) ..... 1737
11-50. GPMC Timing Parameters for Asynchronous Single Write ..... 1738
11-51. Supported Memory Interfaces ..... 1739
11-52. NAND Interface Bus Operations Summary ..... 1741
11-53. NOR Interface Bus Operations Summary ..... 1741
11-54. GPMC Registers ..... 1743
11-55. GPMC_REVISION Field Descriptions ..... 1744
11-56. GPMC_SYSCONFIG Field Descriptions ..... 1744
11-57. GPMC_SYSSTATUS Field Descriptions ..... 1745
11-58. GPMC_IRQSTATUS Field Descriptions ..... 1746
11-59. GPMC_IRQENABLE Field Descriptions ..... 1747
11-60. GPMC_TIMEOUT_CONTROL Field Descriptions ..... 1748
11-61. GPMC_ERR_ADDRESS Field Descriptions ..... 1748
11-62. GPMC_ERR_TYPE Field Descriptions ..... 1749
11-63. GPMC_CONFIG Field Descriptions ..... 1750
11-64. GPMC_STATUS Field Descriptions ..... 1751
11-65. GPMC_CONFIG1_i Field Descriptions ..... 1752
11-66. GPMC_CONFIG2_i Field Descriptions ..... 1755
11-67. GPMC_CONFIG3_i Field Descriptions ..... 1756
11-68. GPMC_CONFIG4_i Field Descriptions ..... 1758
11-69. GPMC_CONFIG5_i Field Descriptions ..... 1760
11-70. GPMC_CONFIG6_i Field Descriptions ..... 1761
11-71. GPMC_CONFIG7_i Field Descriptions ..... 1762
11-72. GPMC_NAND_COMMAND_i Field Descriptions ..... 1763
11-73. GPMC_NAND_ADDRESS_i Field Descriptions ..... 1763
11-74. GPMC_NAND_DATA_i Field Descriptions ..... 1763
11-75. GPMC_PREFETCH_CONFIG1 Field Descriptions ..... 1764
11-76. GPMC_PREFETCH_CONFIG2 Field Descriptions ..... 1766
11-77. GPMC_PREFETCH_CONTROL Field Descriptions ..... 1766
11-78. GPMC_PREFETCH_STATUS Field Descriptions ..... 1767
11-79. GPMC_ECC_CONFIG Field Descriptions ..... 1768
11-80. GPMC_ECC_CONTROL Field Descriptions ..... 1769
11-81. GPMC_ECC_SIZE_CONFIG Field Descriptions ..... 1770
11-82. GPMC_ECCj_RESULT Field Descriptions ..... 1772
11-83. GPMC_BCH_RESULT0_i Field Descriptions ..... 1773


Copyright © 2011–2016, Texas Instruments Incorporated

11-84. GPMC_BCH_RESULT1_i Field Descriptions ..... 1773
11-85. GPMC_BCH_RESULT2_i Field Descriptions ..... 1773
11-86. GPMC_BCH_RESULT3_i Field Descriptions ..... 1774
11-87. GPMC_BCH_RESULT4_i Field Descriptions ..... 1774
11-88. GPMC_BCH_RESULT5_i Field Descriptions ..... 1775
11-89. GPMC_BCH_RESULT6_i Field Descriptions ..... 1775
11-90. GPMC_BCH_SWDATA Field Descriptions ..... 1775
12-1. HDVPSS Acronyms ..... 1777
12-2. HDVPSS Data Format ..... 1777
12-3. VPDMA Client Number ..... 1783
12-4. HDVPSS Mux Select Register 1 ..... 1791
12-5. HDVPSS Mux Select Register 2 ..... 1792
12-6. HDVPSS Mux Select Register 3 ..... 1793
12-7. HDVPSS Mux Select Register 4 ..... 1795
12-8. HDVPSS Mux Select Register 5 ..... 1797
12-9. HDVPSS Mux Select Register 6 ..... 1799
12-10. HDVPSS Mux Select Register 7 ..... 1801
12-11. HDVPSS Interrupts ..... 1802
12-12. VPDMA Modes of Operation ..... 1811
12-13. Quantized Coefficients of HDTV Application with Video Data Range ..... 1815
12-14. Quantized Coefficients of HDTV Application with Graphics Data Range ..... 1815
12-15. Quantized Coefficients of SDTV Application with Video Data Range ..... 1818
12-16. Quantized Coefficients of SDTV Application with Graphics Data Range ..... 1818
12-17. Input Layers and Associated Blenders ..... 1820
12-18. Frame Configuration Attribute (Configuration descriptor payload word 0-3) ..... 1831
12-19. Region Configuration Attribute (Inbound data descriptor word 4-7) ..... 1832
12-20. Blending and Transparency Configuration ..... 1835
12-21. Region Scaler Configuration Attributes 1 ..... 1836
12-22. Region Scaler Attributes 2 ..... 1836
12-23. Region Scaler Attributes 3 ..... 1836
12-24. Region Scaler Attributes 4 ..... 1837
12-25. Region Scaler Attributes 5 ..... 1837
12-26. Region Scaler Attributes 6 ..... 1838
12-27. Region Scaler Attributes 7 ..... 1838
12-28. Region Scaler Attributes 8 ..... 1838
12-29. Region Scaler Attributes 9 ..... 1839
12-30. Region Scaler Attributes 10 ..... 1839
12-31. Region Scaler Attributes 11 ..... 1840
12-32. TV Formats ..... 1844
12-33. SDTV Horizontal Timing Parameter ..... 1848
12-34. Non-Interlaced Format ..... 1851
12-35. Color Bar Table ..... 1856
12-36. 2× Interpolation Filter Coefficients ..... 1856
12-37. Color Systems ..... 1857
12-38. Luma LPF Filter Coefficient ..... 1863
12-39. Chroma LPF Filter Coefficient ..... 1863
12-40. Sub-carrier Increment Parameters ..... 1865
12-41. Typical Register Setting for Various Video Standards ..... 1872
12-42. DAC Output Select ..... 1873


Copyright © 2011–2016, Texas Instruments Incorporated

12-43. DAC 2x Oversampling Filter Coefficients ..... 1873
12-44. Display Ports and VENC Names ..... 1875
12-45. OSD Interface Signals ..... 1876
12-46. DVO Formats ..... 1881
12-47. Definition of SAV and EAV words ..... 1881
12-48. JEIDA and VESA Format ..... 1884
12-49. Programmable Capacity of First Line Encoder ..... 1885
12-50. Programmable Capacity of Second Line Encoder ..... 1885
12-51. Gamma Correction LUT ..... 1885
12-52. Memory Map Addresses for Configuration Register and LUT Memories ..... 1886
12-53. Valid Input Port Configurations ..... 1893
12-54. Polarity Table for FID Determination By VSYNC Skew ..... 1900
12-55. Fourth Byte of EAV/SAV Code Word ..... 1904
12-56. Error Correction Matrix ..... 1905
12-57. Multiplexing Configurations and Pixel Clock Rates ..... 1908
12-58. Split Line Table ..... 1911
12-59. Meta Data Layout ..... 1911
12-60. TI Line Mux Mode Channel ID Remapping ..... 1912
12-61. Channel ID Embedded in EAV/SAV ..... 1913
12-62. Valid Embedded Sync Mux Mode and Data Bus Width Combinations ..... 1914
12-63. Register Layout ..... 1914
12-64. VIP Parser Interrupts ..... 1915
12-65. Module Interrupts ..... 1931
12-66. NF Configuration for Each Basic Mode ..... 1937
12-67. Vertical Scaler Configuration Parameters ..... 1943
12-68. Horizontal Scaler Related Parameter Configuration ..... 1947
12-69. Horizontal Decimation Selection ..... 1947
12-70. Horizontal Polyphase Filter Related Parameter Configuration ..... 1948
12-71. Coefficient Data Files ..... 1952
12-72. Memory Databus Write Order ..... 1982
12-73. Data Packet Descriptor Word 0 Field Descriptions ..... 1984
12-74. Data Packet Descriptor Word 1 Field Description ..... 1986
12-75. Data Packet Descriptor Word 2 Field Descriptions ..... 1987
12-76. Data Packet Descriptor Word 3 Field Descriptions ..... 1987
12-77. Priority Bit Fields ..... 1989
12-78. Data Packet Descriptor Word 4 Inbound Data Field Descriptions ..... 1989
12-79. Data Packet Descriptor Word 4 Outbound Data Field Descriptions ..... 1990
12-80. Data Packet Descriptor Word 5 Inbound Data Field Descriptions ..... 1991
12-81. Data Packet Descriptor Word 5 Outbound Data Field Descriptions ..... 1991
12-82. Configuration Descriptor Header Word0 Field Descriptions ..... 1992
12-83. Configuration Descriptor Header Word1 Field Descriptions ..... 1992
12-84. Configuration Descriptor Header Word2 Field Descriptions ..... 1992
12-85. Configuration Descriptor Header Word3 Field Descriptions ..... 1993
12-86. Address Data Block Format Field Descriptions ..... 1993
12-87. Destination Field Description ..... 1994
12-88. Control Descriptor Header Description ..... 1994
12-89. Control Descriptor Types Summary ..... 1995
12-90. Sync on Client Field Descriptions (Word - 1) ..... 1995
12-91. Sync on Client Field Descriptions (Word - 3) ..... 1996


Copyright © 2011–2016, Texas Instruments Incorporated

12-92. Sync on List Field Descriptions (Word - 3) ..... 1996
12-93. Sync on External Event Field Descriptions (Word - 3) ..... 1996
12-94. Sync on Channel Field Descriptions (Word - 3) ..... 1997
12-95. Change Client Interrupt Field Descriptions (Word - 1) ..... 1997
12-96. Change Client Interrupt Field Descriptions (Word - 2) ..... 1997
12-97. Change Client Interrupt Field Descriptions (Word - 3) ..... 1997
12-98. Send Interrupt Field Descriptions (Word - 3) ..... 1998
12-99. Reload List Field Descriptions (Word - 0) ..... 1998
12-100. Reload List Field Descriptions (Word - 1) ..... 1998
12-101. Reload List Field Descriptions (Word - 3) ..... 1998
12-102. Abort Channel Field Descriptions (Word - 3) ..... 1999
12-103. Toggle LM Field Field Descriptions (Word - 2) ..... 1999
12-104. Toggle LM Field Field Descriptions (Word - 3) ..... 1999
12-105. VPDMA Channels ..... 2001
12-106. VPDMA Client Buffering ..... 2008
12-107. HDVPSS Client Functionality ..... 2011
12-108. HDVPSS Interrupt from VPDMA ..... 2012
12-109. HDVPSS Interrupt Sources ..... 2013
12-110. Graphics Region Descriptor Format ..... 2030
12-111. Frame Configuration Descriptor Format ..... 2031
12-112. Frame with Resizer Configuration Descriptor Format ..... 2031
12-113. Example Graphics List Description ..... 2035
12-114. HDVPSS Module Instances ..... 2061
12-115. HDVPSS Registers ..... 2062
12-116. CHR_US REGISTERS ..... 2063
12-117. CHR_US_reg0 Register Field Descriptions ..... 2064
12-118. CHR_US_reg1 Register Field Descriptions ..... 2065
12-119. CHR_US_reg2 Register Field Descriptions ..... 2066
12-120. CHR_US_reg3 Register Field Descriptions ..... 2067
12-121. CHR_US_reg4 Register Field Descriptions ..... 2068
12-122. CHR_US_reg5 Register Field Descriptions ..... 2069
12-123. CHR_US_reg6 Register Field Descriptions ..... 2070
12-124. CHR_US_reg7 Register Field Descriptions ..... 2071
12-125. CIG REGISTERS ..... 2072
12-126. CIG_reg0 Register Field Descriptions ..... 2073
12-127. CIG_reg1 Register Field Descriptions ..... 2074
12-128. CIG_reg2 Register Field Descriptions ..... 2075
12-129. CIG_reg3 Register Field Descriptions ..... 2076
12-130. CIG_reg4 Register Field Descriptions ..... 2077
12-131. CIG_reg5 Register Field Descriptions ..... 2078
12-132. CIG_reg6 Register Field Descriptions ..... 2079
12-133. CIG_reg7 Register Field Descriptions ..... 2080
12-134. CIG_reg8 Register Field Descriptions ..... 2081
12-135. CIG_reg9 Register Field Descriptions ..... 2082
12-136. CIG_reg10 Register Field Descriptions ..... 2083
12-137. COMP REGISTERS ..... 2084
12-138. COMP_hdmi_settings Register Field Descriptions ..... 2086
12-139. COMP_dvo2_settings Register Field Descriptions ..... 2087
12-140. COMP_sd_settings Register Field Descriptions ..... 2088


Copyright © 2011–2016, Texas Instruments Incorporated

12-141. COMP_back_color_settings Register Field Descriptions ..... 2089
12-142. CSC REGISTERS ..... 2090
12-143. CSC_csc00 Register Field Descriptions ..... 2091
12-144. CSC_csc01 Register Field Descriptions ..... 2092
12-145. CSC_csc02 Register Field Descriptions ..... 2093
12-146. CSC_csc03 Register Field Descriptions ..... 2094
12-147. CSC_csc04 Register Field Descriptions ..... 2095
12-148. CSC_csc05 Register Field Descriptions ..... 2096
12-149. DEI REGISTERS ..... 2097
12-150. dei_reg0 Register Field Descriptions ..... 2098
12-151. dei_reg1 Register Field Descriptions ..... 2099
12-152. dei_reg2 Register Field Descriptions ..... 2100
12-153. dei_reg3 Register Field Descriptions ..... 2101
12-154. dei_reg4 Register Field Descriptions ..... 2102
12-155. dei_reg5 Register Field Descriptions ..... 2103
12-156. dei_reg6 Register Field Descriptions ..... 2104
12-157. dei_reg7 Register Field Descriptions ..... 2105
12-158. dei_reg8 Register Field Descriptions ..... 2106
12-159. dei_reg9 Register Field Descriptions ..... 2107
12-160. dei_reg10 Register Field Descriptions ..... 2108
12-161. dei_reg11 Register Field Descriptions ..... 2109
12-162. dei_reg12 Register Field Descriptions ..... 2110
12-163. dei_reg13 Register Field Descriptions ..... 2111
12-164. dei_reg14 Register Field Descriptions ..... 2112
12-165. INTC_CLKC_CONTROL REGISTERS ..... 2114
12-166. pid Register Field Descriptions ..... 2116
12-167. sysconfig Register Field Descriptions ..... 2117
12-168. intc_intr0_status_raw0 Register Field Descriptions ..... 2118
12-169. intc_intr0_status_raw1 Register Field Descriptions ..... 2120
12-170. intc_intr0_status_ena0 Register Field Descriptions ..... 2122
12-171. intc_intr0_status_ena1 Register Field Descriptions ..... 2124
12-172. intc_intr0_ena_set0 Register Field Descriptions ..... 2126
12-173. intc_intr0_ena_set1 Register Field Descriptions ..... 2128
12-174. intc_intr0_ena_clr0 Register Field Descriptions ..... 2130
12-175. intc_intr0_ena_clr1 Register Field Descriptions ..... 2132
12-176. intc_intr1_status_raw0 Register Field Descriptions ..... 2134
12-177. intc_intr1_status_raw1 Register Field Descriptions ..... 2136
12-178. intc_intr1_status_ena0 Register Field Descriptions ..... 2138
12-179. intc_intr1_status_ena1 Register Field Descriptions ..... 2140
12-180. intc_intr1_ena_set0 Register Field Descriptions ..... 2142
12-181. intc_intr1_ena_set1 Register Field Descriptions ..... 2144
12-182. intc_intr1_ena_clr0 Register Field Descriptions ..... 2146
12-183. intc_intr1_ena_clr1 Register Field Descriptions ..... 2148
12-184. intc_intr2_status_raw0 Register Field Descriptions ..... 2150
12-185. intc_intr2_status_raw1 Register Field Descriptions ..... 2152
12-186. intc_intr2_status_ena0 Register Field Descriptions ..... 2154
12-187. intc_intr2_status_ena1 Register Field Descriptions ..... 2156
12-188. intc_intr2_ena_set0 Register Field Descriptions ..... 2158
12-189. intc_intr2_ena_set1 Register Field Descriptions ..... 2160


Copyright © 2011–2016, Texas Instruments Incorporated

12-190. intc_intr2_ena_clr0 Register Field Descriptions ..... 2162
12-191. intc_intr2_ena_clr1 Register Field Descriptions ..... 2164
12-192. intc_intr3_status_raw0 Register Field Descriptions ..... 2166
12-193. intc_intr3_status_raw1 Register Field Descriptions ..... 2168
12-194. intc_intr3_status_ena0 Register Field Descriptions ..... 2170
12-195. intc_intr3_status_ena1 Register Field Descriptions ..... 2172
12-196. intc_intr3_ena_set0 Register Field Descriptions ..... 2174
12-197. intc_intr3_ena_set1 Register Field Descriptions ..... 2176
12-198. intc_intr3_ena_clr0 Register Field Descriptions ..... 2178
12-199. intc_intr3_ena_clr1 Register Field Descriptions ..... 2180
12-200. intc_eoi Register Field Descriptions ..... 2182
12-201. clkc_clken Register Field Descriptions ..... 2183
12-202. clkc_dps Register Field Descriptions ..... 2185
12-203. clkc_vip1dps Register Field Descriptions ..... 2187
12-204. clkc_vip2dps Register Field Descriptions ..... 2190
12-205. clkc_venc_ena Register Field Descriptions ..... 2193
12-206. clkc_range_map Register Field Descriptions ..... 2194
12-207. clkc_underflow Register Field Descriptions ..... 2195
12-208. VPDMA REGISTERS ..... 2196
12-209. VPDMA_pid Register Field Descriptions ..... 2199
12-210. VPDMA_list_addr Register Field Descriptions ..... 2200
12-211. VPDMA_list_attr Register Field Descriptions ..... 2201
12-212. VPDMA_list_stat_sync Register Field Descriptions ..... 2202
12-213. VPDMA_vpi_ctl_address Register Field Descriptions ..... 2204
12-214. VPDMA_vpi_ctl_data Register Field Descriptions ..... 2205
12-215. VPDMA_bg_rgb Register Field Descriptions ..... 2206
12-216. VPDMA_bg_yuv Register Field Descriptions ..... 2207
12-217. VPDMA_descriptor_top Register Field Descriptions ..... 2208
12-218. VPDMA_descriptor_bottom Register Field Descriptions ..... 2209
12-219. VPDMA_current_descriptor Register Field Descriptions ..... 2210
12-220. VPDMA_descriptor_status_control Register Field Descriptions ..... 2211
12-221. VPDMA_int0_channel0_int_stat Register Field Descriptions ..... 2212
12-222. VPDMA_int0_channel0_int_mask Register Field Descriptions ..... 2215
12-223. VPDMA_int0_channel1_int_stat Register Field Descriptions ..... 2217
12-224. VPDMA_int0_channel1_int_mask Register Field Descriptions ..... 2221
12-225. VPDMA_int0_channel2_int_stat Register Field Descriptions ..... 2224
12-226. VPDMA_int0_channel2_int_mask Register Field Descriptions ..... 2228
12-227. VPDMA_int0_channel3_int_stat Register Field Descriptions ..... 2231
12-228. VPDMA_int0_channel3_int_mask Register Field Descriptions ..... 2235
12-229. VPDMA_int0_channel4_int_stat Register Field Descriptions ..... 2238
12-230. VPDMA_int0_channel4_int_mask Register Field Descriptions ..... 2242
12-231. VPDMA_int0_channel5_int_stat Register Field Descriptions ..... 2245
12-232. VPDMA_int0_channel5_int_mask Register Field Descriptions ..... 2249
12-233. VPDMA_int0_channel6_int_stat Register Field Descriptions ..... 2252
12-234. VPDMA_int0_channel6_int_mask Register Field Descriptions ..... 2253
12-235. VPDMA_int0_client0_int_stat Register Field Descriptions ..... 2254
12-236. VPDMA_int0_client0_int_mask Register Field Descriptions ..... 2257
12-237. VPDMA_int0_client1_int_stat Register Field Descriptions ..... 2259
12-238. VPDMA_int0_client1_int_mask Register Field Descriptions ..... 2263


Copyright © 2011–2016, Texas Instruments Incorporated

12-239. VPDMA_int0_list0_int_stat Register Field Descriptions ..... 2266
12-240. VPDMA_int0_list0_int_mask Register Field Descriptions ..... 2269
12-241. VPDMA_int1_channel0_int_stat Register Field Descriptions ..... 2272
12-242. VPDMA_int1_channel0_int_mask Register Field Descriptions ..... 2275
12-243. VPDMA_int1_channel1_int_stat Register Field Descriptions ..... 2277
12-244. VPDMA_int1_channel1_int_mask Register Field Descriptions ..... 2281
12-245. VPDMA_int1_channel2_int_stat Register Field Descriptions ..... 2284
12-246. VPDMA_int1_channel2_int_mask Register Field Descriptions ..... 2288
12-247. VPDMA_int1_channel3_int_stat Register Field Descriptions ..... 2291
12-248. VPDMA_int1_channel3_int_mask Register Field Descriptions ..... 2295
12-249. VPDMA_int1_channel4_int_stat Register Field Descriptions ..... 2298
12-250. VPDMA_int1_channel4_int_mask Register Field Descriptions ..... 2302
12-251. VPDMA_int1_channel5_int_stat Register Field Descriptions ..... 2305
12-252. VPDMA_int1_channel5_int_mask Register Field Descriptions ..... 2309
12-253. VPDMA_int1_channel6_int_stat Register Field Descriptions ..... 2312
12-254. VPDMA_int1_channel6_int_mask Register Field Descriptions ..... 2313
12-255. VPDMA_int1_client0_int_stat Register Field Descriptions ..... 2314
12-256. VPDMA_int1_client0_int_mask Register Field Descriptions ..... 2317
12-257. VPDMA_int1_client1_int_stat Register Field Descriptions ..... 2319
12-258. VPDMA_int1_client1_int_mask Register Field Descriptions ..... 2323
12-259. VPDMA_int1_list0_int_stat Register Field Descriptions ..... 2326
12-260. VPDMA_int1_list0_int_mask Register Field Descriptions ..... 2329
12-261. VPDMA_int2_channel0_int_stat Register Field Descriptions ..... 2332
12-262. VPDMA_int2_channel0_int_mask Register Field Descriptions ..... 2335
12-263. VPDMA_int2_channel1_int_stat Register Field Descriptions ..... 2337
12-264. VPDMA_int2_channel1_int_mask Register Field Descriptions ..... 2341
12-265. VPDMA_int2_channel2_int_stat Register Field Descriptions ..... 2344
12-266. VPDMA_int2_channel2_int_mask Register Field Descriptions ..... 2348
12-267. VPDMA_int2_channel3_int_stat Register Field Descriptions ..... 2351
12-268. VPDMA_int2_channel3_int_mask Register Field Descriptions ..... 2355
12-269. VPDMA_int2_channel4_int_stat Register Field Descriptions ..... 2358
12-270. VPDMA_int2_channel4_int_mask Register Field Descriptions ..... 2362
12-271. VPDMA_int2_channel5_int_stat Register Field Descriptions ..... 2365
12-272. VPDMA_int2_channel5_int_mask Register Field Descriptions ..... 2369
12-273. VPDMA_int2_channel6_int_stat Register Field Descriptions ..... 2372
12-274. VPDMA_int2_channel6_int_mask Register Field Descriptions ..... 2373
12-275. VPDMA_int2_client0_int_stat Register Field Descriptions ..... 2374
12-276. VPDMA_int2_client0_int_mask Register Field Descriptions ..... 2377
12-277. VPDMA_int2_client1_int_stat Register Field Descriptions ..... 2379
12-278. VPDMA_int2_client1_int_mask Register Field Descriptions ..... 2383
12-279. VPDMA_int2_list0_int_stat Register Field Descriptions ..... 2386
12-280. VPDMA_int2_list0_int_mask Register Field Descriptions ..... 2389
12-281. VPDMA_int3_channel0_int_stat Register Field Descriptions ..... 2392
12-282. VPDMA_int3_channel0_int_mask Register Field Descriptions ..... 2395
12-283. VPDMA_int3_channel1_int_stat Register Field Descriptions ..... 2397
12-284. VPDMA_int3_channel1_int_mask Register Field Descriptions ..... 2401
12-285. VPDMA_int3_channel2_int_stat Register Field Descriptions ..... 2404
12-286. VPDMA_int3_channel2_int_mask Register Field Descriptions ..... 2408
12-287. VPDMA_int3_channel3_int_stat Register Field Descriptions ..... 2411


Copyright © 2011–2016, Texas Instruments Incorporated

12-288. VPDMA_int3_channel3_int_mask Register Field Descriptions ..... 2415
12-289. VPDMA_int3_channel4_int_stat Register Field Descriptions ..... 2418
12-290. VPDMA_int3_channel4_int_mask Register Field Descriptions ..... 2422
12-291. VPDMA_int3_channel5_int_stat Register Field Descriptions ..... 2425
12-292. VPDMA_int3_channel5_int_mask Register Field Descriptions ..... 2429
12-293. VPDMA_int3_channel6_int_stat Register Field Descriptions ..... 2432
12-294. VPDMA_int3_channel6_int_mask Register Field Descriptions ..... 2433
12-295. VPDMA_int3_client0_int_stat Register Field Descriptions ..... 2434
12-296. VPDMA_int3_client0_int_mask Register Field Descriptions ..... 2437
12-297. VPDMA_int3_client1_int_stat Register Field Descriptions ..... 2439
12-298. VPDMA_int3_client1_int_mask Register Field Descriptions ..... 2443
12-299. VPDMA_int3_list0_int_stat Register Field Descriptions ..... 2446
12-300. VPDMA_int3_list0_int_mask Register Field Descriptions ..... 2449
12-301. VPDMA_dei_hq_1_chroma_cstat Register Field Descriptions ..... 2452
12-302. VPDMA_dei_hq_1_luma_cstat Register Field Descriptions ..... 2453
12-303. VPDMA_dei_hq_2_luma_cstat Register Field Descriptions ..... 2454
12-304. VPDMA_dei_hq_2_chroma_cstat Register Field Descriptions ..... 2455
12-305. VPDMA_dei_hq_3_luma_cstat Register Field Descriptions ..... 2456
12-306. VPDMA_dei_hq_3_chroma_cstat Register Field Descriptions ..... 2457
12-307. VPDMA_dei_hq_mv_in_cstat Register Field Descriptions ..... 2458
12-308. VPDMA_dei_hq_mv_out_cstat Register Field Descriptions ..... 2459
12-309. VPDMA_dei_sc_out_cstat Register Field Descriptions ..... 2460
12-310. VPDMA_pip_wrbk_cstat Register Field Descriptions ..... 2461
12-311. VPDMA_sc_in_chroma_cstat Register Field Descriptions ..... 2462
12-312. VPDMA_sc_in_luma_cstat Register Field Descriptions ..... 2463
12-313. VPDMA_sc_out_cstat Register Field Descriptions ..... 2464
12-314. VPDMA_comp_wrbk_cstat Register Field Descriptions ..... 2465
12-315. VPDMA_grpx1_data_cstat Register Field Descriptions ..... 2466
12-316. VPDMA_grpx2_data_cstat Register Field Descriptions ..... 2467
12-317. VPDMA_grpx3_data_cstat Register Field Descriptions ..... 2468
12-318. VPDMA_vip1_lo_y_cstat Register Field Descriptions ..... 2469
12-319. VPDMA_vip1_lo_uv_cstat Register Field Descriptions ..... 2470
12-320. VPDMA_vip1_up_y_cstat Register Field Descriptions ..... 2471
12-321. VPDMA_vip1_up_uv_cstat Register Field Descriptions ..... 2472
12-322. VPDMA_vip2_lo_y_cstat Register Field Descriptions ..... 2473
12-323. VPDMA_vip2_lo_uv_cstat Register Field Descriptions ..... 2474
12-324. VPDMA_vip2_up_y_cstat Register Field Descriptions ..... 2475
12-325. VPDMA_vip2_up_uv_cstat Register Field Descriptions ..... 2476
12-326. VPDMA_grpx1_st_cstat Register Field Descriptions ..... 2477
12-327. VPDMA_grpx2_st_cstat Register Field Descriptions ..... 2478
12-328. VPDMA_grpx3_st_cstat Register Field Descriptions ..... 2479
12-329. VPDMA_nf_422_in_cstat Register Field Descriptions ..... 2480
12-330. VPDMA_nf_420_y_in_cstat Register Field Descriptions ..... 2481
12-331. VPDMA_nf_420_uv_in_cstat Register Field Descriptions ..... 2482
12-332. VPDMA_nf_420_y_out_cstat Register Field Descriptions ..... 2483
12-333. VPDMA_nf_420_uv_out_cstat Register Field Descriptions ..... 2484
12-334. VPDMA_vbi_sdvenc_cstat Register Field Descriptions ..... 2485
12-335. VPDMA_vpi_ctl_cstat Register Field Descriptions ..... 2486
12-336. VPDMA_hdmi_wrbk_out_cstat Register Field Descriptions ..... 2487


Copyright © 2011–2016, Texas Instruments Incorporated

12-337. VPDMA_trans1_chroma_cstat Register Field Descriptions ..... 2488
12-338. VPDMA_trans1_luma_cstat Register Field Descriptions ..... 2489
12-339. VPDMA_trans2_chroma_cstat Register Field Descriptions ..... 2490
12-340. VPDMA_trans2_luma_cstat Register Field Descriptions ..... 2491
12-341. VPDMA_vip1_anc_a_cstat Register Field Descriptions ..... 2492
12-342. VPDMA_vip1_anc_b_cstat Register Field Descriptions ..... 2493
12-343. VPDMA_vip2_anc_a_cstat Register Field Descriptions ..... 2494
12-344. VPDMA_vip2_anc_b_cstat Register Field Descriptions ..... 2495
12-345. HD_VENC_D REGISTERS ..... 2496
12-346. HD_VENC_D_cfg0 Register Field Descriptions ..... 2497
12-347. HD_VENC_D_cfg1 Register Field Descriptions ..... 2499
12-348. HD_VENC_D_cfg2 Register Field Descriptions ..... 2500
12-349. HD_VENC_D_cfg3 Register Field Descriptions ..... 2501
12-350. HD_VENC_D_cfg4 Register Field Descriptions ..... 2502
12-351. HD_VENC_D_cfg5 Register Field Descriptions ..... 2503
12-352. HD_VENC_D_cfg6 Register Field Descriptions ..... 2504
12-353. HD_VENC_D_cfg7 Register Field Descriptions ..... 2505
12-354. HD_VENC_D_cfg8 Register Field Descriptions ..... 2506
12-355. HD_VENC_D_cfg9 Register Field Descriptions ..... 2507
12-356. HD_VENC_D_cfg10 Register Field Descriptions ..... 2508
12-357. HD_VENC_D_cfg11 Register Field Descriptions ..... 2509
12-358. HD_VENC_D_cfg12 Register Field Descriptions ..... 2510
12-359. HD_VENC_D_cfg13 Register Field Descriptions ..... 2511
12-360. HD_VENC_D_cfg14 Register Field Descriptions ..... 2512
12-361. HD_VENC_D_cfg15 Register Field Descriptions ..... 2513
12-362. HD_VENC_D_cfg16 Register Field Descriptions ..... 2514
12-363. HD_VENC_D_cfg17 Register Field Descriptions ..... 2515
12-364. HD_VENC_D_cfg18 Register Field Descriptions ..... 2516
12-365. HD_VENC_D_cfg19 Register Field Descriptions ..... 2517
12-366. HD_VENC_D_cfg20 Register Field Descriptions ..... 2518
12-367. HD_VENC_D_cfg21 Register Field Descriptions ..... 2519
12-368. HD_VENC_D_cfg22 Register Field Descriptions ..... 2520
12-369. HD_VENC_D_cfg23 Register Field Descriptions ..... 2521
12-370. HD_VENC_D_cfg24 Register Field Descriptions ..... 2522
12-371. HD_VENC_D_cfg25 Register Field Descriptions ..... 2523
12-372. HD_VEND_D_GAMMA_LUT Register Field Descriptions ..... 2524
12-373. NOISE_FILTER REGISTERS ..... 2525
12-374. nf_reg0 Register Field Descriptions ..... 2526
12-375. nf_reg1 Register Field Descriptions ..... 2527
12-376. nf_reg2 Register Field Descriptions ..... 2528
12-377. nf_reg3 Register Field Descriptions ..... 2529
12-378. nf_reg4 Register Field Descriptions ..... 2530
12-379. nf_reg5 Register Field Descriptions ..... 2531
12-380. nf_reg6 Register Field Descriptions ..... 2532
12-381. nf_reg7 Register Field Descriptions ..... 2533
12-382. nf_reg8 Register Field Descriptions ..... 2534
12-383. nf_reg9 Register Field Descriptions ..... 2535
12-384. SC_M REGISTERS ..... 2536
12-385. SC_M_cfg_sc0 Register Field Descriptions ..... 2537


Copyright © 2011–2016, Texas Instruments Incorporated

12-386. SC_M_cfg_sc1 Register Field Descriptions ..... 2539
12-387. SC_M_cfg_sc2 Register Field Descriptions ..... 2540
12-388. SC_M_cfg_sc3 Register Field Descriptions ..... 2541
12-389. SC_M_cfg_sc4 Register Field Descriptions ..... 2542
12-390. SC_M_cfg_sc5 Register Field Descriptions ..... 2543
12-391. SC_M_cfg_sc6 Register Field Descriptions ..... 2544
12-392. SC_M_cfg_sc8 Register Field Descriptions ..... 2545
12-393. SC_M_cfg_sc9 Register Field Descriptions ..... 2546
12-394. SC_M_cfg_sc10 Register Field Descriptions ..... 2547
12-395. SC_M_cfg_sc11 Register Field Descriptions ..... 2548
12-396. SC_M_cfg_sc12 Register Field Descriptions ..... 2549
12-397. SC_M_cfg_sc13 Register Field Descriptions ..... 2550
12-398. SC_M_cfg_sc17 Register Field Descriptions ..... 2551
12-399. SC_M_cfg_sc18 Register Field Descriptions ..... 2552
12-400. SC_M_cfg_sc19 Register Field Descriptions ..... 2553
12-401. SC_M_cfg_sc20 Register Field Descriptions ..... 2554
12-402. SC_M_cfg_sc21 Register Field Descriptions ..... 2555
12-403. SC_M_cfg_sc22 Register Field Descriptions ..... 2556
12-404. SC_M_cfg_sc23 Register Field Descriptions ..... 2557
12-405. SC_M_cfg_sc24 Register Field Descriptions ..... 2558
12-406. SC_M_cfg_sc25 Register Field Descriptions ..... 2559
12-407. SD_VENC REGISTERS ..... 2560
12-408. SD_VENC_pid Register Field Descriptions ..... 2562
12-409. SD_VENC_vmod Register Field Descriptions ..... 2563
12-410. SD_VENC_slave Register Field Descriptions ..... 2564
12-411. SD_VENC_size Register Field Descriptions ..... 2565
12-412. SD_VENC_pol Register Field Descriptions ..... 2566
12-413. SD_VENC_dtvs0 Register Field Descriptions ..... 2567
12-414. SD_VENC_dtvs1 Register Field Descriptions ..... 2568
12-415. SD_VENC_dtvs2 Register Field Descriptions ..... 2569
12-416. SD_VENC_dtvs3 Register Field Descriptions ..... 2570
12-417. SD_VENC_dtvs4 Register Field Descriptions ..... 2571
12-418. SD_VENC_dtvs5 Register Field Descriptions ..... 2572
12-419. SD_VENC_dtvs6 Register Field Descriptions ..... 2573
12-420. SD_VENC_dtvs7 Register Field Descriptions ..... 2574
12-421. SD_VENC_tvdetgp0 Register Field Descriptions ..... 2575
12-422. SD_VENC_tvdetgp1 Register Field Descriptions ..... 2576
12-423. SD_VENC_irq0 Register Field Descriptions ..... 2577
12-424. SD_VENC_estat Register Field Descriptions ..... 2578
12-425. SD_VENC_ectl Register Field Descriptions ..... 2579
12-426. SD_VENC_etmg0 Register Field Descriptions ..... 2580
12-427. SD_VENC_etmg1 Register Field Descriptions ..... 2581
12-428. SD_VENC_etmg2 Register Field Descriptions ..... 2582
12-429. SD_VENC_etmg3 Register Field Descriptions ..... 2583
12-430. SD_VENC_etmg4 Register Field Descriptions ..... 2584
12-431. SD_VENC_cvbs0 Register Field Descriptions ..... 2585
12-432. SD_VENC_cvbs1 Register Field Descriptions ..... 2586
12-433. SD_VENC_ccsc0 Register Field Descriptions ..... 2587
12-434. SD_VENC_ccsc1 Register Field Descriptions ..... 2588


Copyright © 2011–2016, Texas Instruments Incorporated

12-435. SD_VENC_ccsc2 Register Field Descriptions ..... 2589
12-436. SD_VENC_ccsc3 Register Field Descriptions ..... 2590
12-437. SD_VENC_ccsc4 Register Field Descriptions ..... 2591
12-438. SD_VENC_ccsc5 Register Field Descriptions ..... 2592
12-439. SD_VENC_ccsc6 Register Field Descriptions ..... 2593
12-440. SD_VENC_ccsc7 Register Field Descriptions ..... 2594
12-441. SD_VENC_ccsc8 Register Field Descriptions ..... 2595
12-442. SD_VENC_cygclp Register Field Descriptions ..... 2596
12-443. SD_VENC_cubclp Register Field Descriptions ..... 2597
12-444. SD_VENC_cvrclp Register Field Descriptions ..... 2598
12-445. SD_VENC_ylpf0 Register Field Descriptions ..... 2599
12-446. SD_VENC_ylpf1 Register Field Descriptions ..... 2600
12-447. SD_VENC_clpf0 Register Field Descriptions ..... 2601
12-448. SD_VENC_clpf1 Register Field Descriptions ..... 2602
12-449. SD_VENC_upf0 Register Field Descriptions ..... 2603
12-450. SD_VENC_upf1 Register Field Descriptions ..... 2604
12-451. SD_VENC_l21ctl Register Field Descriptions ..... 2605
12-452. SD_VENC_l21do Register Field Descriptions ..... 2606
12-453. SD_VENC_l21de Register Field Descriptions ..... 2607
12-454. SD_VENC_wss Register Field Descriptions ..... 2608
12-455. SD_VENC_scctl0 Register Field Descriptions ..... 2609
12-456. SD_VENC_scctl1 Register Field Descriptions ..... 2610
12-457. SD_VENC_dacsel Register Field Descriptions ..... 2611
12-458. SD_VENC_dupf0 Register Field Descriptions ..... 2612
12-459. SD_VENC_dupf1 Register Field Descriptions ..... 2613
12-460. SD_VENC_dactst Register Field Descriptions ..... 2614
12-461. SD_VENC_vtest Register Field Descriptions ..... 2615
12-462. VCOMP REGISTERS ..... 2616
12-463. VCOMP_reg0 Register Field Descriptions ..... 2617
12-464. VCOMP_reg1 Register Field Descriptions ..... 2618
12-465. VCOMP_reg2 Register Field Descriptions ..... 2619
12-466. VCOMP_reg3 Register Field Descriptions ..... 2620
12-467. VCOMP_reg4 Register Field Descriptions ..... 2621
12-468. VCOMP_reg5 Register Field Descriptions ..... 2622
12-469. VCOMP_reg6 Register Field Descriptions ..... 2623
12-470. VCOMP_reg7 Register Field Descriptions ..... 2624
12-471. VCOMP_reg8 Register Field Descriptions ..... 2625
12-472. VCOMP_reg9 Register Field Descriptions ..... 2626
12-473. VCOMP_reg10 Register Field Descriptions ..... 2627
12-474. VCOMP_reg11 Register Field Descriptions ..... 2628
12-475. VCOMP_reg12 Register Field Descriptions ..... 2629
12-476. VIP_PARSER REGISTERS ..... 2630
12-477. VIP_PARSER_main Register Field Descriptions ..... 2632
12-478. VIP_PARSER_port_a Register Field Descriptions ..... 2633
12-479. VIP_PARSER_xtra_port_a Register Field Descriptions ..... 2636
12-480. VIP_PARSER_port_b Register Field Descriptions ..... 2637
12-481. VIP_PARSER_xtra_port_b Register Field Descriptions ..... 2640
12-482. VIP_PARSER_fiq_mask Register Field Descriptions ..... 2641
12-483. VIP_PARSER_fiq_clear Register Field Descriptions ..... 2643


Copyright © 2011–2016, Texas Instruments Incorporated

12-484. VIP_PARSER_fiq_status Register Field Descriptions ..... 2645
12-485. VIP_PARSER_output_port_a_src_fid Register Field Descriptions ..... 2647
12-486. VIP_PARSER_output_port_a_enc_fid Register Field Descriptions ..... 2649
12-487. VIP_PARSER_output_port_b_src_fid Register Field Descriptions ..... 2651
12-488. VIP_PARSER_output_port_b_enc_fid Register Field Descriptions ..... 2653
12-489. VIP_PARSER_output_port_a_src0_size Register Field Descriptions ..... 2655
12-490. VIP_PARSER_output_port_a_src1_size Register Field Descriptions ..... 2656
12-491. VIP_PARSER_output_port_a_src2_size Register Field Descriptions ..... 2657
12-492. VIP_PARSER_output_port_a_src3_size Register Field Descriptions ..... 2658
12-493. VIP_PARSER_output_port_a_src4_size Register Field Descriptions ..... 2659
12-494. VIP_PARSER_output_port_a_src5_size Register Field Descriptions ..... 2660
12-495. VIP_PARSER_output_port_a_src6_size Register Field Descriptions ..... 2661
12-496. VIP_PARSER_output_port_a_src7_size Register Field Descriptions ..... 2662
12-497. VIP_PARSER_output_port_a_src8_size Register Field Descriptions ..... 2663
12-498. VIP_PARSER_output_port_a_src9_size Register Field Descriptions ..... 2664
12-499. VIP_PARSER_output_port_a_src10_size Register Field Descriptions ..... 2665
12-500. VIP_PARSER_output_port_a_src11_size Register Field Descriptions ..... 2666
12-501. VIP_PARSER_output_port_a_src12_size Register Field Descriptions ..... 2667
12-502. VIP_PARSER_output_port_a_src13_size Register Field Descriptions ..... 2668
12-503. VIP_PARSER_output_port_a_src14_size Register Field Descriptions ..... 2669
12-504. VIP_PARSER_output_port_a_src15_size Register Field Descriptions ..... 2670
12-505. VIP_PARSER_output_port_b_src0_size Register Field Descriptions ..... 2671
12-506. VIP_PARSER_output_port_b_src1_size Register Field Descriptions ..... 2672
12-507. VIP_PARSER_output_port_b_src2_size Register Field Descriptions ..... 2673
12-508. VIP_PARSER_output_port_b_src3_size Register Field Descriptions ..... 2674
12-509. VIP_PARSER_output_port_b_src4_size Register Field Descriptions ..... 2675
12-510. VIP_PARSER_output_port_b_src5_size Register Field Descriptions ..... 2676
12-511. VIP_PARSER_output_port_b_src6_size Register Field Descriptions ..... 2677
12-512. VIP_PARSER_output_port_b_src7_size Register Field Descriptions ..... 2678
12-513. VIP_PARSER_output_port_b_src8_size Register Field Descriptions ..... 2679
12-514. VIP_PARSER_output_port_b_src9_size Register Field Descriptions ..... 2680
12-515. VIP_PARSER_output_port_b_src10_size Register Field Descriptions ..... 2681
12-516. VIP_PARSER_output_port_b_src11_size Register Field Descriptions ..... 2682
12-517. VIP_PARSER_output_port_b_src12_size Register Field Descriptions ..... 2683
12-518. VIP_PARSER_output_port_b_src13_size Register Field Descriptions ..... 2684
12-519. VIP_PARSER_output_port_b_src14_size Register Field Descriptions ..... 2685
12-520. VIP_PARSER_output_port_b_src15_size Register Field Descriptions ..... 2686
12-521. VIP_PARSER_port_a_vdet_vec Register Field Descriptions ..... 2687
12-522. VIP_PARSER_port_b_vdet_vec Register Field Descriptions ..... 2688
12-523. VIP_PARSER_xtra2_port_a Register Field Descriptions ..... 2689
12-524. VIP_PARSER_xtra3_port_a Register Field Descriptions ..... 2690
12-525. VIP_PARSER_xtra4_port_a Register Field Descriptions ..... 2691
12-526. VIP_PARSER_xtra5_port_a Register Field Descriptions ..... 2692
12-527. VIP_PARSER_xtra2_port_b Register Field Descriptions ..... 2693
12-528. VIP_PARSER_xtra3_port_b Register Field Descriptions ..... 2694
12-529. VIP_PARSER_xtra4_port_b Register Field Descriptions ..... 2695
12-530. VIP_PARSER_xtra5_port_b Register Field Descriptions ..... 2696
12-531. VIP_PARSER_xtra6_port_a Register Field Descriptions ..... 2697
12-532. VIP_PARSER_xtra7_port_b Register Field Descriptions ..... 2698


Copyright © 2011–2016, Texas Instruments Incorporated

13-1. HDMI Video Timings (CEA-861-D) ..... 2702
13-2. HDMI Video Timings (VESA DMT) ..... 2702
13-3. CEC Clock Generation ..... 2704
13-4. HDMI I/O Signal Description ..... 2705
13-5. Integration Attributes ..... 2706
13-6. Clocks and Resets ..... 2707
13-7. Hardware Requests ..... 2707
13-8. Local Power-Management Features ..... 2707
13-9. HDMI Interrupt Events ..... 2708
13-10. Video Port Signals ..... 2709
13-11. HDMI Video Port Mapping ..... 2709
13-12. PCM, 16-Bit Format ..... 2712
13-13. PCM, 24-Bit Format ..... 2713
13-14. Speaker Mapping Versus Channel ..... 2713
13-15. IEC 60958 Sample Format ..... 2716
13-16. IEC 60937 Format ..... 2716
13-17. HDMI Wrapper Registers ..... 2720
13-18. IP Revision Identifier Register (HDMI_WP_REVISION) Field Descriptions ..... 2720
13-19. Clock Management Configuration Register (HDMI_WP_SYSCONFIG) Field Descriptions ..... 2721
13-20. Raw Interrupt Status Register (HDMI_WP_IRQSTATUS_RAW) Field Descriptions ..... 2722
13-21. Interrupt Status Register (HDMI_WP_IRQSTATUS) Field Descriptions ..... 2723
13-22. Interrupt Enable Register (HDMI_WP_IRQENABLE_SET) Field Descriptions ..... 2724
13-23. Interrupt Disable Register (HDMI_WP_IRQENABLE_CLEAR) Field Descriptions ..... 2725
13-24. Glitch Filter Register (HDMI_WP_DEBOUNCE) Field Descriptions ..... 2726
13-25. Configuration of HDMI Wrapper Video Register (HDMI_WP_VIDEO_CFG) Field Descriptions ..... 2727
13-26. Configuration of Clocks Register (HDMI_WP_CLK) Field Descriptions ..... 2728
13-27. Audio Configuration in FIFO Register (HDMI_WP_AUDIO_CFG) Field Descriptions ..... 2729
13-28. Audio Configuration of DMA Register (HDMI_WP_AUDIO_CFG2) Field Descriptions ..... 2730
13-29. Audio FIFO Control Register (HDMI_WP_AUDIO_CTRL) Field Descriptions ..... 2731
13-30. TX Data of FIFO Register (HDMI_WP_AUDIO_DATA) Field Descriptions ..... 2732
13-31. HDMI Core System Registers ..... 2733
13-32. Vendor ID Register (VND_IDL) Field Descriptions ..... 2735
13-33. Vendor ID Register (VND_IDH) Field Descriptions ..... 2736
13-34. Device IDL Register (DEV_IDL) Field Descriptions ..... 2736
13-35. Device IDH Register (DEV_IDH) Field Descriptions ..... 2736
13-36. Device Revision Register (DEV_REV) Field Descriptions ..... 2737
13-37. Software Reset Register (SRST) Field Descriptions ..... 2737
13-38. System Control Register 1 (SYS_CTRL1) Field Descriptions ..... 2738
13-39. System Status Register (SYS_STAT) Field Descriptions ..... 2739
13-40. System Control Register 3 (SYS_CTRL3) Field Descriptions ..... 2739
13-41. Data Control Register (DCTL) Field Descriptions ..... 2740
13-42. HDCP Control Register (HDCP_CTRL) Field Descriptions ..... 2741
13-43. HDCP BKSV Register (BKSV__0-BKSV__4) Field Descriptions ..... 2742
13-44. HDCP AN Register (AN__0-AN__7) Field Descriptions ..... 2742
13-45. HDCP AKSV Register (AKSV__0-AKSV__4) Field Descriptions ..... 2742
13-46. HDCP Ri Register (RI1) Field Descriptions ..... 2743
13-47. HDCP Ri2 Register (RI2) Field Descriptions ..... 2743
13-48. HDCP Ri 128 Compare Register (RI_128_COMP) Field Descriptions ..... 2743
13-49. HDCP I Counter Register (I_CNT) Field Descriptions ..... 2744


Copyright © 2011–2016, Texas Instruments Incorporated

13-50. Ri Status Register (RI_STAT) Field Descriptions ..... 2744
13-51. Ri Command Register (RI_CMD) Field Descriptions ..... 2745
13-52. Ri Line Start Register (RI_START) Field Descriptions ..... 2745
13-53. Ri From RX Register (Low) (RI_RX_L) Field Descriptions ..... 2746
13-54. Ri From RX Registers (High) (RI_RX_H) Field Descriptions ..... 2746
13-55. Ri Debug Registers (RI_DEBUG) Field Descriptions ..... 2747
13-56. VIDEO DE Delay Register (DE_DLY) Field Descriptions ..... 2747
13-57. VIDEO DE Control Register (DE_CTRL) Field Descriptions ..... 2748
13-58. VIDEO DE Top Register (DE_TOP) Field Descriptions ..... 2749
13-59. VIDEO DE Count Register (DE_CNTL) Field Descriptions ..... 2749
13-60. VIDEO DE Count Register (DE_CNTH) Field Descriptions ..... 2749
13-61. VIDEO DE Line Register (DE_LINL) Field Descriptions ..... 2750
13-62. VIDEO DE Line Register (DE_LINH_1) Field Descriptions ..... 2750
13-63. Video H Resolution Register (HRES_L) Field Descriptions ..... 2751
13-64. Video H Resolution Register (HRES_H) Field Descriptions ..... 2751
13-65. Video V Resolution Low Register (VRES_L) Field Descriptions ..... 2752
13-66. Video V Resolution Register (VRES_H) Field Descriptions ..... 2752
13-67. Video Interlace Adjustment Register (IADJUST) Field Descriptions ..... 2753
13-68. Video SYNC Polarity Detection Register (POL_DETECT) Field Descriptions ..... 2754
13-69. Video Hbit to HSYNC Register (HBIT_2HSYNC1) Field Descriptions ..... 2755
13-70. Video Hbit to HSYNC Register (HBIT_2HSYNC2) Field Descriptions ..... 2755
13-71. Video Field2 HSYNC Offset Register (FLD2_HS_OFSTL) Field Descriptions ..... 2756
13-72. Video Field2 HSYNC Offset Register (FLD2_HS_OFSTH) Field Descriptions ..... 2756
13-73. Video HSYNC Length Register (HWIDTH1) Field Descriptions ..... 2757
13-74. Video HSYNC Length Register (HWIDTH2) Field Descriptions ..... 2757
13-75. Video Vbit to VSYNC Register (VBIT_TO_VSYNC) Field Descriptions ..... 2758
13-76. Video VSYNC Length Register (VWIDTH) Field Descriptions ..... 2758
13-77. Video Control Register (VID_CTRL) Field Descriptions ..... 2759
13-78. Video Action Enable Register (VID_ACEN) Field Descriptions ..... 2760
13-79. Video Mode1 Register (VID_MODE) Field Descriptions ..... 2761
13-80. Video Blanking Registers (VID_BLANK1) Field Descriptions ..... 2762
13-81. Video Blanking Register (VID_BLANK2) Field Descriptions ..... 2762
13-82. Video Blanking Register (VID_BLANK3) Field Descriptions ..... 2762
13-83. Deep Color Header Register (DC_HEADER) Field Descriptions ..... 2763
13-84. Video Mode2 Register (VID_DITHER) Field Descriptions ..... 2764
13-85. RGB_2_xvYCC control Register (RGB2XVYCC_CT) Field Descriptions ..... 2765
13-86. RGB_2_xvYCC Conversion R_2_Y Register (R2Y_COEFF_LOW) Field Descriptions ..... 2766
13-87. RGB_2_xvYCC Conversion R_2_Y Register (R2Y_COEFF_UP) Field Descriptions ..... 2766
13-88. RGB_2_xvYCC Conversion G_2_Y Register (G2Y_COEFF_LOW) Field Descriptions ..... 2767
13-89. RGB_2_xvYCC Conversion G_2_Y Register (G2Y_COEFF_UP) Field Descriptions ..... 2767
13-90. RGB_2_xvYCC Conversion B_2_Y Register (B2Y_COEFF_LOW) Field Descriptions ..... 2768
13-91. RGB_2_xvYCC Conversion B_2_Y Register (B2Y_COEFF_UP) Field Descriptions ..... 2768
13-92. RGB_2_xvYCC Conversion R_2_Cb Register (R2CB_COEFF_LOW) Field Descriptions ..... 2769
13-93. RGB_2_xvYCC Conversion R_2_Cb Register (R2CB_COEFF_UP) Field Descriptions ..... 2769
13-94. RGB_2_xvYCC Conversion G_2_Cb Register (G2CB_COEFF_LOW) Field Descriptions ..... 2770
13-95. RGB_2_xvYCC Conversion G_2_Cb Register (G2CB_COEFF_UP) Field Descriptions ..... 2770
13-96. RGB_2_xvYCC Conversion B_2_Cb Register (B2CB_COEFF_LOW) Field Descriptions ..... 2771
13-97. RGB_2_xvYCC Conversion B_2_Cb Register (B2CB_COEFF_UP) Field Descriptions ..... 2771
13-98. RGB_2_xvYCC Conversion R_2_Cr Register (R2CR_COEFF_LOW) Field Descriptions ..... 2772


Copyright © 2011–2016, Texas Instruments Incorporated

13-99. RGB_2_xvYCC Conversion R_2_Cr Register (R2CR_COEFF_UP) Field Descriptions ..... 2772
13-100. RGB_2_xvYCC Conversion G_2_Cr Register (G2CR_COEFF_LOW) Field Descriptions ..... 2773
13-101. RGB_2_xvYCC Conversion G_2_Cr Register (G2CR_COEFF_UP) Field Descriptions ..... 2773
13-102. RGB_2_xvYCC Conversion B_2_Cr Register (B2CR_COEFF_LOW) Field Descriptions ..... 2774
13-103. RGB_2_xvYCC Conversion B_2_Cr Register (B2CR_COEFF_UP) Field Descriptions ..... 2774
13-104. RGB_2_xvYCC RGB Input Offset Register (RGB_OFFSET_LOW) Field Descriptions ..... 2775
13-105. RGB_2_xvYCC RGB Input Offset Register (RGB_OFFSET_UP) Field Descriptions ..... 2775
13-106. RGB_2_xvYCC Conversion Y Output Offset Register (Y_OFFSET_LOW) Field Descriptions ..... 2776
13-107. RGB_2_xvYCC Conversion Y Output Offset Register (Y_OFFSET_UP) Field Descriptions ..... 2776
13-108. RGB_2_xvYCC Conversion CbCr Output Offset Register (CBCR_OFFSET_LOW) Field Descriptions ..... 2777
13-109. RGB_2_xvYCC Conversion CbCr Output Offset Register (CBCR_OFFSET_UP) Field Descriptions ..... 2777
13-110. Interrupt State Register (INTR_STATE) Field Descriptions ..... 2777
13-111. Interrupt Source Register (INTR1) Field Descriptions ..... 2778
13-112. Interrupt Source Register (INTR2) Field Descriptions ..... 2779
13-113. Interrupt Source Register (INTR3) Field Descriptions ..... 2780
13-114. Interrupt Source Register (INTR4) Field Descriptions ..... 2781
13-115. Interrupt Unmask Register (INT_UNMASK1) Field Descriptions ..... 2782
13-116. Interrupt Unmask Register (INT_UNMASK2) Field Descriptions ..... 2783
13-117. Interrupt Unmask Register (INT_UNMASK3) Field Descriptions ..... 2784
13-118. Interrupt Unmask Register (INT_UNMASK4) Field Descriptions ..... 2785
13-119. Interrupt Control Register (INT_CTRL) Field Descriptions ..... 2786
13-120. xvYCC_2_RGB Control Register (XVYCC2RGB_CTL) Field Descriptions ..... 2787
13-121. xvYCC_2_RGB Conversion Y_2_R Register (Y2R_COEFF_LOW) Field Descriptions ..... 2788
13-122. xvYCC_2_RGB Conversion Y_2_R Register (Y2R_COEFF_UP) Field Descriptions ..... 2788
13-123. xvYCC_2_RGB Conversion Cr_2_R Register (CR2R_COEFF_LOW) Field Descriptions ..... 2789
13-124. xvYCC_2_RGB Conversion Cr_2_R Register (C2R2R_COEFF_UP) Field Descriptions ..... 2789
13-125. xvYCC_2_RGB Conversion Cb_2_B Register (CB2B_COEFF_LOW) Field Descriptions ..... 2790
13-126. xvYCC_2_RGB Conversion Cb_2_B Register (CB2B_COEFF_UP) Field Descriptions ..... 2790
13-127. CR2G_COEFF_LOW Field Descriptions ..... 2791
13-128. xvYCC_2_RGB Conversion Cr_2_G Register (CR2G_COEFF_UP) Field Descriptions ..... 2791
13-129. xvYCC_2_RGB Conversion Cb_2_G Register (CB2G_COEFF_LOW) ..... 2792
13-130. xvYCC_2_RGB Conversion Cb_2_G Register (CB2G_COEFF_UP) Field Descriptions ..... 2792
13-131. xvYCC_2_RGB Conversion Y Offset Register (YOFFSET1_LOW) Field Descriptions ..... 2793
13-132. xvYCC_2_RGB Conversion Y Offset Register (YOFFSET1_UP) Field Descriptions ..... 2793
13-133. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_LOW) Field Descriptions ..... 2794
13-134. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_MID) Field Descriptions ..... 2794
13-135. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_UP) Field Descriptions ..... 2794
13-136. xvYCC_2_RGB Conversion Offset2 Register (OFFSET2_LOW) Field Descriptions ..... 2795
13-137. xvYCC_2_RGB Conversion Offset2 Register (OFFSET2_UP) Field Descriptions ..... 2795
13-138. xvYCC_2_RGB Conversion DC Level Register (DCLEVEL_LOW) Field Descriptions ..... 2796
13-139. xvYCC_2_RGB Conversion DC Level Register (DCLEVEL_UP) Field Descriptions ..... 2796
13-140. DDC I2C Manual Register (DDC_MAN) Field Descriptions ..... 2797
13-141. DDC I2C Target Slave Address Register (DDC_ADDR) Field Descriptions ..... 2798
13-142. DDC I2C Target Segment Address Register (DDC_SEGM) Field Descriptions ..... 2798
13-143. DDC I2C Target Offset Address Register (DDC_OFFSET) Field Descriptions ..... 2798
13-144. DDC I2C Data Count Register (DDC_COUNT1) Field Descriptions ..... 2799
13-145. DDC I2C Data Count Register (DDC_COUNT2) Field Descriptions ..... 2799
13-146. DDC I2C Status Register (DDC_STATUS) Field Descriptions ..... 2800
13-147. DDC I2C Command Register (DDC_CMD) Field Descriptions ..... 2801


Copyright © 2011–2016, Texas Instruments Incorporated

13-148. DDC I2C Data Register (DDC_DATA) Field Descriptions ..... 2802
13-149. DDC I2C FIFO Count Register (DDC_FIFOCNT) Field Descriptions ..... 2802
13-150. ROM Status Register (EPST) Field Descriptions ..... 2803
13-151. ROM Command Register (EPCM) Field Descriptions ..... 2804
13-152. HDMI IP Core Gamut Registers ..... 2805
13-153. Gamut Metadata Register (GAMUT_HEADER1) Field Descriptions ..... 2805
13-154. Gamut Metadata Register (GAMUT_HEADER2) Field Descriptions ..... 2806
13-155. Gamut Metadata Register (GAMUT_HEADER3) Field Descriptions ..... 2807
13-156. Gamut Metadata Registers (GAMUT_DBYTE__0-GAMUT_DBYTE__27) Field Descriptions ..... 2807
13-157. HDMI IP Core Audio Video Registers ..... 2808
13-158. ACR Control Register (ACR_CTRL) Field Descriptions ..... 2809
13-159. ACR Audio Frequency Register (FREQ_SVAL) Field Descriptions ..... 2810
13-160. ACR N Software Value Register 1 (N_SVAL1) Field Descriptions ..... 2811
13-161. ACR N Software Value Register 2 (N_SVAL2) Field Descriptions ..... 2811
13-162. ACR N Software Value Register 3 (N_SVAL3) Field Descriptions ..... 2811
13-163. ACR CTS Software Value Register 1 (CTS_SVAL1) Field Descriptions ..... 2812
13-164. ACR CTS Software Value Register 2 (CTS_SVAL2) Field Descriptions ..... 2812
13-165. ACR CTS Software Value Register 3 (CTS_SVAL3) Field Descriptions ..... 2812
13-166. ACR CTS Hardware Value Register 1 (CTS_HVAL1) Field Descriptions ..... 2813
13-167. ACR CTS Hardware Value Register 2 (CTS_HVAL2) Field Descriptions ..... 2813
13-168. ACR CTS Hardware Value Register 3 (CTS_HVAL3) Field Descriptions ..... 2813
13-169. Audio In Mode Register (AUD_MODE) Field Descriptions ..... 2814
13-170. Audio In S/PDIF Control Register (SPDIF_CTRL) Field Descriptions ..... 2815
13-171. Audio In S/PDIF Extracted Fs and Length Register (HW_SPDIF_FS) Field Descriptions ..... 2816
13-172. Audio In I2S Channel Swap Register (SWAP_I2S) Field Descriptions ..... 2817
13-173. Audio Error Threshold Register (SPDIF_ERTH) Field Descriptions ..... 2818
13-174. Audio In I2S Data In Map Register (I2S_IN_MAP) Field Descriptions ..... 2819
13-175. Audio In I2S Control Register (I2S_IN_CTRL) Field Descriptions ..... 2820
13-176. I2S_CHST0 Field Descriptions ..... 2821
13-177. I2S_CHST1 Field Descriptions ..... 2821
13-178. Audio In I2S Channel Status Register 2 (I2S_CHST2) Field Descriptions ..... 2822
13-179. Audio In I2S Channel Status Register 3 (I2S_CHST3) Field Descriptions ..... 2822
13-180. Audio In I2S Channel Status Register 4 (I2S_CHST4) Field Descriptions ..... 2823
13-181. Audio In I2S Channel Status Register 5 (I2S_CHST5) Field Descriptions ..... 2824
13-182. Audio Sample Rate Conversion Register (ASRC) Field Descriptions ..... 2825
13-183. Audio I2S Input Length Register (I2S_IN_LEN) Field Descriptions ..... 2826
13-184. HDMI Control Register (HDMI_CTRL) Field Descriptions ..... 2827
13-185. Audio Path Status Register (AUDO_TXSTAT) Field Descriptions ..... 2828
13-186. Audio Input Data Rate Adjustment Register 1 (AUD_PAR_BUSCLK_1) Field Descriptions ..... 2829
13-187. AUD_PAR_BUSCLK_2 Field Descriptions ..... 2829
13-188. AUD_PAR_BUSCLK_3 Field Descriptions ..... 2829
13-189. Test Control Register (TEST_TXCTRL) Field Descriptions ..... 2830
13-190. Diagnostic Power Down Register (DPD) Field Descriptions ..... 2831
13-191. Packet Buffer Control 1 Register (PB_CTRL1) Field Descriptions ..... 2832
13-192. Packet Buffer Control 2 Register (PB_CTRL2) Field Descriptions ..... 2833
13-193. AVI InfoFrame Register (AVI_TYPE) Field Descriptions ..... 2834
13-194. AVI InfoFrame Register (AVI_VERS) Field Descriptions ..... 2834
13-195. AVI InfoFrame Register (AVI_LEN) Field Descriptions ..... 2834
13-196. AVI InfoFrame Register (AVI_CHSUM) Field Descriptions ..... 2835


Copyright © 2011–2016, Texas Instruments Incorporated

13-197. AVI InfoFrame Registers (AVI_DBYTE_0-AVI_DBYTE_14) Field Descriptions ..... 2835
13-198. SPD InfoFrame Register (SPD_TYPE) Field Descriptions ..... 2836
13-199. SPD InfoFrame Register (SPD_VERS) Field Descriptions ..... 2836
13-200. SPD InfoFrame Register (SPD_LEN) Field Descriptions ..... 2836
13-201. SPD InfoFrame Register (SPD_CHSUM) Field Descriptions ..... 2837
13-202. SPD InfoFrame Registers (SPD_DBYTE_0-SPD_DBYTE_26) Field Descriptions ..... 2837
13-203. Audio InfoFrame Register (AUDIO_TYPE) Field Descriptions ..... 2838
13-204. Audio InfoFrame Register (AUDIO_VERS) Field Descriptions ..... 2838
13-205. Audio InfoFrame Register (AUDIO_LEN) Field Descriptions ..... 2838
13-206. Audio InfoFrame Register (AUDIO_CHSUM) Field Descriptions ..... 2839
13-207. Audio InfoFrame Registers (AUDIO_DBYTE_0-AUDIO_DBYTE_9) Field Descriptions ..... 2839
13-208. MPEG InfoFrame Register (MPEG_TYPE) Field Descriptions ..... 2840
13-209. MPEG InfoFrame Register (MPEG_VERS) Field Descriptions ..... 2840
13-210. MPEG InfoFrame Register (MPEG_LEN) Field Descriptions ..... 2840
13-211. MPEG InfoFrame Register (MPEG_CHSUM) Field Descriptions ..... 2841
13-212. MPEG InfoFrame Registers (MPEG_DBYTE_0-MPEG_DBYTE_26) Field Descriptions ..... 2841
13-213. Generic Packet Registers (GEN_DBYTE_0-GEN_DBYTE_30) Field Descriptions ..... 2841
13-214. General Control Packet Register (CP_BYTE1) Field Descriptions ..... 2842
13-215. Generic Packet 2 Registers (GEN2_DBYTE_0-GEN2_DBYTE_30) Field Descriptions ..... 2842
13-216. CEC Slave ID Register (CEC_ADDR_ID) Field Descriptions ..... 2843
13-217. HDMI IP Core CEC Registers ..... 2844
13-218. CEC Device ID Register (CEC_DEV_ID) Field Descriptions ..... 2844
13-219. CEC Specification Register (CEC_SPEC) Field Descriptions ..... 2845
13-220. EC Specification Suffix Register (CEC_SUFF) Field Descriptions ..... 2845
13-221. CEC Firmware Revision Register (CEC_FW) Field Descriptions ..... 2846
13-222. CEC Debug Register 0 (CEC_DBG_0) Field Descriptions ..... 2846
13-223. CEC Debug Register 1 (CEC_DBG_1) Field Descriptions ..... 2846
13-224. CEC Debug Register 2 (CEC_DBG_2) Field Descriptions ..... 2847
13-225. CEC Debug Register 3 (CEC_DBG_3) Field Descriptions ..... 2848
13-226. CEC Tx Initialization Register (CEC_TX_INIT) Field Descriptions ..... 2849
13-227. CEC Tx Destination Register (CEC_TX_DEST) Field Descriptions ..... 2849
13-228. CEC Setup Register (CEC_SETUP) Field Descriptions ..... 2850
13-229. CEC Tx Command Register (CEC_TX_COMMAND) Field Descriptions ..... 2850
13-230. CEC Tx Operand Registers (CEC_TX_OPERAND_0-CEC_TX_OPERAND_14) Field Descriptions ..... 2851
13-231. CEC Transmit Data Register (CEC_TRANSMIT_DATA) Field Descriptions ..... 2851
13-232. CEC Capture ID0 Register (CEC_CA_7_0) Field Descriptions ..... 2852
13-233. CEC Capture ID0 Register (CEC_CA_15_8) Field Descriptions ..... 2852
13-234. CEC Interrupt Enable Register 0 (CEC_INIT_ENABLE_0) Field Descriptions ..... 2853
13-235. CEC Interrupt Enable Register 1 (CEC_INIT_ENABLE_1) Field Descriptions ..... 2854
13-236. CEC Interrupt Status Register 0 (CEC_INIT_STATUS_0) Field Descriptions ..... 2855
13-237. CEC_INIT_STATUS1 Field Descriptions ..... 2856
13-238. CEC RX Control Register (CEC_RX_CONTROL) Field Descriptions ..... 2857
13-239. CEC Rx Count Register (CEC_RX_COUNT) Field Descriptions ..... 2857
13-240. CEC Rx Command Header Register (CEC_RX_CMD_HEADER) Field Descriptions ..... 2858
13-241. CEC Rx Command Register (CEC_RX_COMMAND) Field Descriptions ..... 2858
13-242. CEC Rx Operand Registers (CEC_RX_OPERAND_0-CEC_RX_OPERAND_14) Field Descriptions ..... 2859
13-243. HDMI PHY Registers ..... 2860
13-244. TMDS Control Register 2 (TMDS_CNTL2) Field Descriptions ..... 2860
13-245. TMDS Control Register 3 (TMDS_CNTL3) Field Descriptions ..... 2861


Copyright © 2011–2016, Texas Instruments Incorporated

13-246. BIST Control Register (BIST_CNTL) Field Descriptions ..... 2862
13-247. TMDS Control Register 9 (TMDS_CNTL9) Field Descriptions ..... 2862
14-1. Interrupt Controller Resets ..... 2867
14-2. Interrupt Controller Interrupt Inputs and Outputs ..... 2867
14-3. Interrupt Controller (INTC) Registers ..... 2879
14-4. INTCPS_REVISION Register Field Descriptions ..... 2880
14-5. INTCPS_SYSCONFIG Register Field Descriptions ..... 2880
14-6. INTCPS_SYSSTATUS Register Field Descriptions ..... 2881
14-7. INTCPS_SIR_IRQ Register Field Descriptions ..... 2881
14-8. INTCPS_SIR_FIQ Register Field Descriptions ..... 2882
14-9. INTCPS_CONTROL Register Field Descriptions ..... 2882
14-10. INTCPS_PROTECTION Register Field Descriptions ..... 2883
14-11. INTCPS_IDLE Register Field Descriptions ..... 2883
14-12. INTCPS_IRQ_PRIORITY Register Field Descriptions ..... 2884
14-13. INTCPS_FIQ_PRIORITY Register Field Descriptions ..... 2884
14-14. INTCPS_THRESHOLD Register Field Descriptions ..... 2885
14-15. INTCPS_ITRn Register Field Descriptions ..... 2885
14-16. INTCPS_MIRn Register Field Descriptions ..... 2886
14-17. INTCPS_MIR_CLEARn Register Field Descriptions ..... 2886
14-18. INTCPS_MIR_SETn Register Field Descriptions ..... 2886
14-19. INTCPS_ISR_SETn Register Field Descriptions ..... 2887
14-20. INTCPS_ISR_CLEARn Register Field Descriptions ..... 2887
14-21. INTCPS_PENDING_IRQn Register Field Descriptions ..... 2887
14-22. INTCPS_PENDING_FIQn Register Field Descriptions ..... 2888
14-23. INTCPS_ILRm Register Field Descriptions ..... 2888
15-1. Signal Pads ..... 2892
15-2. Reset State of I2C Signals ..... 2892
15-3. I2C Registers ..... 2903
15-4. I2C_REVNB_LO Register (Module Revision) (LOW BYTES) Field Descriptions ..... 2904
15-5. I2C_REVNB_HI Register (HIGH BYTES) (Module Revision) Field Descriptions ..... 2905
15-6. I2C_SYSC Register (System Configuration) Field Descriptions ..... 2906
15-7. I2C_EOI Register (I2C End of Interrupt) Field Descriptions ..... 2907
15-8. I2C_IRQSTATUS_RAW Register (I2C Status Raw) Field Descriptions ..... 2908
15-9. I2C_IRQSTATUS Register (I2C Status) Field Descriptions ..... 2912
15-10. I2C_IRQENABLE_SET Register (I2C Interrupt Enable Set) Field Descriptions ..... 2914
15-11. I2C_IRQENABLE_CLR Register (I2C Interrupt Enable Clear) Field Descriptions ..... 2916
15-12. I2C_WE Register (I2C Wakeup Enable) Field Descriptions ..... 2918
15-13. I2C_DMARXENABLE_SET Register (Receive DMA Enable Set) Field Descriptions ..... 2921
15-14. I2C_DMATXENABLE_SET Register (Transmit DMA Enable Set) Field Descriptions ..... 2921
15-15. I2C_DMARXENABLE_CLR Register (Receive DMA Enable Clear) Field Descriptions ..... 2922
15-16. I2C_DMATXENABLE_CLR Register (Transmit DMA Enable Clear) Field Descriptions ..... 2922
15-17. I2C_DMARXWAKE_EN Register (Receive DMA Wakeup) Field Descriptions ..... 2923
15-18. I2C_DMATXWAKE_EN Register (Transmit DMA Wakeup) Field Descriptions ..... 2925
15-19. I2C_SYSS Register (System Status) Field Descriptions ..... 2927
15-20. I2C_BUF Register (Buffer Configuration) Field Descriptions ..... 2928
15-21. I2C_CNT Register (Data Counter) Field Descriptions ..... 2930
15-22. I2C_DATA Register (Data Access) Field Descriptions ..... 2931
15-23. I2C_CON Register (I2C Configuration) Field Descriptions ..... 2932
15-24. I2C_OA Register (I2C Own Address) Field Descriptions ..... 2934


Copyright © 2011–2016, Texas Instruments Incorporated

15-25. I2C_SA Register (I2C Slave Address) Field Descriptions ..... 2935
15-26. I2C_PSC Register (I2C Clock Prescaler) Field Descriptions ..... 2936
15-27. I2C_SCLL Register (I2C SCL Low Time) Field Descriptions ..... 2937
15-28. I2C_SCLH Register (I2C SCL High Time) Field Descriptions ..... 2937
15-29. I2C_SYSTEST Register (System Test) Field Descriptions ..... 2938
15-30. I2C_BUFSTAT Register (I2C Buffer Status) Field Descriptions ..... 2941
15-31. I2C_OA1 Register (OA1) (Own Address 1) Field Descriptions ..... 2942
15-32. I2C_OA2 Register (I2C Own Address 2) Field Descriptions ..... 2943
15-33. I2C_OA3 Register (I2C Own Address 3) Field Descriptions ..... 2944
15-34. I2C_ACTOA Register (Active Own Address) Field Descriptions ..... 2945
15-35. I2C_SBLOCK Register (I2C Clock Blocking Enable) Field Descriptions ..... 2946
16-1. Biphase-Mark Encoder ..... 2955
16-2. Preamble Codes ..... 2956
16-3. McASP Interface Signals ..... 2964
16-4. Channel Status and User Data for Each DIT Block ..... 2971
16-5. Transmit Bitstream Data Alignment ..... 2984
16-6. Receive Bitstream Data Alignment ..... 2986
16-7. McASP Registers Accessed Through Configuration Bus ..... 3003
16-8. McASP Registers Accessed Through Data Port ..... 3005
16-9. McASP AFIFO Registers Accessed Through Peripheral Configuration Port ..... 3005
16-10. Revision Identification Register (REV) Field Descriptions ..... 3006
16-11. Pin Function Register (PFUNC) Field Descriptions ..... 3007
16-12. Pin Direction Register (PDIR) Field Descriptions ..... 3009
16-13. Pin Data Output Register (PDOUT) Field Descriptions ..... 3011
16-14. Pin Data Input Register (PDIN) Field Descriptions ..... 3013
16-15. Pin Data Set Register (PDSET) Field Descriptions ..... 3015
16-16. Pin Data Clear Register (PDCLR) Field Descriptions ..... 3017
16-17. Global Control Register (GBLCTL) Field Descriptions ..... 3018
16-18. Audio Mute Control Register (AMUTE) Field Descriptions ..... 3020
16-19. Digital Loopback Control Register (DLBCTL) Field Descriptions ..... 3022
16-20. Digital Mode Control Register (DITCTL) Field Descriptions ..... 3023
16-21. Receiver Global Control Register (RGBLCTL) Field Descriptions ..... 3024
16-22. Receive Format Unit Bit Mask Register (RMASK) Field Descriptions ..... 3025
16-23. Receive Bit Stream Format Register (RFMT) Field Descriptions ..... 3026
16-24. Receive Frame Sync Control Register (AFSRCTL) Field Descriptions ..... 3028
16-25. Receive Clock Control Register (ACLKRCTL) Field Descriptions ..... 3029
16-26. Receive High-Frequency Clock Control Register (AHCLKRCTL) Field Descriptions ..... 3030
16-27. Receive TDM Time Slot Register (RTDM) Field Descriptions ..... 3031
16-28. Receiver Interrupt Control Register (RINTCTL) Field Descriptions ..... 3032
16-29. Receiver Status Register (RSTAT) Field Descriptions ..... 3033
16-30. Current Receive TDM Time Slot Registers (RSLOT) Field Descriptions ..... 3034
16-31. Receive Clock Check Control Register (RCLKCHK) Field Descriptions ..... 3035
16-32. Receiver DMA Event Control Register (REVTCTL) Field Descriptions ..... 3036
16-33. Transmitter Global Control Register (XGBLCTL) Field Descriptions ..... 3037
16-34. Transmit Format Unit Bit Mask Register (XMASK) Field Descriptions ..... 3038
16-35. Transmit Bit Stream Format Register (XFMT) Field Descriptions ..... 3039
16-36. Transmit Frame Sync Control Register (AFSXCTL) Field Descriptions ..... 3041
16-37. Transmit Clock Control Register (ACLKXCTL) Field Descriptions ..... 3042
16-38. Transmit High-Frequency Clock Control Register (AHCLKXCTL) Field Descriptions ..... 3043


Copyright © 2011–2016, Texas Instruments Incorporated

16-39. Transmit TDM Time Slot Register (XTDM) Field Descriptions ..... 3044
16-40. Transmitter Interrupt Control Register (XINTCTL) Field Descriptions ..... 3045
16-41. Transmitter Status Register (XSTAT) Field Descriptions ..... 3046
16-42. Current Transmit TDM Time Slot Register (XSLOT) Field Descriptions ..... 3047
16-43. Transmit Clock Check Control Register (XCLKCHK) Field Descriptions ..... 3048
16-44. Transmitter DMA Event Control Register (XEVTCTL) Field Descriptions ..... 3049
16-45. Serializer Control Registers (SRCTL*n*) Field Descriptions ..... 3050
16-46. Write FIFO Control Register (WFIFOCTL) Field Descriptions ..... 3053
16-47. Write FIFO Status Register (WFIFOSTS) Field Descriptions ..... 3054
16-48. Read FIFO Control Register (RFIFOCTL) Field Descriptions ..... 3055
16-49. Read FIFO Status Register (RFIFOSTS) Field Descriptions ..... 3056
17-1. Phases, Words and Bits Per Frame Control Bits ..... 3063
17-2. Effects of DLB and ALB Bits on Clock Modes ..... 3068
17-3. Channels, Block, Partitions ..... 3078
17-4. Eight Partitions – Receive Channel Assignment and Control ..... 3079
17-5. Eight Partitions – Transmit Channel Assignment and Control ..... 3079
17-6. Selecting a Transmit Multichannel Selection Mode with the XMCM Bit Field ..... 3082
17-7. McBSP Channel Control Options ..... 3082
17-8. Analysis of the Receiver Smart Idle Behavior ..... 3088
17-9. Input Clock Selection for Sample Rate Generator ..... 3090
17-10. How to Calculate the Length of the Receive Frame ..... 3094
17-11. Example: Use of RJUST Bit Field with 12-Bit Data Value ABCh ..... 3095
17-12. Example: Use of RJUST Bit Field with 20-Bit Data Value ABCDE ..... 3095
17-13. FSRM and GSYNC Effects on Frame-Synchronization Signal and McBSP.FSR Pin ..... 3096
17-14. CLKRM Effect on Receive Clock Signal and McBSP.CLKR Pin ..... 3098
17-15. How to Calculate the Length of the Transmit Frame ..... 3102
17-16. How FSXM and FSGM Select the Source of Transmit Frame-Synchronization Pulses ..... 3104
17-17. CLKXM Bit Effect on Transmit Clock and McBSP.CLKX Pin ..... 3106
17-18. Using McBSP Pins for General-Purpose I/O ..... 3109
17-19. McBSP Registers ..... 3110
17-20. Revision Number Register (REVNB) Field Descriptions ..... 3112
17-21. System Configuration Register (SYSCONFIG_REG) Field Descriptions ..... 3113
17-22. End of Interrupt Register (EOI) Field Descriptions ..... 3114
17-23. Interrupt Status Raw Register (IRQSTATUS_RAW) Field Descriptions ..... 3115
17-24. Interrupt Status Register (IRQSTATUS) Field Descriptions ..... 3117
17-25. Interrupt Enable Set Register (IRQENABLE_SET) Field Descriptions ..... 3119
17-26. Interrupt Enable Clear Register (IRQENABLE_CLR) Field Descriptions ..... 3121
17-27. DMA Rx Enable Set Register (DMARXENABLE_SET) Field Descriptions ..... 3123
17-28. DMA Tx Enable Set Register (DMATXENABLE_SET) Field Descriptions ..... 3123
17-29. DMA Rx Enable Clear Register (DMARXENABLE_CLR) Field Descriptions ..... 3124
17-30. DMA Tx Enable Clear Register (DMATXENABLE_CLR) Field Descriptions ..... 3124
17-31. DMA Rx Wake Enable Register (DMARXWAKE_EN) Field Descriptions ..... 3125
17-32. DMA Tx Wake Enable Register (DMATXWAKE_EN) Field Descriptions ..... 3127
17-33. McBSP_DRR_REG Field Descriptions ..... 3129
17-34. McBSP_DXR_REG Field Descriptions ..... 3129
17-35. McBSP_SPCR2_REG Field Descriptions ..... 3130
17-36. McBSP_SPCR1_REG Field Descriptions ..... 3131
17-37. McBSP_RCR2_REG Field Descriptions ..... 3132
17-38. McBSP_RCR1_REG Field Descriptions ..... 3133


Copyright © 2011–2016, Texas Instruments Incorporated

17-39. McBSP_XCR2_REG Field Descriptions ..... 3134
17-40. McBSP_XCR1_REG Field Descriptions ..... 3135
17-41. McBSP_SRGR2_REG Field Descriptions ..... 3136
17-42. McBSP_SRGR1_REG Field Descriptions ..... 3137
17-43. McBSP_MCR2_REG Field Descriptions ..... 3138
17-44. McBSP_MCR1_REG Field Descriptions ..... 3140
17-45. McBSP_RCERA_REG Field Descriptions ..... 3142
17-46. McBSP_RCERB_REG Field Descriptions ..... 3142
17-47. McBSP_XCERA_REG Field Descriptions ..... 3143
17-48. McBSP_XCERB_REG Field Descriptions ..... 3143
17-49. McBSP_PCR_REG Field Descriptions ..... 3144
17-50. McBSP_RCERC_REG Field Descriptions ..... 3146
17-51. McBSP_RCERD_REG Field Descriptions ..... 3146
17-52. McBSP_XCERC_REG Field Descriptions ..... 3147
17-53. McBSP_XCERD_REG Field Descriptions ..... 3147
17-54. McBSP_RCERE_REG Field Descriptions ..... 3148
17-55. McBSP_RCERF_REG Field Descriptions ..... 3148
17-56. McBSP_XCERE_REG Field Descriptions ..... 3149
17-57. McBSP_XCERF_REG Field Descriptions ..... 3149
17-58. McBSP_RCERG_REG Field Descriptions ..... 3150
17-59. McBSP_RCERH_REG Field Descriptions ..... 3150
17-60. McBSP_XCERG_REG Field Descriptions ..... 3151
17-61. McBSP_XCERH_REG Field Descriptions ..... 3151
17-62. McBSP_THRSH2_REG Field Descriptions ..... 3152
17-63. McBSP_THRSH1_REG Field Descriptions ..... 3152
17-64. McBSP_IRQSTATUS_REG Field Descriptions ..... 3153
17-65. McBSP_IRQENABLE_REG Field Descriptions ..... 3155
17-66. McBSP_WAKEUPEN_REG Field Descriptions ..... 3157
17-67. McBSP_XCCR_REG Field Descriptions ..... 3158
17-68. McBSP_RCCR_REG Field Descriptions ..... 3160
17-69. McBSP_XBUFFSTAT_REG Field Descriptions ..... 3161
17-70. McBSP_RBUFFSTAT_REG Field Descriptions ..... 3161
17-71. McBSP_STATUS_REG Field Descriptions ..... 3162
18-1. MMC/SD/SDIO Controller Pins and Descriptions ..... 3167
18-2. Response Type Summary ..... 3171
18-3. Local Power Management Features ..... 3176
18-4. Clock Activity Settings ..... 3177
18-5. Events ..... 3177
18-6. Memory Size, BLEN, and Buffer Relationship ..... 3185
18-7. MMC, SD, SDIO Responses in the MMCHS_RSP*n* Registers ..... 3186
18-8. CC and TC Values Upon Error Detected ..... 3187
18-9. MMC/SD/SDIO Controller Transfer Stop Command Summary ..... 3193
18-10. MMC/SD/SDIO Hardware Status Features ..... 3198
18-11. Global Init for Surrounding Modules ..... 3199
18-12. MMC/SD/SDIO Controller Wake-Up Configuration ..... 3200
18-13. MMC/SD/SDIO Registers ..... 3204
18-14. IP Revision Identifier Register (MMCHS_HL_REV) Field Descriptions ..... 3205
18-15. IP Module Hardware Configuration Register (MMCHS_HL_HWINFO) Field Descriptions ..... 3206
18-16. Clock Management Configuration (MMCHS_HL_SYSCONFIG Field Descriptions ..... 3207


Copyright © 2011–2016, Texas Instruments Incorporated

18-17. System Configuration Register (MMCHS_SYSCONFIG) Field Descriptions ..... 3208
18-18. System Status Register (MMCHS_SYSSTATUS) Field Descriptions ..... 3210
18-19. Card Status Response Error (MMCHS_CSRE) Field Descriptions ..... 3210
18-20. System Test Register (MMCHS_SYSTEST) Field Descriptions ..... 3211
18-21. Configuration Register (MMCHS_CON) Field Descriptions ..... 3215
18-22. Power Counter Register (MMCHS_PWCNT) Field Descriptions ..... 3219
18-23. Card Status Response Error (MMCHS_SDMASA) Field Descriptions ..... 3219
18-24. Transfer Length Configuration Register (MMCHS_BLK) Field Descriptions ..... 3220
18-25. Command Argument Register (MMCHS_ARG) Field Descriptions ..... 3221
18-26. Command and Transfer Mode Register (MMCHS_CMD) Field Descriptions ..... 3222
18-27. Command Response[31:0] Register (MMCHS_RSP10) Field Descriptions ..... 3225
18-28. Command Response[63:32] Register (MMCHS_RSP32) Field Descriptions ..... 3225
18-29. Command Response[95:64] Register (MMCHS_RSP54) Field Descriptions ..... 3226
18-30. Command Response[127:96] Register (MMCHS_RSP76) Field Descriptions ..... 3226
18-31. Data Register (MMCHS_DATA) Field Descriptions ..... 3227
18-32. Present State Register (MMCHS_PSTATE) Field Descriptions ..... 3228
18-33. Control Register (MMCHS_HCTL) Field Descriptions ..... 3230
18-34. SD System Control Register (MMCHS_SYSCTL) Field Descriptions ..... 3233
18-35. Interrupt Status Register (MMCHS_STAT) Field Descriptions ..... 3235
18-36. Interrupt SD Enable Register (MMCHS_IE) Field Descriptions ..... 3240
18-37. Interrupt Signal Enable Register (MMCHS_ISE) Field Descriptions ..... 3243
18-38. Auto CMD12 Error Status Register (MMCHS_AC12) Field Descriptions ..... 3246
18-39. Capabilities Register (MMCHS_CAPA) Field Descriptions ..... 3247
18-40. Force Event Register (MMCHS_FE) Field Descriptions ..... 3249
18-41. ADMA Error Status Register (MMCHS_ADMAES) Field Descriptions ..... 3251
18-42. ADMA System Address Low Bits (MMCHS_ADMASAL) Field Descriptions ..... 3252
18-43. ADMA System Address High Bits Register (MMCHS_ADMASAH) Field Descriptions ..... 3252
18-44. Versions Register (MMCHS_REV) Field Descriptions ..... 3253
19-1. Terminology Used ..... 3255
19-2. PCIe Transaction Layer Packets Supported ..... 3259
19-3. ..... 3263
19-4. Register Blocks That Make up the PCIe Configuration Registers ..... 3267
19-5. PCIe System Module Registers ..... 3269
19-6. PCISS Interrupt Events ..... 3270
19-7. ..... 3276
19-8. Supported Idle/Standby States ..... 3276
19-9. Compatibilities of Various States ..... 3277
19-10. LTSSM States ..... 3281
19-11. Applications Registers Summary ..... 3282
19-12. PID Register Field Descriptions ..... 3284
19-13. CMD_STATUS Register Field Descriptions ..... 3285
19-14. CFG_SETUP Register Field Descriptions ..... 3286
19-15. IOBASE Register Field Descriptions ..... 3286
19-16. TLPCFG Register Field Descriptions ..... 3287
19-17. RSTCMD Register Field Descriptions ..... 3287
19-18. PMCMD Register Field Descriptions ..... 3287
19-19. PMCFG Register Field Descriptions ..... 3288
19-20. ACT_STATUS Register Field Descriptions ..... 3288
19-21. OB_SIZE Register Field Descriptions ..... 3289


Copyright © 2011–2016, Texas Instruments Incorporated

19-22. DIAG_CTRL Register Field Descriptions ..... 3289
19-23. ENDIAN Register Field Descriptions ..... 3289
19-24. PRIORITY Register Field Descriptions ..... 3290
19-25. EOI Register Field Descriptions ..... 3290
19-26. MSI_IRQ Register Field Descriptions ..... 3291
19-27. EP_IRQ_SET Register Field Descriptions ..... 3291
19-28. EP_IRQ_CLR Register Field Descriptions ..... 3291
19-29. EP_IRQ_STATUS Register Field Descriptions ..... 3292
19-30. GPR0 Register Field Descriptions ..... 3292
19-31. GPR1 Register Field Descriptions ..... 3292
19-32. GPR2 Register Field Descriptions ..... 3293
19-33. GPR3 Register Field Descriptions ..... 3293
19-34. MSI0_STATUS_RAW Register Field Descriptions ..... 3293
19-35. MSI0_IRQ_STATUS Register Field Descriptions ..... 3293
19-36. MSI0_IRQ_ENABLE_SET Register Field Descriptions ..... 3294
19-37. MSI0_IRQ_ENABLE_CLR Register Field Descriptions ..... 3294
19-38. IRQ_STATUS_RAW Register Field Descriptions ..... 3294
19-39. IRQ_STATUS Register Field Descriptions ..... 3295
19-40. IRQ_ENABLE_SET Register Field Descriptions ..... 3295
19-41. IRQ_ENABLE_CLR Register Field Descriptions ..... 3296
19-42. ERR_IRQ_STATUS_RAW Register Field Descriptions ..... 3296
19-43. ERR_IRQ_STATUS Register Field Descriptions ..... 3297
19-44. ERR_IRQ_ENABLE_SET Register Field Descriptions ..... 3297
19-45. ERR_IRQ_ENABLE_CLR Register Field Descriptions ..... 3298
19-46. PMRST_IRQ_STATUS_RAW Register Field Descriptions ..... 3298
19-47. PMRST_IRQ_STATUS Register Field Descriptions ..... 3299
19-48. PMRST_ENABLE_SET Register Field Descriptions ..... 3299
19-49. PMRST_ENABLE_CLR Register Field Descriptions ..... 3300
19-50. OB_OFFSET_INDEXn Register Field Descriptions ..... 3300
19-51. OB_OFFSETn_HI Register Field Descriptions ..... 3301
19-52. IB_BAR0 Register Field Descriptions ..... 3301
19-53. IB_START0_LO Register Field Descriptions ..... 3301
19-54. IB_START0_HI Register Field Descriptions ..... 3302
19-55. IB_OFFSET0 Register Field Descriptions ..... 3302
19-56. IB_BAR1 Register Field Descriptions ..... 3302
19-57. IB_START1_LO Register Field Descriptions ..... 3303
19-58. IB_START1_HI Register Field Descriptions ..... 3303
19-59. IB_OFFSET 1 Register Field Descriptions ..... 3303
19-60. IB_BAR2 Register Field Descriptions ..... 3304
19-61. IB_START2_LO Register Field Descriptions ..... 3304
19-62. IB_START2_HI Register Field Descriptions ..... 3304
19-63. IB_OFFSET2 Register Field Descriptions ..... 3305
19-64. IB_BAR3 Register Field Descriptions ..... 3305
19-65. IB_START3_LO Register Field Descriptions ..... 3305
19-66. IB_START3_HI Register Field Descriptions ..... 3306
19-67. IB_OFFSET3 Register Field Descriptions ..... 3306
19-68. PCS_CFG0 Register Field Descriptions ..... 3307
19-69. PCS_CFG1 Register Field Descriptions ..... 3307
19-70. PCS_STATUS Register Field Descriptions ..... 3308


Copyright © 2011–2016, Texas Instruments Incorporated

19-71. SERDES_RXCFG0 Register Field Descriptions ..... 3308
19-72. SERDES_RXCFG1 Register Field Descriptions ..... 3311
19-73. SERDES_RXCFG2 Register Field Descriptions ..... 3312
19-74. SERDES_RXCFG3 Register Field Descriptions ..... 3313
19-75. SERDES_RXCFG4 Register Field Descriptions ..... 3314
19-76. SERDES_TXCFG0 Register Field Descriptions ..... 3317
19-77. SERDES_TXCFG1 Register Field Descriptions ..... 3319
19-78. SERDES_TXCFG2 Register Field Descriptions ..... 3321
19-79. SERDES_TXCFG3 Register Field Descriptions ..... 3322
19-80. SERDES_TXCFG4 Register Field Descriptions ..... 3323
19-81. Configuration Registers Common to Type 0 and Type 1 Headers ..... 3323
19-82. VENDOR_DEVICE_ID Register Field Descriptions ..... 3324
19-83. STATUS_COMMAND Register Field Descriptions ..... 3324
19-84. CLASSCODE_REVID Register Field Descriptions ..... 3325
19-85. Configuration Type 0 Register Summary ..... 3325
19-86. BIST_HEADER Register Field Descriptions ..... 3326
19-87. BAR0 Register Field Descriptions ..... 3327
19-88. BAR1 Register Field Descriptions ..... 3327
19-89. BAR1 Register Field Descriptions ..... 3328
19-90. BAR2 Register Field Descriptions ..... 3328
19-91. BAR3 Register Field Descriptions ..... 3329
19-92. BAR3 Register Field Descriptions ..... 3329
19-93. BAR4 Register Field Descriptions ..... 3330
19-94. BAR5 Register Field Descriptions ..... 3330
19-95. BAR5 Register Field Descriptions ..... 3331
19-96. CARDBUS Register Field Descriptions ..... 3331
19-97. SUBSYS_VNDR_ID Register Field Descriptions ..... 3331
19-98. EXPNSN_ROM Register Field Descriptions ..... 3332
19-99. CAP_PTR Register Field Descriptions ..... 3332
19-100. Reserved Register Field Descriptions ..... 3332
19-101. INT_PIN Register Field Descriptions ..... 3333
19-102. Configuration Type 1 Registers ..... 3333
19-103. BIST_HEADER Register Field Descriptions ..... 3334
19-104. BAR0 Register Field Descriptions ..... 3334
19-105. BAR1 Register Field Descriptions ..... 3335
19-106. BAR1 Register Field Descriptions ..... 3335
19-107. BUSNUM Register Field Descriptions ..... 3335
19-108. SECSTAT Register Field Descriptions ..... 3336
19-109. MEMSPACE Register Field Descriptions ..... 3337
19-110. PREFETCH_MEM Register Field Descriptions ..... 3337
19-111. PREFETCH_BASE Field Descriptions ..... 3337
19-112. PREFETCH_LIMIT Register Field Descriptions ..... 3338
19-113. IOSPACE Register Field Descriptions ..... 3338
19-114. CAP_PTR Register Field Descriptions ..... 3338
19-115. EXPNSN_ROM Register Field Descriptions ..... 3339
19-116. BRIDGE_INT Register Field Descriptions ..... 3339
19-117. Power Management Capability Registers ..... 3340
19-118. PMCAP Register Field Descriptions ..... 3340
19-119. PM_CTL-STAT Register Field Descriptions ..... 3341


Copyright © 2011–2016, Texas Instruments Incorporated

19-120. Message Signaled Interrupts Registers ..... 3341
19-121. MSI_CAP Register Field Descriptions ..... 3342
19-122. MSI_LOW32 Register Field Descriptions ..... 3342
19-123. MSI_UP32 Register Field Descriptions ..... 3343
19-124. MSI_DATA Register Field Descriptions ..... 3343
19-125. PCIExpress Capabilities Registers ..... 3343
19-126. PCIES_CAP Register Field Descriptions ..... 3344
19-127. DEVICE_CAP Register Field Descriptions ..... 3344
19-128. DEV_STAT_CTRL Register Field Descriptions ..... 3345
19-129. LINK_CAP Register Field Descriptions ..... 3346
19-130. LINK_STAT_CTRL Register Field Descriptions ..... 3347
19-131. SLOT_CAP Register Field Descriptions ..... 3348
19-132. SLOT_STAT_CTRL Register Field Descriptions ..... 3349
19-133. ROOT_CTRL_CAP Register Field Descriptions ..... 3349
19-134. ROOT_STATUS Register Field Descriptions ..... 3350
19-135. DEV_CAP2 Register Field Descriptions ..... 3350
19-136. DEV_STAT_CTRL2 Register Field Descriptions ..... 3351
19-137. LINK_CTRL2 Register Field Descriptions ..... 3351
19-138. PCI Express Extended Capabilities Registers ..... 3352
19-139. PCIE_EXTCAP Register Field Descriptions ..... 3352
19-140. PCIE_UNCERR Register Field Descriptions ..... 3353
19-141. PCIE_UNCERR_MASK Register Field Descriptions ..... 3354
19-142. PCIE_UNCERR_SVRTY Register Field Descriptions ..... 3354
19-143. PCIE_CERR Register Field Descriptions ..... 3355
19-144. PCIE_CERR_MASK Register Field Descriptions ..... 3356
19-145. PCIE_ACCR Register Field Descriptions ..... 3356
19-146. HDR_LOG0 Register Field Descriptions ..... 3356
19-147. HDR_LOG1 Register Field Descriptions ..... 3357
19-148. HDR_LOG2 Register Field Descriptions ..... 3357
19-149. HDR_LOG3 Register Field Descriptions ..... 3357
19-150. RC_ERR_CMD Register Field Descriptions ..... 3358
19-151. RC_ERR_ST Register Field Descriptions ..... 3358
19-152. ERR_SRC_ID Register Field Descriptions ..... 3359
19-153. Port Logic Registers ..... 3359
19-154. PL_ACKTIMER Register Field Descriptions ..... 3359
19-155. PL_OMSG Register Field Descriptions ..... 3360
19-156. PL_FORCE_LINK Register Field Descriptions ..... 3360
19-157. ACK_FREQ Register Field Descriptions ..... 3361
19-158. PL_LINK_CTRL Register Field Descriptions ..... 3361
19-159. LANE_SKEW Register Field Descriptions ..... 3362
19-160. SYM_NUM Register Field Descriptions ..... 3362
19-161. SYMTIMER_FLTMASK Register Field Descriptions ..... 3363
19-162. FLT_MASK2 Register Field Descriptions ..... 3364
19-163. DEBUG0 Register Field Descriptions ..... 3364
19-164. DEBUG1 Register Field Descriptions ..... 3365
19-165. PL_GEN2 Register Field Descriptions ..... 3366
20-1. RTC Signals ..... 3369
20-2. Interrupt Trigger Events ..... 3370
20-3. RTC Register Names and Values ..... 3372


Copyright © 2011–2016, Texas Instruments Incorporated

20-4. Real-Time Clock (RTC) Registers ..... 3376
20-5. Seconds Register (SECONDS_REG) Field Descriptions ..... 3377
20-6. Minutes Register (MINUTES_REG) Field Descriptions ..... 3377
20-7. Hours Register (HOURS_REG) Field Descriptions ..... 3378
20-8. Day of the Month (DAYS_REG) Field Descriptions ..... 3378
20-9. Month Register (MONTHS_REG) Field Descriptions ..... 3379
20-10. Year Register (YEARS_REG) Field Descriptions ..... 3379
20-11. Day of the Week (WEEKS_REG) Field Descriptions ..... 3380
20-12. Alarm Second Register (ALARM_SECONDS_REG) Field Descriptions ..... 3380
20-13. Alarm Minute Register (ALARM_MINUTES_REG) Field Descriptions ..... 3381
20-14. Alarm Hour Register (ALARM_HOURS_REG) Field Descriptions ..... 3381
20-15. Alarm Day of the Month Register (ALARM_DAYS_REG) Field Descriptions ..... 3382
20-16. Alarm Month Register (ALARM_MONTHS_REG) Field Descriptions ..... 3382
20-17. Alarm Year Register (ALARM_YEARS_REG) Field Descriptions ..... 3383
20-18. Control Register (CTRL_REG) Field Descriptions ..... 3384
20-19. Status Register (STATUS_REG) Field Descriptions ..... 3386
20-20. Interrupt Register (INTERRUPTS_REG) Field Descriptions ..... 3387
20-21. Compensation (LSB) Register (COMP_LSB_REG) Field Descriptions ..... 3388
20-22. Compensation (MSB) Register (COMP_MSB_REG) Field Descriptions ..... 3389
20-23. Oscillator Register (OSC_REG) Field Descriptions ..... 3390
20-24. Scratch Registers (SCRATCHx_REG) Field Descriptions ..... 3390
20-25. Kick0 Register (KICK0R) Field Descriptions ..... 3391
20-26. Kick1 Register (KICK1R) Field Descriptions ..... 3391
20-27. RTC Revision Register (RTC_REVISION) Field Descriptions ..... 3392
20-28. System Configuration Register (RTC_SYSCONFIG) Field Descriptions ..... 3392
20-29. Wakeup Enable Register (RTC_IRQWAKEEN) Field Descriptions ..... 3393
21-1. SATA Interface Signal Descriptions ..... 3400
21-2. SATA Controller Registers ..... 3424
21-3. HBA Capabilities Register (CAP) Field Descriptions ..... 3426
21-4. Global HBA Control Register (GHC) Field Descriptions ..... 3427
21-5. Interrupt Status Register (IS) Field Descriptions ..... 3428
21-6. Ports Implemented Register (PI) Field Descriptions ..... 3429
21-7. AHCI Version Register (VS) Field Descriptions ..... 3429
21-8. Command Completion Coalescing Control Register (CCC_CTL) Field Descriptions ..... 3430
21-9. Command Completion Coalescing Ports Register (CCC_PORTS) Field Description ..... 3431
21-10. BIST Active FIS Register (BISTAFR) Field Descriptions ..... 3432
21-11. BIST Control Register (BISTCR) Field Descriptions ..... 3433
21-12. BIST FIS Count Register (BISTFCTR) Field Description ..... 3435
21-13. BIST Status Register (BISTSR) Field Description ..... 3435
21-14. BIST DWORD Error Count Register (BISTDECR) Field Description ..... 3436
21-15. BIST DWORD Error Count Register (TIMER1MS) Field Description ..... 3436
21-16. Global Parameter 1 Register (GPARAM1R) Field Descriptions ..... 3437
21-17. Global Parameter 2 Register (GPARAM2R) Field Descriptions ..... 3438
21-18. Port Parameter Register (PPARAMR) Field Descriptions ..... 3439
21-19. Version Register (VERSIONR) Field Description ..... 3440
21-20. ID Register (IDR) Field Description ..... 3440
21-21. Port Command List Base Address Register (P0CLB) Field Description ..... 3441
21-22. Port FIS Base Address Register (P0FB) Field Description ..... 3441
21-23. Port Interrupt Status Register (P0IS) Field Descriptions ..... 3442


Copyright © 2011–2016, Texas Instruments Incorporated

21-24. Port Interrupt Enable Register (P0IE) Field Descriptions ..... 3444
21-25. Port Command Register (P0CMD) Field Descriptions ..... 3445
21-26. Port Task File Data Register (P0TFD) Field Descriptions ..... 3447
21-27. Port Signature Register (P0SIG) Field Description ..... 3448
21-28. Port Serial ATA Status Register (P0SSTS) Field Descriptions ..... 3448
21-29. Port Serial ATA Control Register (P0SCTL) Field Descriptions ..... 3450
21-30. Port Serial ATA Error Register (P0SERR) Field Descriptions ..... 3451
21-31. Port Serial ATA Active Register (P0SACT) Field Description ..... 3453
21-32. Port Command Issue Register (P0CI) Field Description ..... 3453
21-33. Port Serial ATA Notification Register (P0SNTF) Field Description ..... 3454
21-34. Port DMA Control Register (P0DMACR) Field Description ..... 3455
21-35. Idle Register (IDLE) Field Description ..... 3457
21-36. PHY Configuration Receive 0 Register Field Descriptions ..... 3458
21-37. PHY Configuration Receive 1 Register (CFGRX1) Field Descriptions ..... 3460
21-38. PHY Configuration Receive 2 Register (CFGRX2) Field Descriptions ..... 3461
21-39. PHY Configuration Receive 3 Register (CFGRX3) Field Descriptions ..... 3462
21-40. PHY Configuration Receive 4 Register (CFGRX4) Field Descriptions ..... 3464
21-41. Receive Bus PHY-to-Controller Status Register Field Descriptions Field Descriptions ..... 3465
21-42. PHY Configuration Transmit 0 Register (CFGTX0) Field Descriptions ..... 3466
21-43. PHY Configuration Transmit 1 Register (CFGTX1) Field Descriptions ..... 3469
21-44. PHY Configuration Transmit 2 Register (CFGTX2) Field Descriptions ..... 3471
21-45. PHY Configuration Transmit 2 Register (CFGTX3) Field Descriptions ..... 3473
21-46. PHY Configuration Transmit 2 Register (CFGTX4) Field Descriptions ..... 3474
21-47. Transmit Bus Controller-to-PHY Status Register Field Descriptions ..... 3474
21-48. SATA PLL Controller Registers ..... 3475
21-49. SATA PLL Configuration 0 Register (SATA_PLLCFG0) Field Description ..... 3475
21-50. SATA PLL Configuration 1 Register (SATA_PLLCFG1) Field Description ..... 3477
21-51. SATA PLL Configuration 2 Register (SATA_PLLCFG2) Field Description ..... 3478
21-52. SATA PLL Configuration 3 Register (SATA_PLLCFG3) Field Description ..... 3479
21-53. SATA PLL Configuration 4 Register (SATA_PLLCFG4) Field Description ..... 3480
21-54. SATA PLL Status (SATA_PLLSTATUS) Field Description ..... 3482
21-55. SATA Receive Status (SATA_RXSTATUS) Field Description ..... 3482
21-56. SATA Transmit Status (SATA_TXSTATUS) Field Description ..... 3483
21-57. SATA TEST Configuration (SATA_TESTCFG) Field Description ..... 3483
21-58. SATA Subsystem Controller Registers ..... 3484
21-59. Clock Domain Power Control Register (CM_DEFAULT_L3_MED_CLKSTCTRL) Field Descriptions ..... 3484
21-60. Clock Domain SATA Control Register (CM_DEFAULT_SATA_CLKCTRL) Field Descriptions ..... 3485
22-1. SPI Interface Pins ..... 3488
22-2. Phase and Polarity Combinations ..... 3492
22-3. Chip Select ↔ Clock Edge Delay Depending on Configuration ..... 3502
22-4. CLKSPIO High/Low Time Computation ..... 3503
22-5. Clock Granularity Examples ..... 3503
22-6. FIFO Writes, Word Length Relationship ..... 3504
22-7. SPI Receive Mode Initialization ..... 3522
22-8. SPI Transmit Mode Initialization ..... 3522
22-9. SPI Transmit-and-Receive Mode Initialization ..... 3522
22-10. Receive-Only Procedure – Polling Method ..... 3523
22-11. Receive-Only Procedure – Interrupt Method ..... 3523
22-12. Transmit-Only Procedure – Polling Method ..... 3523


Copyright © 2011–2016, Texas Instruments Incorporated

22-13. Transmit-and-Receive Procedure – Polling Method ..... 3524
22-14. Receive-Only Procedure With Word Count – Polling Method ..... 3524
22-15. Transmit-Only Procedure Without Word Count – Polling Method ..... 3524
22-16. Transmit-Only Procedure With Word Count – Interrupt Method ..... 3524
22-17. Transmit-and-Receive Procedure With Word Count – Polling Method ..... 3525
22-18. Transmit-and-Receive Procedure With Word Count – Interrupt Method ..... 3525
22-19. Transmit-and-Receive Procedure Without Word Count – Polling Method ..... 3525
22-20. SPI Registers ..... 3526
22-21. McSPI IP Revision Register (MCSPI_HL_REV) Field Descriptions ..... 3527
22-22. McSPI IP Hardware Information Register (MCSPI_HL_HWINFO) Field Descriptions ..... 3528
22-23. McSPI IP System Configuration Register (MCSPI_HL_SYSCONFIG) Field Descriptions ..... 3529
22-24. McSPI Revision Register (MCSPI_REVISION) Field Descriptions ..... 3530
22-25. McSPI System Configuration Register (MCSPI_SYSCONFIG) Field Descriptions ..... 3531
22-26. McSPI System Status Register (MCSPI_SYSSTATUS) Field Descriptions ..... 3532
22-27. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions ..... 3533
22-28. McSPI Interrupt Enable Register (MCSPI_IRQENABLE) Field Descriptions ..... 3536
22-29. McSPI Wakeup Enable Register (MCSPI_WAKEUPENABLE) Field Descriptions ..... 3538
22-30. McSPI System Test Register (MCSPI_SYST) Field Descriptions ..... 3539
22-31. McSPI Module Control Register(MCSPI_MODULCTRL) Field Descriptions ..... 3541
22-32. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) Field Descriptions ..... 3543
22-33. Data Lines Configurations ..... 3546
22-34. McSPI Channel *(i)* Status Register (MCSPI_CH(i)STAT) Field Descriptions ..... 3547
22-35. McSPI Channel *(i)* Control Register (MCSPI_CH(I)CTRL) Field Descriptions ..... 3548
22-36. McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*) Field Descriptions ..... 3549
22-37. McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*) Field Descriptions ..... 3549
22-38. McSPI Transfer Levels Register (MCSPI_XFERLEVEL) Field Descriptions ..... 3550
22-39. DMA Address Aligned FIFO Transmitter Register (MCSPI_DAFTX) Field Descriptions ..... 3551
22-40. DMA Address Aligned FIFO Receiver Register (MCSPI_DAFRX) Field Descriptions ..... 3551
23-1. Timer Resolution and Maximum Range ..... 3554
23-2. Prescaler Functionality ..... 3558
23-3. Prescaler Clock Ratios Value ..... 3561
23-4. Value and Corresponding Interrupt Period ..... 3561
23-5. OCP Error Reporting ..... 3562
23-6. Timer Registers ..... 3565
23-7. TIDR Register Field Descriptions ..... 3566
23-8. TIOCP_CFG Register Field Descriptions ..... 3567
23-9. IRQ_EOI Register Field Descriptions ..... 3568
23-10. IRQSTATUS_RAW Register Field Descriptions ..... 3569
23-11. IRQSTATUS Register Field Descriptions ..... 3570
23-12. IRQENABLE_SET Register Field Descriptions ..... 3571
23-13. IRQENABLE_CLR Register Field Descriptions ..... 3572
23-14. IRQWAKEEN Register Field Descriptions ..... 3573
23-15. TCLR Register Field Descriptions ..... 3574
23-16. TCRR Register Field Descriptions ..... 3575
23-17. TLDR Register Field Descriptions ..... 3575
23-18. TTGR Register Field Descriptions ..... 3576
23-19. TWPS Register Field Descriptions ..... 3576
23-20. TMAR Register Field Descriptions ..... 3577
23-21. TCAR1 Register Field Descriptions ..... 3577


Copyright © 2011–2016, Texas Instruments Incorporated

23-22. TSICR Register Field Descriptions ..... 3578
23-23. TCAR2 Register Field Descriptions ..... 3578
24-1. UART Signal Description ..... 3582
24-2. UART Mode Selection ..... 3582
24-3. FIR Transmit Frame Format ..... 3596
24-4. UART Mode Interrupts ..... 3608
24-5. IrDA Mode Interrupts ..... 3609
24-6. CIR Mode Interrupts ..... 3609
24-7. UART Baud Rate Settings (48-MHz Clock) ..... 3612
24-8. IrDA Baud Rates Settings ..... 3612
24-9. UART Registers ..... 3614
24-10. Receiver Holding Register (RHR) Field Descriptions ..... 3615
24-11. Transmit Holding Register (THR) Field Descriptions ..... 3615
24-12. UART Interrupt Enable Register (IER) Field Descriptions ..... 3616
24-13. IrDA Interrupt Enable Register (IER) Field Descriptions ..... 3617
24-14. CIR Interrupt Enable Register (IER) Field Descriptions ..... 3618
24-15. UART Interrupt Identification Register (IIR) Field Descriptions ..... 3619
24-16. IrDA Interrupt Identification Register (IIR) Field Descriptions ..... 3620
24-17. CIR Interrupt Identification Register (IIR) Field Descriptions ..... 3621
24-18. FIFO Control Register (FCR) Field Descriptions ..... 3622
24-19. Line Control Register (LCR) Field Descriptions ..... 3623
24-20. Modem Control Register (MCR) Field Descriptions ..... 3624
24-21. UART Line Status Register (LSR) Field Descriptions ..... 3625
24-22. IrDA Line Status Register (LSR) Field Descriptions ..... 3626
24-23. CIR Line Status Register (LSR) Field Descriptions ..... 3627
24-24. Modem Status Register (MSR) Field Descriptions ..... 3628
24-25. Transmission Control Register (TCR) Field Descriptions ..... 3629
24-26. Scratchpad Register (SPR) Field Descriptions ..... 3629
24-27. Trigger Level Register (TLR) Field Descriptions ..... 3630
24-28. RX FIFO Trigger Level Setting Summary ..... 3630
24-29. TX FIFO Trigger Level Setting Summary ..... 3630
24-30. Mode Definition Register 1 (MDR1) Field Descriptions ..... 3631
24-31. Mode Definition Register 2 (MDR2) Field Descriptions ..... 3632
24-32. Mode Definition Register 3 (MDR3) Field Descriptions ..... 3633
24-33. Status FIFO Line Status Register (SFLSR) Field Descriptions ..... 3634
24-34. RESUME Register Field Descriptions ..... 3634
24-35. Status FIFO Register Low (SFREGL) Field Descriptions ..... 3635
24-36. Status FIFO Register High (SFREGH) Field Descriptions ..... 3635
24-37. BOF Control Register (BLR) Field Descriptions ..... 3636
24-38. Auxiliary Control Register (ACREG) Field Descriptions ..... 3637
24-39. Supplementary Control Register (SCR) Field Descriptions ..... 3638
24-40. Supplementary Status Register (SSR) Field Descriptions ..... 3639
24-41. BOF Length Register (EBLR) Field Descriptions ..... 3640
24-42. Module Version Register (MVR) Field Descriptions ..... 3641
24-43. System Configuration Register (SYSC) Field Descriptions ..... 3642
24-44. System Status Register (SYSS) Field Descriptions ..... 3642
24-45. Wake-Up Enable Register (WER) Field Descriptions ..... 3643
24-46. Carrier Frequency Prescaler Register (CFPS) Field Descriptions ..... 3644
24-47. Divisor Latches Low Register (DLL) Field Descriptions ..... 3645


Copyright © 2011–2016, Texas Instruments Incorporated

24-48. Divisor Latches High Register (DLH) Field Descriptions ..... 3645
24-49. Enhanced Feature Register (EFR) Field Descriptions ..... 3646
24-50. EFR[3:0] Software Flow Control Options ..... 3647
24-51. XON1/ADDR1 Register Field Descriptions ..... 3647
24-52. XON2/ADDR2 Register Field Descriptions ..... 3647
24-53. XOFF1 Register Field Descriptions ..... 3648
24-54. XOFF2 Register Field Descriptions ..... 3648
24-55. Transmit Frame Length Low Register (TXFLL) Field Descriptions ..... 3649
24-56. Transmit Frame Length High Register (TXFLH) Field Descriptions ..... 3649
24-57. Received Frame Length Low Register (RXFLL) Field Descriptions ..... 3650
24-58. Received Frame Length High Register (RXFLH) Field Descriptions ..... 3650
24-59. UART Autobauding Status Register (UASR) Field Descriptions ..... 3651
25-1. 53 Bytes Test Packet Content ..... 3660
25-2. USBSS Interface Signals ..... 3661
25-3. Subsystem Interrupts ..... 3662
25-4. Controller Interrupts ..... 3664
25-5. PERI_TXCSR Register Bit Configuration for Bulk IN Transactions ..... 3675
25-6. PERI_RXCSR Register Bit Configuration for Bulk OUT Transactions ..... 3676
25-7. PERI_TXCSR Register Bit Configuration for Isochronous IN Transactions ..... 3679
25-8. PERI_RXCSR Register Bit Configuration for Isochronous OUT Transactions ..... 3679
25-9. Isochronous OUT Error Handling: Peripheral Mode ..... 3682
25-10. Packet Descriptor Word 0 (PD0) Bit Field Descriptions ..... 3702
25-11. Packet Descriptor Word 1 (PD1) Bit Field Descriptions ..... 3702
25-12. Packet Descriptor Word 2 (PD2) Bit Field Descriptions ..... 3702
25-13. Packet Descriptor Word 3 (PD3) Bit Field Descriptions ..... 3703
25-14. Packet Descriptor Word 4 (PD4) Bit Field Descriptions ..... 3703
25-15. Packet Descriptor Word 5 (PD5) Bit Field Descriptions ..... 3703
25-16. Packet Descriptor Word 6 (PD6) Bit Field Descriptions ..... 3704
25-17. Packet Descriptor Word 7 (PD7) Bit Field Descriptions ..... 3704
25-18. Buffer Descriptor Word 0 (BD0) Bit Field Descriptions ..... 3705
25-19. Buffer Descriptor Word 1 (BD1) Bit Field Descriptions ..... 3705
25-20. Buffer Descriptor Word 2 (BD2) Bit Field Descriptions ..... 3705
25-21. Buffer Descriptor Word 3 (BD3) Bit Field Descriptions ..... 3706
25-22. Buffer Descriptor Word 4 (BD4) Bit Field Descriptions ..... 3706
25-23. Buffer Descriptor Word 5 (BD5) Bit Field Descriptions ..... 3706
25-24. Buffer Descriptor Word 6 (BD6) Bit Field Descriptions ..... 3706
25-25. Buffer Descriptor Word 7 (BD7) Bit Field Descriptions ..... 3707
25-26. Teardown Descriptor Word 0 Bit Field Descriptions ..... 3708
25-27. Teardown Descriptor Words 1 to 7 Bit Field Descriptions ..... 3708
25-28. Queue-Endpoint Mapping ..... 3711
25-29. USBSS Submodule Base Addresses and Size ..... 3726
25-30. USBSS Registers ..... 3726
25-31. USBSS Revision Register (REVREG) Field Description ..... 3727
25-32. USBSS SYSCONFIG Register (SYSCONFIG) Field Descriptions ..... 3728
25-33. USBSS End of Interrupt Register (EOI) Field Descriptions ..... 3730
25-34. USBSS IRQ_STATUS_RAW (IRQSTATRAW) Field Descriptions ..... 3731
25-35. USBSS IRQ_STATUS (IRQSTAT) Field Descriptions ..... 3732
25-36. USBSS IRQ_ENABLE_SET Register (IRQENABLER) Field Descriptions ..... 3733
25-37. USBSS IRQ_ENABLE_CLR Register (IRQCLEARR) Field Descriptions ..... 3734


Copyright © 2011–2016, Texas Instruments Incorporated

25-38. USBSS IRQ_DMA_THRESHOLD_TX0_0 Register (IRQDMATHOLDTX00) Field Descriptions ..... 3735
25-39. USBSS IRQ_DMA_THRESHOLD_TX0_1 Register (IRQDMATHOLDTX01) Field Descriptions..... 3735
25-40. USBSS IRQ_DMA_THRESHOLD_TX0_2 Register (IRQDMATHOLDTX02) Field Descriptions..... 3736
25-41. USBSS IRQ_DMA_THRESHOLD_TX0_3 Register (IRQDMATHOLDTX03) Field Descriptions..... 3736
25-42. USBSS IRQ_DMA_THRESHOLD_RX0_0 Register (IRQDMATHOLDRX00) Field Descriptions ..... 3737
25-43. USBSS IRQ_DMA_THRESHOLD_RX0_1 Register (IRQDMATHOLDRX00) Field Descriptions ..... 3737
25-44. USBSS IRQ_DMA_THRESHOLD_RX0_2 Register (IRQDMATHOLDRX02) Field Descriptions ..... 3738
25-45. USBSS IRQ_DMA_THRESHOLD_RX0_3 Register (IRQDMATHOLDRX03) Field Descriptions ..... 3738
25-46. USBSS IRQ_DMA_THRESHOLD_TX1_0 Register (IRQDMATHOLDTX10) Field Descriptions ..... 3739
25-47. USBSS IRQ_DMA_THRESHOLD_TX1_1 Register (IRQDMATHOLDTX11) Field Descriptions ..... 3739
25-48. USBSS IRQ_DMA_THRESHOLD_TX1_2 Register (IRQDMATHOLDTX12) Field Descriptions..... 3740
25-49. USBSS IRQ_DMA_THRESHOLD_TX1_3 Register (IRQDMATHOLDTX13) Field Descriptions..... 3740
25-50. USBSS IRQ_DMA_THRESHOLD_RX1_0 Register (IRQDMATHOLDRX10) Field Descriptions ..... 3741
25-51. USBSS IRQ_DMA_THRESHOLD_RX1_1 Register (IRQDMATHOLDRX11) Field Descriptions ..... 3741
25-52. USBSS IRQ_DMA_THRESHOLD_RX1_2 Register (IRQDMATHOLDRX12) Field Descriptions ..... 3742
25-53. USBSS IRQ_DMA_THRESHOLD_RX1_3 Register (IRQDMATHOLDRX13) Field Descriptions ..... 3742
25-54. USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0) Field Descriptions ..... 3743
25-55. USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1) Field Descriptions..... 3745
25-56. USBSS IRQ_FRAME_THRESHOLD_TX0_0 Register (IRQFRAMETHOLDTX00) Field Descriptions ..... 3747
25-57. USBSS IRQ_FRAME_THRESHOLD_TX0_1 Register (IRQFRAMETHOLDTX01) Field Descriptions ..... 3747
25-58. USBSS IRQ_FRAME_THRESHOLD_TX0_2 Register (IRQFRAMETHOLDTX02) Field Descriptions..... 3748
25-59. USBSS IRQ_FRAME_THRESHOLD_TX0_3 Register (IRQFRAMETHOLDTX03) Field Descriptions..... 3748
25-60. USBSS IRQ_FRAME_THRESHOLD_RX0_0 Register (IRQFRAMETHOLDRX00) Field Descriptions ..... 3749
25-61. USBSS IRQ_FRAME_THRESHOLD_RX0_1 Register (IRQFRAMETHOLDRX01) Field Descriptions ..... 3749
25-62. USBSS IRQ_FRAME_THRESHOLD_RX0_2 Register (IRQFRAMETHOLDRX02) Field Descriptions ..... 3750
25-63. USBSS IRQ_FRAME_THRESHOLD_RX0_3 Register (IRQFRAMETHOLDRX03) Field Descriptions ..... 3750
25-64. USBSS IRQ_FRAME_THRESHOLD_TX1_0 Register (IRQFRAMETHOLDTX10) Field Descriptions ..... 3751
25-65. USBSS IRQ_FRAME_THRESHOLD_TX1_1 Register (IRQFRAMETHOLDTX11) Field Descriptions..... 3751
25-66. USBSS IRQ_FRAME_THRESHOLD_TX1_2 Register (IRQFRAMETHOLDTX12) Field Descriptions..... 3752
25-67. USBSS IRQ_FRAME_THRESHOLD_TX1_3 Register (IRQFRAMETHOLDTX13) Field Descriptions..... 3752
25-68. USBSS IRQ_FRAME_THRESHOLD_RX1_0 Register (IRQFRAMETHOLDRX10) Field Descriptions ..... 3753
25-69. USBSS IRQ_FRAME_THRESHOLD_RX1_1 Register (IRQFRAMETHOLDRX11) Field Descriptions ..... 3753
25-70. USBSS IRQ_FRAME_THRESHOLD_RX1_2 Register (IRQFRAMETHOLDRX12) Field Descriptions ..... 3754
25-71. USBSS IRQ_FRAME_THRESHOLD_RX1_3 Register (IRQFRAMETHOLDRX13) Field Descriptions ..... 3754
25-72. USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0) Field Descriptions ..... 3755
25-73. USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1) Field Descriptions ..... 3757
25-74. USB0 Controller Registers ..... 3759
25-75. USB0 Revision Register (USB0REV) Field Descriptions ..... 3760
25-76. USB0 Control Register (USB0CTRL) Field Descriptions ..... 3761
25-77. USB0 Status Register (USB0STAT) Field Descriptions..... 3762
25-78. USB0 IRQ_MERGED_STATUS Register (USB0IRQMSTAT) Field Descriptions..... 3762
25-79. USB0 IRQ_EOI Register (USB0IRQEOI) Field Descriptions ..... 3763
25-80. USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0) Field Descriptions ..... 3764
25-81. USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1) Field Descriptions..... 3766
25-82. USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0) Field Descriptions..... 3768
25-83. USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1) Field Descriptions..... 3770
25-84. USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0) Field Descriptions ..... 3772
25-85. USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1) Field Descriptions ..... 3774
25-86. USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0) Field Descriptions ..... 3776


Copyright © 2011–2016, Texas Instruments Incorporated

25-87. USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1) Field Descriptions ..... 3778
25-88. USB0 Tx Mode Register (USB0TXMODE) Field Descriptions ..... 3780
25-89. USB0 Rx Mode Register (USB0RXMODE) Field Descriptions ..... 3782
25-90. USB0 Generic RNDIS EPn Size Register (USB0GENRNDISEPn) Field Descriptions ..... 3784
25-91. USB0 Auto Req Register (USB0AUTOREQ) Field Descriptions ..... 3785
25-92. USB0 SRP Fix Time Register (USB0SRPFIXTIME) Field Descriptions ..... 3788
25-93. USB0 Teardown Register (USB0TDOWN) Field Descriptions ..... 3788
25-94. USB0 Threshold XDMA Idle Register Field Descriptions ..... 3789
25-95. USB0 PHY UTMI Register (USB0UTMI) Field Descriptions ..... 3790
25-96. USB0 MGC UTMI Loopback Register (USB0UTMILB) Field Descriptions ..... 3791
25-97. USB0 Mode Register (USB0MODE) Field Descriptions ..... 3792
25-98. USB1 Controller Registers ..... 3793
25-99. USB1 Revision Register (USB1REV) Field Descriptions ..... 3794
25-100. USB1 Control Register (USB1CTRL) Field Descriptions ..... 3795
25-101. USB1 Status Register (USB1STAT) Field Descriptions ..... 3796
25-102. USB1 IRQ_MERGED_STATUS Register (USB1IRQMSTAT) Field Descriptions ..... 3796
25-103. USB1 IRQ_EOI Register (USB1IRQEOI) Field Descriptions ..... 3797
25-104. USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW0) Field Descriptions ..... 3798
25-105. USB1 IRQ_STATUS_RAW_1 Register (USB1IRQSTATRAW1) Field Descriptions ..... 3800
25-106. USB1 IRQ_STATUS_0 Register (USBIRQSTAT0) Field Descriptions ..... 3802
25-107. USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1) Field Descriptions ..... 3804
25-108. USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0) Field Descriptions ..... 3806
25-109. USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1) Field Descriptions ..... 3808
25-110. USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0) Field Descriptions ..... 3810
25-111. USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1) Field Descriptions ..... 3812
25-112. USB1 Tx Mode Register (USB1TXMODE) Field Descriptions ..... 3814
25-113. USB1 Rx Mode Register (USB1RXMODE) Field Descriptions ..... 3816
25-114. USB1 Auto Req Register (USB1AUTOREQ) Field Descriptions ..... 3819
25-115. USB1 SRP Fix Time Register (USB1SRPFIXTIME) Field Descriptions ..... 3822
25-116. USB1 Teardown Register (USB1TDOWN) Field Descriptions ..... 3822
25-117. USB1 Threshold XDMA Idle Register Field Descriptions ..... 3823
25-118. USB1 PHY UTMI Register (USB1UTMI) Field Descriptions ..... 3824
25-119. USB1 MGC UTMI Loopback Register (USB1UTMILB) Field Descriptions ..... 3825
25-120. USB1 Mode Register (USB1MODE) Field Descriptions ..... 3826
25-121. CPPI DMA Controller Registers ..... 3827
25-122. CPPI DMA Revision Registe (DMAREVID) Field Descriptions ..... 3828
25-123. CPPI DMA Teardown Free Descriptor Queue Control Register (TDFDQ) Field Descriptions ..... 3828
25-124. CPPI DMA Emulation Control Register (DMAEMU) Field Descriptions ..... 3829
25-125. Tx Channel N Global Configuration Register (TXGCRn) Field Descriptions ..... 3829
25-126. Rx Channel N Global Configuration Register (RXGCRn) Field Descriptions ..... 3830
25-127. Rx Channel N Host Packet Configuration Register A (RXHPCRAn) Field Descriptions ..... 3832
25-128. Rx Channel N Host Packet Configuration Register B (RXHPCRBn) Field Descriptions ..... 3833
25-129. CPPI DMA Scheduler Registers ..... 3834
25-130. CPPI DMA Scheduler Control Register (DMA_SCHED_CTRL) Field Descriptions ..... 3834
25-131. CPPI DMA Scheduler Table Word N (WORDn) Field Descriptions ..... 3835
25-132. CPPI DMA Queue Manager Registers ..... 3837
25-133. Queue Manager Revision Register (QMGRREVID) Field Descriptions ..... 3838
25-134. Queue Manager Queue Diversion Register (DIVERSION) Field Descriptions ..... 3838
25-135. Queue Manager Free Descriptor/Buffer Starvation Count Register 0 (FDBSC0) Field Descriptions ..... 3839


Copyright © 2011–2016, Texas Instruments Incorporated

25-136. Queue Manager Free Descriptor/Buffer Starvation Count Register 1 (FDBSC1) Field Descriptions ..... 3839
25-137. Queue Manager Free Descriptor/Buffer Starvation Count Register 2 (FDBSC2) Field Descriptions ..... 3840
25-138. Queue Manager Free Descriptor/Buffer Starvation Count Register 3 (FDBSC3) Field Descriptions ..... 3840
25-139. Queue Manager Free Descriptor/Buffer Starvation Count Register 4 (FDBSC4) Field Descriptions ..... 3841
25-140. Queue Manager Free Descriptor/Buffer Starvation Count Register 5 (FDBSC5) Field Descriptions ..... 3841
25-141. Queue Manager Free Descriptor/Buffer Starvation Count Register 6 (FDBSC6) Field Descriptions ..... 3842
25-142. Queue Manager Free Descriptor/Buffer Starvation Count Register 7 (FDBSC7) Field Descriptions ..... 3842
25-143. Queue Manager Linking RAM Region 0 Base Address Register (LRAM0BASE) Field Descriptions ..... 3843
25-144. Queue Manager Linking RAM Region 0 Size Register (LRAM0SIZE) Field Descriptions ..... 3843
25-145. Queue Manager Linking RAM Region 1 Base Address Register (LRAM1BASE) Field Descriptions ..... 3844
25-146. Queue Manager Queue Pending Register 0 (PEND0) Field Descriptions ..... 3844
25-147. Queue Manager Queue Pending Register 1 (PEND1) Field Descriptions ..... 3845
25-148. Queue Manager Queue Pending Register 2 (PEND1) Field Descriptions ..... 3845
25-149. Queue Manager Queue Pending Register 3 (PEND3) Field Descriptions ..... 3845
25-150. Queue Manager Queue Pending Register 4 (PEND4) Field Descriptions ..... 3846
25-151. Queue Manager Memory Region R Base Address Register (QMEMRBASEr) Fi`eld Descriptions ..... 3846
25-152. Queue Manager Memory Region R Control Register (QMEMRCTRLr) Field Descriptions ..... 3847
25-153. Queue Manager Queue N Register A (CTRLAn) Field Descriptions ..... 3848
25-154. Queue Manager Queue N Register B (CTRLBn) Field Descriptions ..... 3848
25-155. Queue Manager Queue N Register C (CTRLCn) Field Descriptions ..... 3849
25-156. Queue Manager Queue N Register D (CTRLDn) Field Descriptions ..... 3849
25-157. Queue Manager Queue N Status Register A (QSTATAn) Field Descriptions ..... 3850
25-158. Queue Manager Queue N Status Register B (QSTATBn) Field Descriptions ..... 3850
25-159. Queue Manager Queue N Status Register C (QSTATCn) Field Descriptions ..... 3851
25-160. Common USB Registers ..... 3852
25-161. Function Address Register (USBn_FADDR) Field Descriptions ..... 3853
25-162. Power Management Register (USBn_POWER) Field Descriptions ..... 3853
25-163. Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 (USBn_INTRTX) Field Descriptions ..... 3854
25-164. Interrupt Register for Receive Endpoints 1 to 15 (INTRRX) Field Descriptions ..... 3855
25-165. Interrupt Enable Register for INTRTX (INTRTXE) Field Descriptions ..... 3856
25-166. Interrupt Enable Register for INTRRX (INTRRXE) Field Descriptions ..... 3857
25-167. Interrupt Register for Common USB Interrupts (USBn_INTRUSB) Field Descriptions ..... 3858
25-168. Interrupt Enable Register for INTRUSB (USBn_INTRUSBE) Field Descriptions ..... 3858
25-169. Frame Number Register (USBn_FRAME) Field Descriptions ..... 3859
25-170. Index Register for Selecting the Endpoint Status and Control Registers (USBn_INDEX) Field Descriptions ..... 3859
25-171. Register to Enable the USB 2.0 Test Modes (USBn_TESTMODE) Field Descriptions ..... 3860
25-172. Indexed Region Registers ..... 3861
25-173. Maximum Packet Size for Peripheral/Host Transmit Endpoint (USBn_TXMAXP) Field Descriptions ..... 3861
25-174. Control Status Register for Endpoint 0 in Peripheral Mode (USBn_PERI_CSR0) Field Descriptions ..... 3862
25-175. Control Status Register for Endpoint 0 in Host Mode (USBn_HOST_CSR0) Field Descriptions ..... 3863
25-176. Control Status Register for Peripheral Transmit Endpoint (USBn_PERI_TXCSR) Field Descriptions ..... 3864
25-177. Control Status Register for Host Transmit Endpoint (USBn_HOST_TXCSR) Field Descriptions ..... 3865
25-178. Maximum Packet Size for Peripheral Host Receive Endpoint (USBn_RXMAXP) Field Descriptions ..... 3866
25-179. Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR) Field Descriptions ..... 3867
25-180. Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR) Field Descriptions ..... 3869
25-181. Count 0 Register (USBn_COUNT0) Field Descriptions ..... 3870
25-182. Receive Count Register (USBn_RXCOUNT) Field Descriptions ..... 3871
25-183. Type Register (Host Mode Only) (USBn_HOST_TYPE0) Field Descriptions ..... 3871


Copyright © 2011–2016, Texas Instruments Incorporated

25-184. Transmit Type Register (Host Mode Only) (USBn_HOST_TXTYPE) Field Descriptions ..... 3872
25-185. NAKLimit0 Register (Host Mode Only) (USBn_HOST_NAKLIMIT0) Field Descriptions ..... 3873
25-186. Transmit Interval Register (Host Mode Only) (USBn_HOST_TXINTERVAL) Field Descriptions ..... 3874
25-187. Receive Type Register (Host Mode Only) (USBn_HOST_RXTYPE) Field Descriptions ..... 3875
25-188. Transmit Interval Register (Host Mode Only) (USBn_HOST_RXINTERVAL) Field Descriptions ..... 3876
25-189. Configuration Data Register (USBn_CONFIGDATA) Field Descriptions ..... 3877
25-190. FIFOs: Transmit and Receive FIFO Register for Endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15) Field Descriptions ..... 3878
25-191. Additional Control and Configuration Registers ..... 3879
25-192. Device Control Register (USBn_DEVCTL) Field Descriptions ..... 3879
25-193. Transmit Endpoint FIFO Size Register (USBn_TXFIFOSZ) Field Descriptions ..... 3880
25-194. Receive Endpoint FIFO Size Register (USBn_RXFIFOSZ) Field Descriptions ..... 3881
25-195. Transmit Endpoint FIFO Address Register (USBn_TXFIFOADDR) Field Descriptions ..... 3881
25-196. Receive Endpoint FIFO Address Register (USBn_RXFIFOADDR) Field Descriptions ..... 3882
25-197. Hardware Version Register (USBn_HWVERS) Field Descriptions ..... 3882
25-198. Target Endpoint Control Registers ..... 3883
25-199. Transmit Function Address (USBn_TXFUNCADDRm) Field Descriptions ..... 3883
25-200. Transmit Hub Address Register (USBn_TXHUBADDRm) Field Descriptions ..... 3884
25-201. Transmit Hub Port Register (USBn_TXHUBPORTm) Field Descriptions ..... 3884
25-202. Receive Function Address Register (USBn_RXFUNCADDRm) Field Descriptions ..... 3885
25-203. Receive Hub Address Register (USBn_RXHUBADDRm) Field Descriptions ..... 3885
25-204. Receive Hub Port Register (USBn_RXHUBPORTm) Field Descriptions ..... 3886
25-205. Peripheral Mode Configuration and Status Register Mapping where n=0,1 (USB0 or USB1) and m=1-15 (endpoint number) ..... 3887
25-206. Host Mode Configuration and Status Register Mapping where n=0,1 (USB0 or USB1) and m=1-15 (endpoint number) ..... 3887
26-1. Watchdog Timer Events ..... 3890
26-2. Count and Prescaler Default Reset Values ..... 3891
26-3. Prescaler Clock Ratio Values ..... 3892
26-4. Reset Period Examples ..... 3892
26-5. Default Watchdog Timer Reset Periods ..... 3893
26-6. Global Initialization of Surrounding Modules ..... 3896
26-7. Watchdog Timer Module Global Initialization ..... 3896
26-8. Watchdog Timer Basic Configuration ..... 3896
26-9. Disable the Watchdog Timer ..... 3897
26-10. Enable the Watchdog Timer ..... 3897
26-11. Watchdog Timer Registers ..... 3898
26-12. WDT_WIDR Register Field Descriptions ..... 3899
26-13. WDT_WDSC Register Field Descriptions ..... 3899
26-14. WDT_WDST Register Field Descriptions ..... 3900
26-15. WDT_WISR Register Field Descriptions ..... 3900
26-16. WDT_WIER Register Field Descriptions ..... 3901
26-17. WDT_WCLR Register Field Descriptions ..... 3901
26-18. WDT_WCRR Register Field Descriptions ..... 3902
26-19. WDT_WLDR Register Field Descriptions ..... 3902
26-20. WDT_WTGR Register Field Descriptions ..... 3903
26-21. WDT_WWPS Register Field Descriptions ..... 3903
26-22. WDT_WDLY Register Field Descriptions ..... 3904
26-23. WDT_WSPR Register Field Descriptions ..... 3904
26-24. WDT_WIRQSTATRAW Register Field Descriptions ..... 3905


Copyright © 2011–2016, Texas Instruments Incorporated

26-25. WDT_WIRQSTAT Register Field Descriptions ..... 3906
26-26. WDT_WIRQENSET Register Field Descriptions ..... 3907
26-27. WDT_WIRQENCLR Register Field Descriptions ..... 3908


Copyright © 2011–2016, Texas Instruments Incorporated

# Read This First

## About This Manual

This Technical Reference Manual (TRM) details the integration, the environment, the functional description, and the programming models for each peripheral and subsystem in the device.

The TRM should not be considered a substitute for the data manual, rather a companion guide that should be used alongside the device-specific data manual to understand the details to program the device. The primary purpose of the TRM is to abstract the programming details of the device from the data manual. This allows the data manual to outline the high-level features of the device without unnecessary information about register descriptions or programming models.

## Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers may be shown with the suffix h or the prefix 0x. For example, the following number is 40 hexadecimal (decimal 64): 40h or 0x40
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties with default reset value below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure can have one of multiple meanings:
    - Not implemented on the device
    - Reserved for future device expansion
    - Reserved for TI testing
    - Reserved configurations of the device that are not supported
  - Writing non-default values to the Reserved bits could cause unexpected behavior and should be avoided.

## Glossary

SLYZ022 — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

## Related Documentation From Texas Instruments

For product information, visit the Texas Instruments website at http://www.ti.com.

**SPRS712 —** TMS320DM8127 DaVinci™ Video Processor

**SPRS647 —** TMS320DM814x DaVinci™ Video Processors

## Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

**TI E2E™ Online Community— *TI's Engineer-to-Engineer (E2E) Community.*** Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.


Copyright © 2011–2016, Texas Instruments Incorporated

**TI Embedded Processors Wiki— *Texas Instruments Embedded Processors Wiki.*** Established to help developers get started with Embedded Processors from Texas Instruments and to foster innovation and growth of general knowledge about the hardware and software surrounding these devices.

## Trademarks

DaVinci, E2E, C674x+ are trademarks of Texas Instruments. TrustZone is a registered trademark of ARM Ltd.. Cortex is a trademark of ARM, Ltd.. Arteris is a registered trademark of Arteris, Inc.. USSE is a trademark of Imagination Technologies Ltd.. POWERVR is a registered trademark of Imagination Technologies Ltd.. Sonics3220, SonicsMX are trademarks of Sonics, Inc..


Copyright © 2011–2016, Texas Instruments Incorporated

# Chip Level Resources

This chapter provides information about the chip level resources.

**Topic** **Page**


1.1 ARM® Cortex™-A8 Microprocessor Unit (MPU) Subsystem Overview ..... 156
1.2 Media Controller Subsystem ..... 158
1.3 C674x DSP ..... 180
1.4 HD Video Coprocessor Subsystem ..... 183
1.5 System MMU ..... 188
1.6 SGX530 Graphics Subsystem ..... 218
1.7 Device Interrupts and EDMA Events ..... 240
1.8 Inter-Processor Communication ..... 255
1.9 Mailbox ..... 260
1.10 Spinlock ..... 290
1.11 Error Location Module ..... 300
1.12 Bus Interconnect ..... 324



Copyright © 2011–2016, Texas Instruments Incorporated

## 1.1 ARM® Cortex™-A8 Microprocessor Unit (MPU) Subsystem Overview

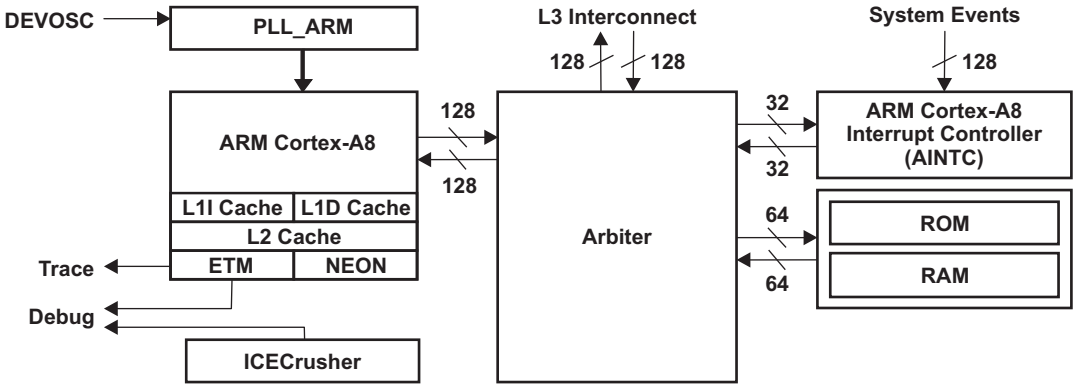
The ARM® Cortex™-A8 Subsystem is designed to allow the ARM Cortex-A8 master control of the device. In general, the ARM Cortex-A8 is responsible for configuration and control of the various subsystems, peripherals, and external memories.

The ARM Cortex-A8 Subsystem includes the following features:

- ARM Cortex-A8 RISC processor:
  - ARMv7 ISA plus Thumb2™, JazelleX™, and Media Extensions
  - Neon™ Floating-Point Unit
  - Enhanced Memory Management Unit (MMU)
  - Little Endian
  - 32K-Byte L1 Instruction Cache
  - 32K-Byte L1 Data Cache
  - 256K-Byte L2 Cache with ECC
- CoreSight Embedded Trace Module (ETM)
- ARM Cortex-A8 Interrupt Controller (AINTC)
- Embedded PLL Controller (PLL_ARM)
- Internal RAM
- Internal ROM

Figure 1-1 shows the ARM Cortex-A8 Subsystem for the device.

**Figure 1-1. ARM Cortex-A8 Subsystem**




### 1.1.1 ARM Cortex-A8 RISC Processor

The ARM Cortex-A8 processor is a member of ARM Cortex family of general-purpose microprocessors. This processor is targeted at multi-tasking applications where full memory management, high performance, low die size, and low power are all important. The ARM Cortex-A8 processor supports the ARM debug architecture and includes logic to assist in both hardware and software debug. The ARM Cortex-A8 processor has a Harvard architecture and provides a complete high-performance subsystem, including:

- ARM Cortex-A8 Integer Core
- Superscalar ARMv7 Instruction Set
- Thumb-2 Instruction Set
- Jazelle RCT Acceleration
- CP14 Debug Coprocessor
- CP15 System Control Coprocessor
- NEON™ 64-/128-bit Hybrid SIMD Engine for Multimedia
- Enhanced VFPv3 Floating-Point Coprocessor


Copyright © 2011–2016, Texas Instruments Incorporated

- Enhanced Memory Management Unit (MMU)
- Separate Level-1 Instruction and Data Caches
- Integrated Level-2 Cache
- 128-bit Interconnect with Level 3 Fast (L3) System Memories and Peripherals
- Embedded Trace Module (ETM).

### 1.1.2 Embedded Trace Module (ETM)

To support real-time trace, the ARM Cortex-A8 processor provides an interface to enable connection of an embedded trace module (ETM). The ETM consists of two parts:

- The Trace port which provides real-time trace capability for the ARM Cortex-A8.
- Triggering facilities that provide trigger resources, which include address and data comparators, counter, and sequencers.

The ARM Cortex-A8 trace port is not pinned out and is, instead, only connected to the system-level Embedded Trace Buffer (ETB). The ETB has a 32KB buffer memory. ETB enabled debug tools are required to read/interpret the captured trace data.

### 1.1.3 ARM Cortex-A8 Interrupt Controller (AINTC)

The ARM Cortex-A8 subsystem contains an interrupt controller (AINTC) that prioritizes all service requests from the system peripherals and generates either IRQ or FIQ to the ARM Cortex-A8 processor.

### 1.1.4 ARM Cortex-A8 PLL (PLL_ARM)

The ARM Cortex-A8 subsystem contains an embedded PLL Controller (PLL_ARM) for generating the subsystem's clocks from the device Clock input.

### 1.1.5 ARM MPU Interconnect

The ARM Cortex-A8 processor is connected through the arbiter to the L3 interconnect port. The L3 interconnect port is 128-bits wide and provides access to the other device modules.


Copyright © 2011–2016, Texas Instruments Incorporated

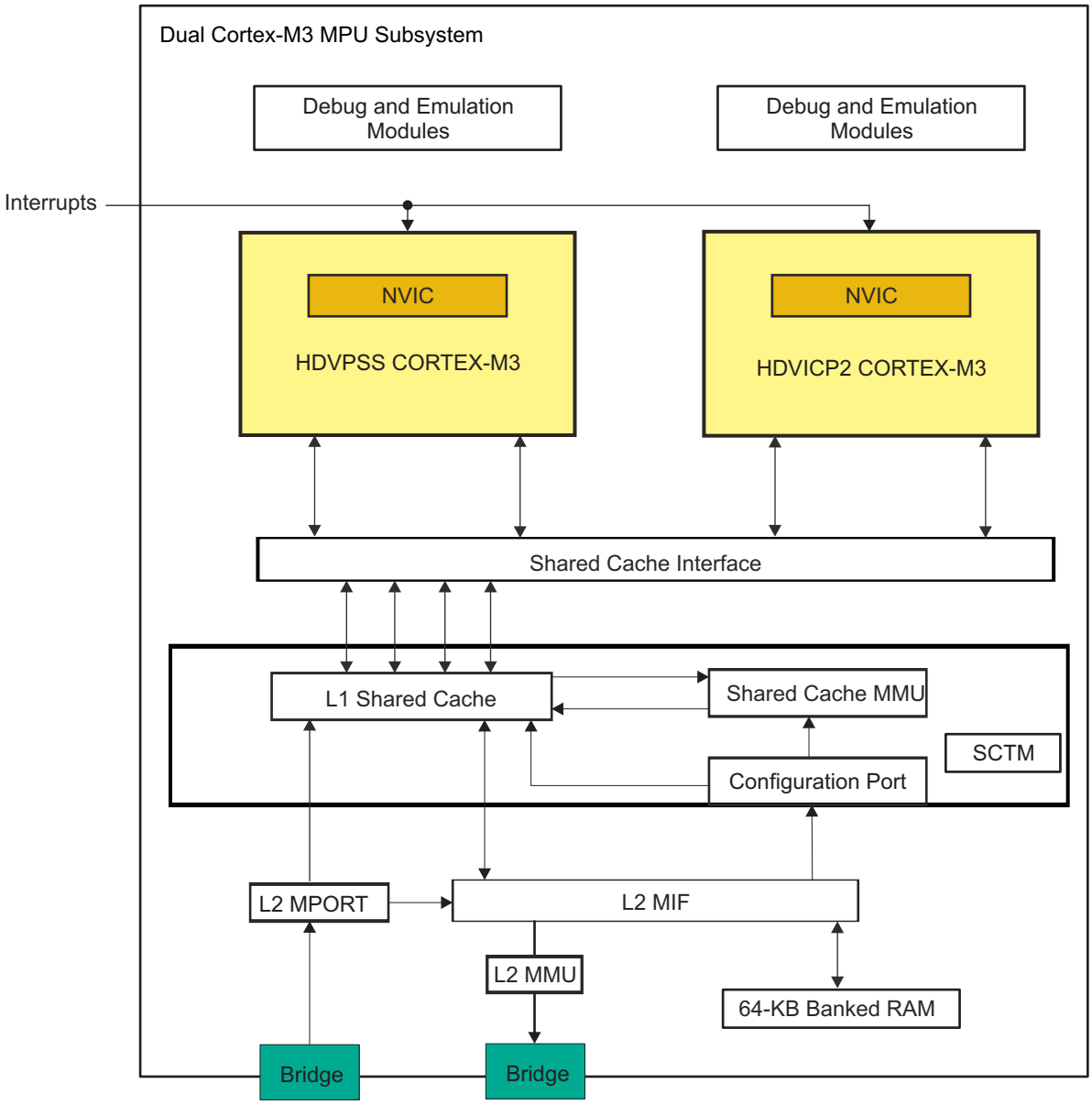
## 1.2 Media Controller Subsystem

### 1.2.1 Dual Cortex-M3 MPU Subsystem Overview

#### 1.2.1.1 Introduction

The Dual Cortex™-M3 MPU subsystem controls and manages the HDVPSS and HDVICP2 subsystems. It contains two ARM® Cortex-M3 processors (CPUs) that share a common L1 cache - Shared cache. Figure 1-2 is a high-level block diagram of the dual Cortex-M3 MPU subsystem.

**Figure 1-2. Dual Cortex-M3 MPU Subsystem Overview**




#### 1.2.1.2 Features

The dual Cortex-M3 MPU subsystem integrates the following:

- Two ARM Cortex-M3 microprocessor:
  - ARMv7-M and Thumb®-2 instruction set architecture


Copyright © 2011–2016, Texas Instruments Incorporated

  - Hardware division and single-cycle multiplication computational acceleration
  - Integrated nested vector interrupt controller (NVIC)
  - Integrated bus matrix
  - Registers:
    - Thirteen general-purpose 32-bit registers
    - Link register (LR)
    - Program counter (PC)
    - Program status register, xPSR
    - Two banked SP registers
  - Integrated power management
  - Extensive debug capabilities
- Shared cache interface:
  - Instruction and data interface
  - Supports paralleled accesses
- L2 MIF (Master Interface) – splitter for access to memory or configuration port
- Configuration port - used for shared cache maintenance and shared cache MMU configuration
- Shared cache
  - 32KB divided into 16 banks
  - 4-way
  - Cache configuration – lock/freeze/preload
  - Internal MMU (memory management unit):
    - 16-entry region-based address translation
    - Read/write control and access type control
    - Execute Never (XN) MMU protection policy
    - Little-endian format
- SCTM - Counter Timer Module
- Banked RAM memory
- Emulation/debug: Emulation feature embedded in Cortex-M3
- L2 MMU: 32 entries with walking table logic
- Power management

### 1.2.2 Dual Cortex-M3 MPU Subsystem Integration

Figure 1-3 shows the signals that interface with other modules.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-3. Dual Cortex-M3 MPU Subsystem Integration Overview**

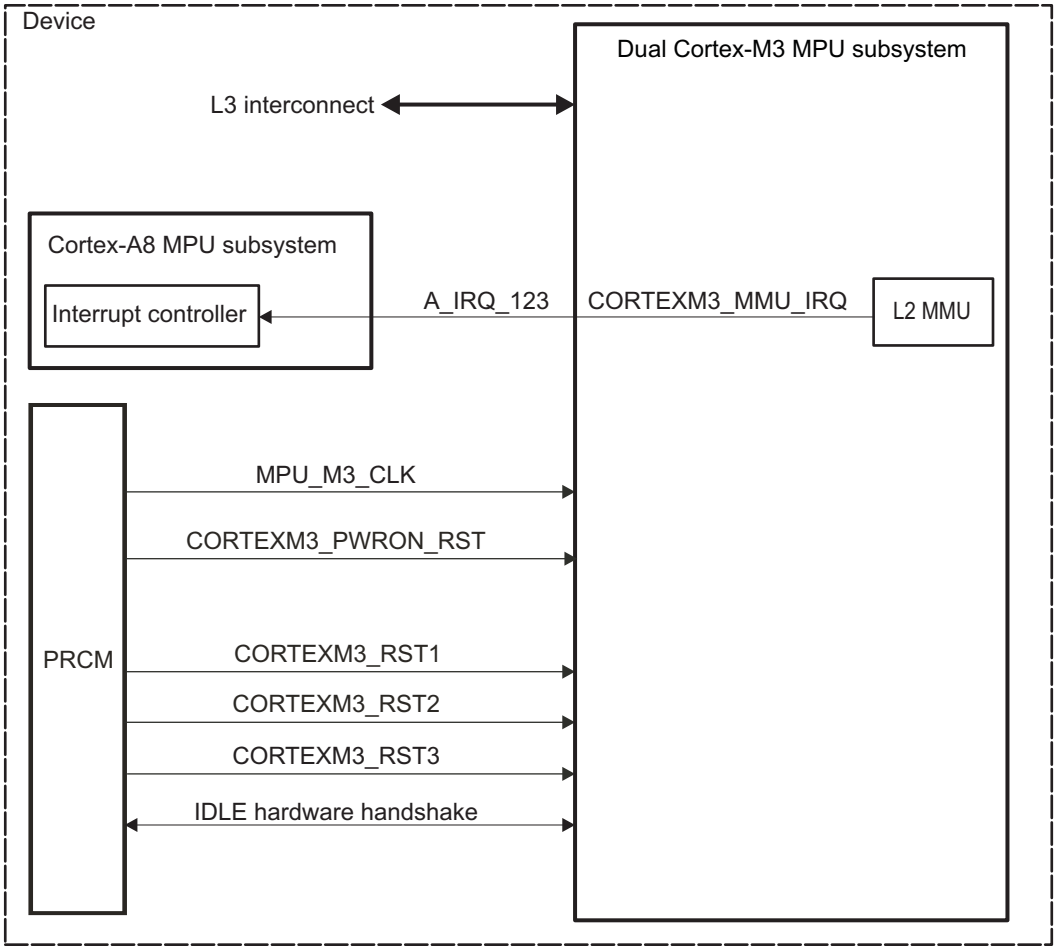



Table 1-1 and Table 1-4 summarize the integration of the module in the device.

**Table 1-1. Integration Attributes**

| Module Instance | Attributes | |
|---|---|---|
| | Power Domain | Interconnect |
| CORTEXM3 | PD_ALWAYS_ON | L3 |

**Table 1-2. Hardware Requests**

| Interrupt Requests | | | | |
|---|---|---|---|---|
| Module Instance | Source Signal Name | Destination Signal Name | Destination | Description |
| L2MMU | CORTEXM3_MMU_IRQ | A_IRQ_123 | Cortex-A8 MPU INTC | L2 MMU Fault Interrupt |

**Table 1-3. Clocks**

| Module Instance | Source Signal Name | Destination Signal Name | Source | Description |
|---|---|---|---|---|
| CORTEXM3 | MPU_M3_CLK | MPU_M3_CLK | PRCM module | Interface and Functional Clock |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-4. Resets**

| Module Instance | Source Signal Name | Destination Signal Name | Source | Description |
|---|---|---|---|---|
| CORTEXM3 | CORTEXM3_PWRON_RST | CORTEXM3_PWRON_RST | PRCM module | Power-on reset, used to reset the whole Media Controller Subsystem |
| | CORTEXM3_RST1 | CORTEXM3_RST1 | PRCM module | Reset signal to the HDVPSS |
| | CORTEXM3_RST2 | CORTEXM3_RST2 | PRCM module | Reset signal to the HDVICP2 |
| | CORTEXM3_RST3 | CORTEXM3_RST3 | PRCM module | Reset signal to the shared cache and L2 MMU |

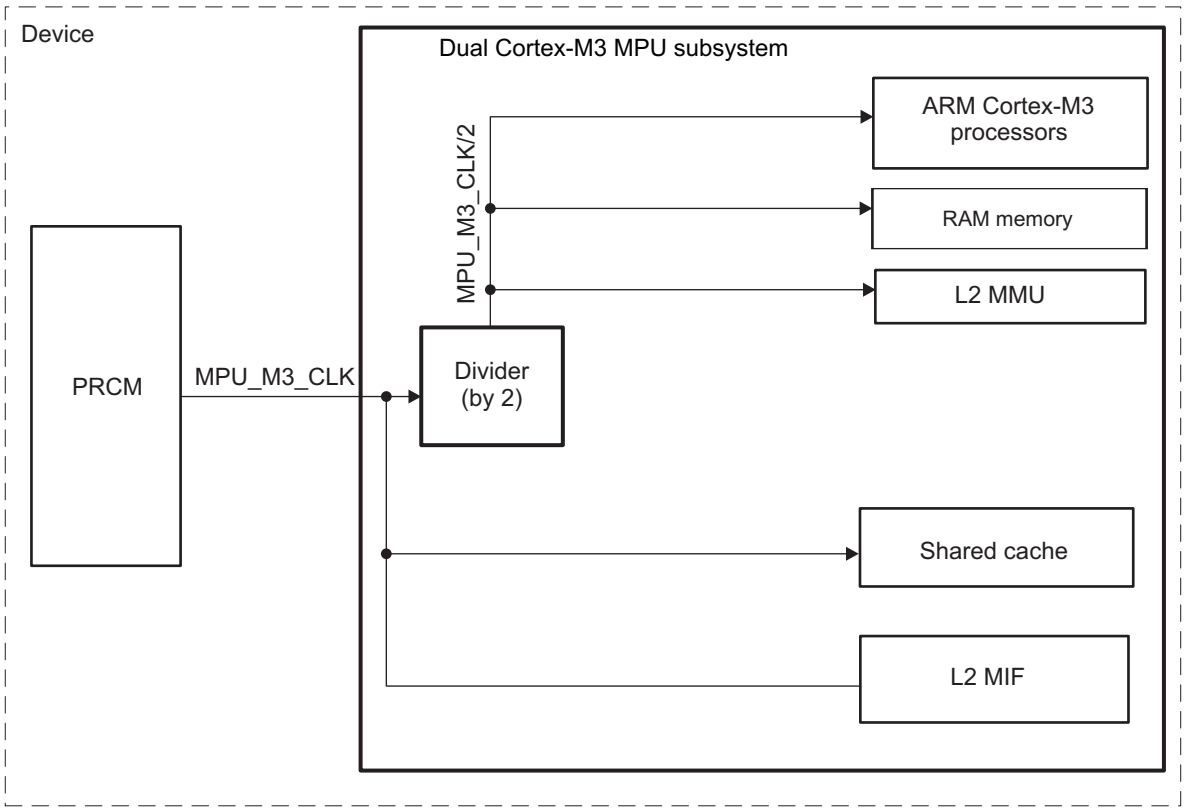
### 1.2.2.1 Dual Cortex-M3 MPU Subsystem Clock and Reset Distribution

#### *1.2.2.1.1 Clock Distribution*

The dual Cortex-M3 subsystem receives only one clock, MPU_M3_CLK, which is divided in two for each ARM Cortex-M3 processor, RAM memory and the L2 MMU module. The shared cache and the L2 MIF are directly clocked by the MPU_M3_CLK, without any division.

Figure 1-4 shows the clocking scheme of the MPU subsystem.

**Figure 1-4. Dual Cortex-M3 MPU Subsystem Clocking Scheme**



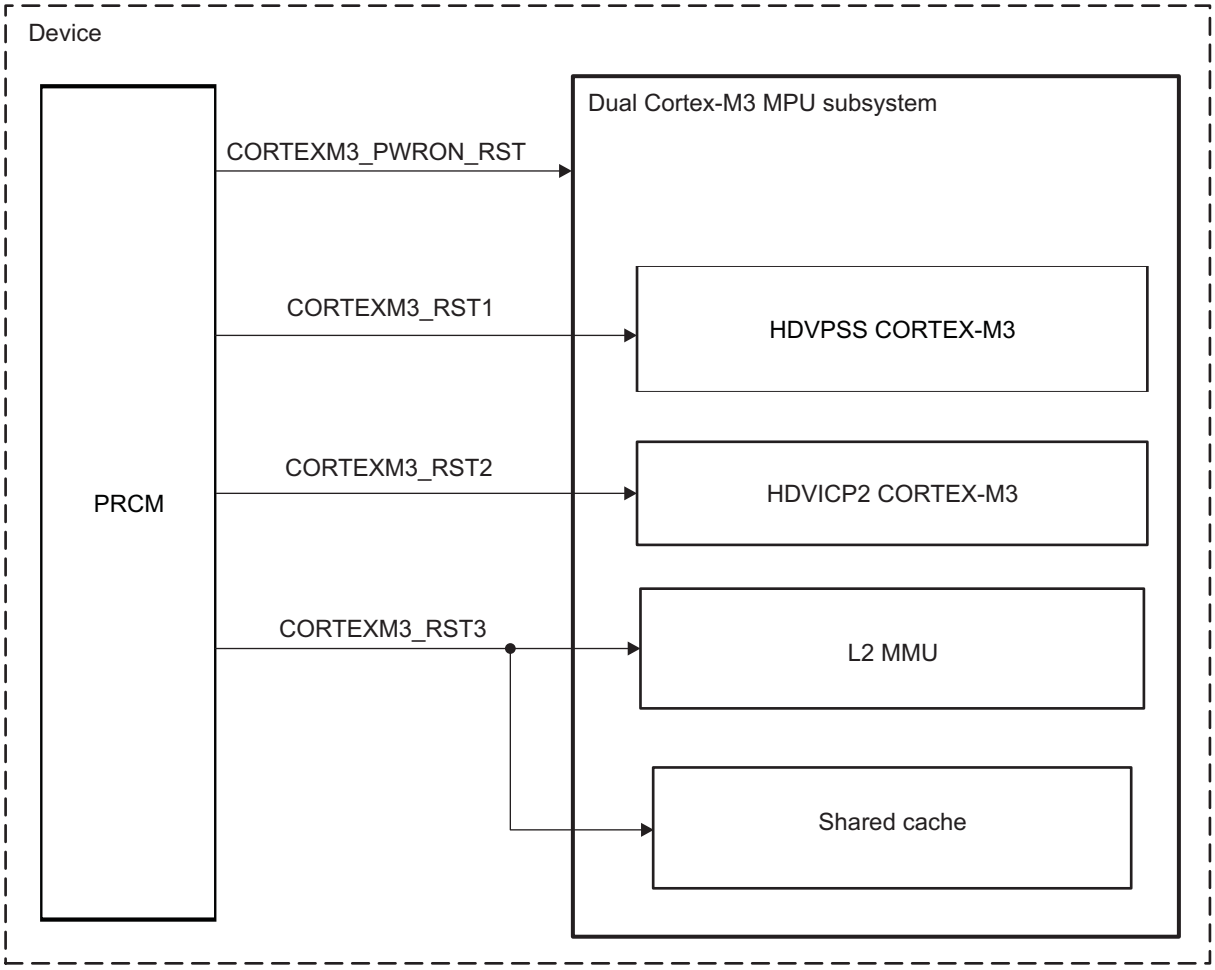

#### *1.2.2.1.2 Reset Distribution*

Three reset signals controlled by the power, reset, and clock management (PRCM) module let the two ARM Cortex-M3 processors and the rest of the dual Cortex-M3 MPU subsystem be reset independently. These three reset signals are: CORTEXM3_RST1, CORTEXM3_RST2, and CORTEXM3_RST3.

Figure 1-5 shows the reset scheme of the MPU subsystem.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-5. Dual Cortex-M3 MPU Subsystem Reset Scheme**




## *1.2.3 Dual Cortex-M3 MPU Subsystem Functional Description*

### 1.2.3.1 Dual Cortex-M3 MPU Subsystem Block Diagram

The dual Cortex-M3 MPU subsystem integrates the following group of submodules:

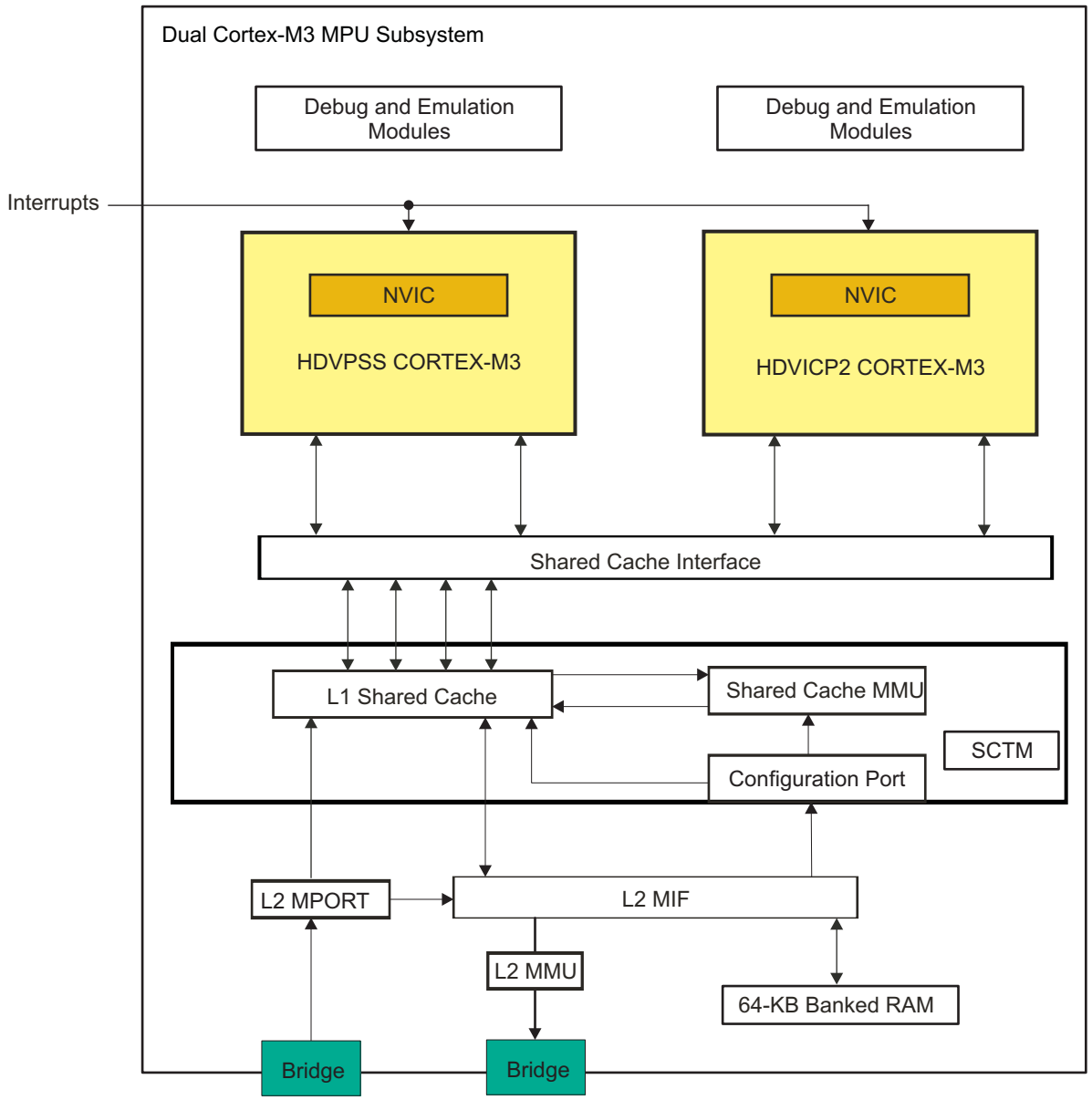
- Two ARM Cortex-M3 processors: Two cores, one core dedicated to HDVPSS and the other core dedicated to HDVICP2. For a description of the ARM Cortex-M3 processor, see the ARM Cortex-M3 Technical Reference Manual, available through your TI representative.
- Interrupt controller (NVIC): To facilitate parallel processing, the interrupt mapping is the same for the two cores. Each Cortex-M3 receives the same interrupts except for a few internal interrupts.
- Shared cache interface: The cache interface converts the data between the different protocols in the subsystem. Four ports are required to support the four buses from the ARM Cortex-M3 processors (two for each processor).The instruction and data connections from each ARM Cortex-M3 are multiplexed, but the ARM Cortex-M3 prevents conflicts on this connection. Default cache policies are provided through the sideband signals and are not used to access the cache. Cache ability is provided through the MMU inside the cache.
- Shared cache: Allows basic maintenance operations, which are performed through a dedicated interface: preload, lock, clean (write out dirty lines, but do not invalidate directly), and invalidate.
- Shared cache MMU: Provides the multi-access cache with region-based address translation, read/write control, access type control, and multilevel cache maintenance. Access to the shared cache MMU is done only under privilege mode. The shared cache MMU can be programmed by the Cortex-A8 MPU subsystem through the Cortex-M3 slave port.


Copyright © 2011–2016, Texas Instruments Incorporated

- SCTM: Counter Timer Module, embedded in the Shared cache. Interconnect configuration port: Ca-che maintenance and MMU configuration are done through an interconnect slave port. Accesses must be performed to a noncacheable area that must be defined within the shared cache MMU. Interconnect accesses are generated from the L2 MIF.
- L2 MMU: Provides address translation for all the accesses done from the Cortex-M3 to the level 3 (L3) interconnect. The L2 MMU can be programmed by the Cortex-A8 MPU subsystem through the Cortex-M3 slave port.
- L3 interconnect port: Allows access to the system memories and peripherals. For the address mapping of the modules in the dual Cortex-M3 MPU subsystem, see the Memory Map section of the device data manual.
- Banked RAM memory. The address range of the on-chip banked RAM memory is from 0x5502 0000 to 0x5503 FFFF.

Figure 1-6 shows the block diagram of the dual Cortex-M3 MPU subsystem.

**Figure 1-6. Dual Cortex-M3 MPU Subsystem**





Copyright © 2011–2016, Texas Instruments Incorporated

### 1.2.4 Shared Cache

Table 1-5 describes the shared cache configuration in the dual Cortex-M3 MPU subsystem platform.

**Table 1-5. Shared Cache Configuration**

| Parameter | Value |
|---|---|
| Way | 4 |
| Size | 32KB |
| Bank Elements | 32b |
| Bank Number | 16 |
| Slave Interface Data Size | 32b |
| Master Interface Data Size | 64b |
| Line size | 256b |
| MMU lookup | Included |
| Number of slaves | 4 |
| Number of masters | 1 |
| Number of fill/prefetch buffers | 4 prefetch buffers |
| Slave Types | Shared cache interface |

Shared cache allows basic maintenance operations, which are performed through a dedicated interface:

1. Preload
2. Lock
3. Clean
4. Invalidate

Maintenance of the cache is performed between the start and end addresses. This allows for direct control of memory regions. All maintenance operations occur in the background and can generate an interrupt when they complete. Such operations are protected by software semaphore, because only one operation at a time can be performed. The maintenance operations can also be performed using MMU small entries.

### 1.2.5 Shared Cache MMU

The MMU for the shared cache provides the multi-access cache with region-based address translation, read/write control, access type control, and multilevel cache maintenance. Table 1-6 describes the MMU configuration in the dual Cortex-M3 MPU subsystem.

**Table 1-6. Shared Cache MMU Configuration**

| Parameter | Harware Configurable |
|---|---|
| Number of large pages | 4 entries |
| Size of large pages | 32MB or 512MB |
| Number of medium pages | 2 entries |
| Size of medium pages | 128KB or 256KB |
| Number of small pages | 10 entries |
| Size of small pages | 4KB or 16KB |
| Number of patch pages | Not included |
| Size of line pages | 256-bit |
| Number of comparison interfaces | 4 |
| Number of comparator sets | 1 |
| Write pipeline data comparison | Disabled |
| Number of unicache maintenance interfaces | 3 |
| Size of entry address | 32-bit |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.2.6 L2 MMU

An additional MMU provides address translation for the accesses done from the dual Cortex-M3 MPU subsystem to the L3 interconnect. The main characteristics of this MMU are:

- 32 entries
- Compatible with ARMv6 architecture MMU translation tables (protection bits not used)
- Page-based or access-based endianness conversion
- 2-level descriptor hierarchy
- One intermediate page table
- Four page sizes (16MB, 1MB, 64KB, 4KB)
- Page table alignment on 128-byte boundary for ARM11™ compatibility

The configuration of the MMU can be done from one of the Cortex-M3 cores or from the L3 interconnect slave port. The accesses done to configure the MMU cannot be part of a burst access. When an MMU fault occurs, an interrupt is generated. This interrupt is connected to the Cortex-M3 cores, but it is generated outside of the dual Cortex-M3 MPU subsystem and is connected to the dual Cortex™-A8 MPU subsystem. When an MMU fault occurs, no other access can be done through the L2 MMU; the current faulty access is stalled, thereby stalling one or both Cortex-M3 cores. The dual Cortex-A8 MPU that received the MMU fault must clean up the fault to resume the execution of the code (or reset Cortex-M3).


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.2.7 Registers

**Table 1-7. Registers**

| Register Name | Type | Register Description | Address Offset |
|---|---|---|---|
| CACHE_CONFIG | RW | Configuration Register | 0000 0004h |
| CACHE_INT | RW | Interrupt Register | 0000 0008h |
| CACHE_OCP | RW | OCP Interface Register | 0000 000Ch |
| CACHE_MAINT | RW | Maintenance Configuration Register | 0000 0010h |
| CACHE_MTSTART | RW | Maintenance Start Configuration Register | 0000 0014h |
| CACHE_MTEND | RW | Maintenance End Configuration Register | 0000 0018h |
| CACHE_CTADDR | RW | Cache Test Address Register | 0000 001Ch |
| CACHE_CTDATA | RW | Cache Test Data Register | 0000 0020h |
| CACHE_SCTM_CTCNTL | RW | Counter Timer Control Register | 0000 0400h |
| CACHE_SCTM_CTSTMCNTL | RW | Counter Timer STM Control Register | 0000 0420h |
| CACHE_SCTM_CTSTMMSTID | RW | Counter Timer STM Master ID Register | 0000 0424h |
| CACHE_SCTM_CTSTMINTVL | RW | Counter Timer STM Interval Register | 0000 0428h |
| CACHE_SCTM_CTSTMSEL0 | RW | Counter Timer STM Count Select Register 0 | 0000 042Ch |
| CACHE_SCTM_CTSTMSEL1 | RW | Counter Timer STM Count Select Register 1 | 0000 0430h |
| CACHE_SCTM_TINTVL_R_i | RW | Counter Timer Interval Number Debug Event Register | 0000 0440h + (04h*i) |
| CACHE_SCTM_CTDBGNUM | RW | Counter Timer Number Debug Event Register | 0000 047Ch |
| CACHE_SCTM_CTGNBL0 | RW | Counter Timer Global Enable Register 0 | 0000 04F0h |
| CACHE_SCTM_CTGNBL1 | RW | Counter Timer Global Enable Register 1 | 0000 04F4h |
| CACHE_SCTM_CTGRST0 | RW | Counter Timer Global Reset Register 0 | 0000 04F8h |
| CACHE_SCTM_CTGRST1 | RW | Counter Timer Global Reset Register 1 | 0000 04FCh |
| CACHE_SCTM_CTCRWT_i | RW | Counter Timer Control Register 0-31 | 0000 0500h + (04h*i) |
| CACHE_SCTM_CTCNTR_k | RW | Counter Timer Count Register 0-31 | 0000 0580h + (04h*k) |
| CACHE_MMU_LARGE_ADDR_l | RW | Large Page Address | 0000 0800h + (04h*l) |
| CACHE_MMU_LARGE_XLTE_l | RW | Large Page Translated Address | 0000 0820h + (04h*l) |
| CACHE_MMU_LARGE_POLY_l | RW | Large Page Policy | 0000 0840h + (04h*l) |
| CACHE_MMU_MED_ADDR_m | RW | Medium Page Address | 0000 0860h + (04h*m) |
| CACHE_MMU_MED_XLTE_m | RW | Medium Page Translated Address | 0000 08A0h + (04h*m) |
| CACHE_MMU_MED_POLY_m | RW | Medium Page Policy | 0000 08E0h + (04h*m) |
| CACHE_MMU_SMALL_ADDR_n | RW | Small Page Address | 0000 0920h + (04h*n) |
| CACHE_MMU_SMALL_XLTE_n | RW | Small Page Translated Address | 0000 09A0h + (04h*n) |
| CACHE_MMU_SMALL_POLY_n | RW | Small Page Policy | 0000 0A20h + (04h*n) |
| CACHE_MMU_MAINT | RW | MMU Start/End Maintenance Config Register | 0000 0CA8h |
| CACHE_MMU_MTSTART | RW | Maintenance Start Address Register | 0000 0CACh |
| CACHE_MMU_MTEND | RW | Maintenance End Address Register | 0000 0CB0h |
| CACHE_MMU_MAINTST | RW | Maintenance Status Register | 0000 0CB4h |
| CACHE_MMU_MMUCONFIG | RW | MMU Configuration Register | 0000 0CB8h |

#### 1.2.7.1 Configuration Registers (CACHE_CFG)

The Configuration Registers (CACHE_CFG) is described in docato-extra-info-title Configuration Registers (CACHE_CFG) Field DescriptionsTable 1-8 .


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-8. Configuration Registers (CACHE_CFG) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | Reserved | Reserved | R | 0 |
| 1 | BYPASS | bypass cache | RW | 0 |
| | | 0x0: everything is non-cacheable | | |
| | | 0x1: everything is cacheable | | |
| 0 | Reserved | Reserved | R | 0 |

### 1.2.7.2 Interrupt Register (CACHE_INT)

The Interrupt Register (CACHE_INT) is described in Table 1-9.

**Table 1-9. Interrupt Register (CACHE_INT) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:9 | Reserved | Reserved | R | 0 |
| 8:5 | PORT | slave interface number that has recorded an error | RW W1toClr | 0 |
| 4 | READ | interface read response error | RW W1toClr | 0 |
| 3 | WRITE | interface write response error | RW W1toClr | 0 |
| 2 | MAINT | maintenance is completed | RW W1toClr | 0 |
| 1 | PAGEFAULT | AMMU page fault | RW W1toClr | 0 |
| 0 | CONFIG | configuration error | RW W1toClr | 0 |

### 1.2.7.3 Interface Configuration Register (CACHE_OCP)

The Interface Configuration Register (CACHE_OCP) is described in Table 1-10.

**Table 1-10. Interface Configuration Register (CACHE_OCP) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | Reserved | Reserved | R | 0 |
| 5 | CLEANBUF | clean write and prefetch buffers in cache | RW | 0 |
| | | 0x0: do not clean | | |
| | | 0x1: clean | | |
| 4 | PREFETCH | always prefetch data | RW | 0 |
| | | 0x0: follow MMU policies | | |
| | | 0x1: always prefetch | | |
| 3 | CACHED | follow cacheable sideband signals | RW | 1 |
| | | 0x0: reads always not allocated, writes write through if cached | | |
| | | 0x1: slave sideband signals determine policy | | |
| 2 | WRALLOCATE | follow write allocate sideband signals | RW | 0 |
| | | 0x0: no writes are allocated independent to sideband | | |
| | | 0x1: follow sideband | | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-10. Interface Configuration Register (CACHE_OCP) Field Descriptions (continued)**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 1 | WRBUFFER | write throughs and write back no allocate are buffered | RW | 0 |
| | | 0x0: write throughs and write back no allocated are not buffered | | |
| | | 0x1: write throughs and write back no allocated are buffered | | |
| 0 | WRAP | OCP wrap mode (critical word first) | RW | 0 |
| | | 0x0: disabled | | |
| | | 0x1: enabled | | |

### 1.2.7.4 Maintenance Configuration Register (CACHE_MAINT)

The Maintenance Configuration Register (CACHE_MAINT) is described in Table 1-11.

**Table 1-11. Maintenance Configuration Register (CACHE_MAINT) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | Reserved | Reserved | R | 0 |
| 5 | INTERRUPT | generate interrupt when maintenance operation is complete | RW | 0 |
| | | 0x0: do not generate interrupt | | |
| | | 0x1: generate interrupt | | |
| 4 | INVALIDATE | invalidate lines in region defined by maintenance start/end addresses | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: invalidate | | |
| 3 | CLEAN | evict dirty lines in region defined by maintenance start/end addresses | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: clean | | |
| 2 | UNLOCK | unlock region defined by maintenance start/end addresses | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: unlock | | |
| 1 | LOCK | lock region defined by maintenance start/end addresses | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: lock | | |
| 0 | PRELOAD | preload region defined by maintenance start/end addresses | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: preload | | |

### 1.2.7.5 Maintenance Start Configuration Register (CACHE_MTSTART)

The Maintenance Start Configuration Register (CACHE_MTSTART) is described in Table 1-12.

**Table 1-12. Maintenance Start Configuration Register (CACHE_MTSTART) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | START_ADDR | start address of maintenance operations, reset to 0 when finished | RW | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.2.7.6 Maintenance End Configuration Register (CACHE_MTEND)

The Maintenance End Configuration Register (CACHE_MTEND) is described in Table 1-13.

**Table 1-13. Maintenance End Configuration Register (CACHE_MTEND) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | END_ADDR | end address of maintenance operations, reset to 0 when finished | RW | 0 |

### 1.2.7.7 Cache Test Address Register (CACHE_CTADDR)

The Cache Test Address Register (CACHE_CTADDR) is described in Table 1-14.

**Table 1-14. Cache Test Address Register (CACHE_CTADDR) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | ADDRESS | address of cache visibility when read CTDATA register, autoincrements | RW | 0 |

### 1.2.7.8 Cache Test Data Register (CACHE_CTDATA)

The Cache Test Data Register (CACHE_CTDATA) is described in Table 1-15.

**Table 1-15. Cache Test Data Register (CACHE_CTDATA) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | DATA | cache data at address of CTADDR register, CTADDR autoincrements each time CTDATA is read | RW | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.2.7.9 Counter Timer Control Register (CACHE_SCTM_CTCNTL)

he Counter Timer Control Register (CACHE_SCTM_CTCNTL) is described in Table 1-16.

**Table 1-16. Counter Timer Control Register (CACHE_SCTM_CTCNTL) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:26 | NUMSTM | Number of timers that can export via STM | R | 0 |
| 25:18 | NUMINPT | Number of event input signals | R | 1Fh |
| 17:13 | NUMTIMR | Number of timers in the module | R | 2 |
| 12:7 | NUMCNTR | Number of counters in the module | R | 8 |
| 6:3 | REVISION | Revision ID of SCTM | R | TI internal data |
| 2:1 | IDLEMODE | Idle mode control | RW | 2 |
| | | 0x0: Force Idle mode | | |
| | | 0x1: Ths SCTM will acknoledge the idle request, but never transition to the idle state | | |
| | | 0x2: Ths SCTM uses the smart idle protocol. This is the default mode | | |
| | | 0x3: Since the SCTM does not support internal wakeup, this mode is identical to smart_idle | | |
| 0 | ENBL | SCTM global enable | RW | 0 |
| | | 0x0: Ths mode is disable. Only the configuration interface is functional. All other logic is reset | | |
| | | 0x1: The module is enabled and individual counter/timers can be configured | | |

### 1.2.7.10 Counter Timer STM Control Register (CACHE_SCTM_CTSTMCNTL)

The Counter Timer STM Control Register (CACHE_SCTM_CTSTMCNTL) is described in Table 1-17.

**Table 1-17. Counter Timer STM Control Register (CACHE_SCTM_CTSTMCNTL) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:11 | Reserved | Reserved | R | 0 |
| 10 | XPORTACT | Indicates if a frame is currently being written to the STM | R | 0 |
| 9:5 | NUMXPORT | The total number of counters designated for export. This will be used as the count in the CSM and CCM headers. The value written should be the total number of counters designated for export -1 | RW | 0 |
| 4 | CCMXPORT | SW control of CCM message export | RW | 0 |
| 3 | CCMVAIL | SCTM supports CCM export | R | 0 |
| 2 | CSMXPORT | SW control of CSM message export | RW | 0 |
| 1 | SENDOVR | Send overflow data in CSM frame | RW | 1 |
| 0 | ENBL | STM global enable | RW | 0 |
| | | Read 0x0: The STM function of the SCTM is disabled | | |
| | | Read 0x1: The STM function of the SCTM is enabled | | |

### 1.2.7.11 Counter Timer STM Master ID Register (CACHE_SCTM_CTSTMMSTID)

The Counter Timer STM Master ID Register (CACHE_SCTM_CTSTMMSTID) is described in Table 1-18.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-18. Counter Timer STM Master ID Register (CACHE_SCTM_CTSTMMSTID) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:7 | Reserved | Reserved | R | 0 |
| 6:0 | MASTID | HW Master ID for this module. | RW | 0 |

### 1.2.7.12 Counter Timer STM Interval Register (CACHE_SCTM_CTSTMINTVL)

The Counter Timer STM Interval Register (CACHE_SCTM_CTSTMINTVL) is described in Table 1-19.

**Table 1-19. Counter Timer STM Interval Register (CACHE_SCTM_CTSTMINTVL) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | Reserved | Reserved | R | 0 |
| 15:0 | INTERVAL | Periodic export interval | RW | 0 |

### 1.2.7.13 Counter Timer STM Count Select Register 0 (CACHE_SCTM_CTSTMSEL0)

The Counter Timer STM Count Select Register 0(CACHE_SCTM_CTSTMSEL0) is described in Table 1-20.

**Table 1-20. Counter Timer STM Count Select Register 0(CACHE_SCTM_CTSTMSEL0) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | COUNTSEL | The counter selection bit-field | RW | 0 |

### 1.2.7.14 Counter Timer Interval Number Debug Event Register (CACHE_SCTM_TINTVL_R_i)

The Counter Timer Interval Number Debug Event Register (CACHE_SCTM_TINTVL_R_i) is described in Table 1-21.

**Table 1-21. Counter Timer Interval Number Debug Event Register (CACHE_SCTM_TINTVL_R_i) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | INTERVAL | Interval match value for the timers in the SCTM | RW | 0 |

### 1.2.7.15 Counter Timer Number Debug Event Register (CACHE_SCTM_CTDBGNUM)

The Counter Timer Number Debug Event Register (CACHE_SCTM_CTDBGNUM) is described in Table 1-22.

**Table 1-22. Counter Timer Number Debug Event Register (CACHE_SCTM_CTDBGNUM) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:3 | Reserved | Reserved | R | 0 |
| 2:0 | NUMEVT | Number of input selectors for debug events | R | 0 |

### 1.2.7.16 Counter Timer Global Enable Register 0 (CACHE_SCTM_CTGNBL0)

The Counter Timer Global Enable Register 0 (CACHE_SCTM_CTGNBL0) is described in Table 1-23.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-23. Counter Timer Global Enable Register 0(CACHE_SCTM_CTGNBL0) Field Description**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:8 | Reserved | Reserved | R | 0 |
| 7:0 | ENABLE | The counter enable bit-field | RW | 0 |

### 1.2.7.17 Counter Timer Global Reset Register 0 (CACHE_SCTM_CTGRST0)

The Counter Timer Global Reset Register 0 (CACHE_SCTM_CTGRST0) is described in Table 1-24.

**Table 1-24. Counter Timer Global Reset Register 0(CACHE_SCTM_CTGRST0) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:8 | Reserved | Reserved | R | 0 |
| 7:0 | RESET | The counter reset bit-field | RW | 0 |

### 1.2.7.18 Counter Timer Control Register 0-31 (CACHE_SCTM_CTCRWT_i)

he Counter Timer Control Register 0-31 (CACHE_SCTM_CTCRWT_i) is described in Table 1-25.

**Table 1-25. Counter Timer Control Register 0-31(CACHE_SCTM_CTCRWT_i) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:21 | Reserved | Reserved | R | 0 |
| 20:16 | INPSEL | Counter Timer input selection | RW | 0 |
| 15:11 | Reserved | Reserved | R | 0 |
| 10 | RESTART | Restart the timer after an interval match | RW | 0 |
| 9 | DBG | Signal debug logic on interval match | RW | 0 |
| 8 | INT | Generate interrupt on interval match | RW | 0 |
| 7 | Reserved | Reserved | R | 0 |
| 6 | OVRFLW | Counter has wrapped since it was last read | R | 0 |
| 5 | IDLE | Counter ignores processor IDLE state | RW | 0 |
| 4 | FREE | Counter ignores processor debug halt state | RW | 0 |
| 3 | DURMODE | Counter is in duration or occurrence mode | RW | 0 |
| 2 | CHAIN | Counter is chained to an adjacent counter | RW | 0 |
| 1 | RESET | Counter reset control | RW | 0 |
| 0 | ENBL | Counter enable control | RW | 0 |

### 1.2.7.19 Counter Timer Count Register 0-31 (CACHE_SCTM_CTCNTR_k)

The Counter Timer Count Register 0-31 (CACHE_SCTM_CTCNTR_k) is described in Table 1-26.

**Table 1-26. Counter Timer Count Register 0-31(CACHE_SCTM_CTCNTR_k) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | COUNT | Counter value | R | 0 |

### 1.2.7.20 Large Page Address (CACHE_MMU_LARGE_ADDR_l)

The Large Page Address (CACHE_MMU_LARGE_ADDR_l) is described in Table 1-27.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-27. Large Page Address (CACHE_MMU_LARGE_ADDR_l) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:28 | ADDRESS | logical source address | RW | 0 |
| 27:0 | Reserved | Reserved | R | 0 |

### 1.2.7.21 Large Page Translated Address (CACHE_MMU_LARGE_XLTE_l)

The Large Page Translated Address (CACHE_MMU_LARGE_XLTE_l) is described in Table 1-28.

**Table 1-28. Large Page Translated Address (CACHE_MMU_LARGE_XLTE_l) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:28 | ADDRESS | logical Destination address or translated address | RW | 0 |
| 27:0 | Reserved | Reserved | R | 0 |

### 1.2.7.22 Large Page Policy (CACHE_MMU_LARGE_POLY_l)

The Large Page Policy (CACHE_MMU_LARGE_POLY_l) is described in Table 1-29.

**Table 1-29. Large Page Policy (CACHE_MMU_LARGE_POLY_l) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:24 | Reserved | Reserved | R | 0 |
| 23 | L2_WR_POLICY | L2 write policy | RW | 0 |
| | | 0x0: write through | | |
| | | 0x1: write back | | |
| 22 | L2_ALLOCATE | L2 allocate policy | RW | 0 |
| | | 0x0: no writes are allocated | | |
| | | 0x1: follow sideband | | |
| 21 | L2_POSTED | L2 posted policy | RW | 0 |
| | | 0x0: not posted | | |
| | | 0x1: posted | | |
| 20 | L2_CACHEABLE | L2 cache policy | RW | 0 |
| | | 0x0: non-cacheable | | |
| | | 0x1: cacheable | | |
| 19 | L1_WR_POLICY | L1 write policy | RW | 0 |
| | | 0x0: write through | | |
| | | 0x1: write back | | |
| 18 | L1_ALLOCATE | L1 allocate policy | RW | 0 |
| | | 0x0: no writes are allocated | | |
| | | 0x1: follow sideband | | |
| 17 | L1_POSTED | L1 posted policy | RW | 0 |
| | | 0x0: not posted | | |
| | | 0x1: posted | | |
| 16 | L1_CACHEABLE | L1 cache policy | RW | 0 |
| | | 0x0: non-cacheable | | |
| | | 0x1: cacheable | | |
| 15:8 | Reserved | Reserved | R | 0 |
| 7 | EXCLUSION | cache exclusion | RW | 0 |
| | | 0x0: do not send exclusion sideband | | |
| | | 0x1: send exclusion sideband | | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-29. Large Page Policy (CACHE_MMU_LARGE_POLY_l) Field Descriptions (continued)**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 6 | PRELOAD | preload region | RW | 0 |
| | | 0x0: do not preload | | |
| | | 0x1: preload | | |
| 5 | READ | read only | RW | 0 |
| 4 | EXECUTE | execute only | RW | 0 |
| 3 | VOLATILE | volatile qualifier, | RW | 0 |
| | | 0x0: do not follow volatile qualifier | | |
| | | 0x1: follow volatile qualifier | | |
| 2 | RESERVED | RESERVED | R | 0 |
| 1 | SIZE | size of page | RW | 0 |
| | | 0: 32 MB | | |
| | | 1: 512 MB | | |
| 0 | ENABLE | enable page | RW | 0 |
| | | 0x0: page not enabled | | |
| | | 0x1: page enabled | | |

### 1.2.7.23 Medium Page Address (CACHE_MMU_MED_ADDR_m)

The Medium Page Address (CACHE_MMU_MED_ADDR_m) is described in Table 1-30.

**Table 1-30. Medium Page Address (CACHE_MMU_MED_ADDR_m) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:17 | ADDRESS | Llogical source address | RW | 0 |
| 16:0 | Reserved | Reserved | R | 0 |

### 1.2.7.24 Medium Page Translated Address (CACHE_MMU_MED_XLTE_m)

The Medium Page Translated Address (CACHE_MMU_MED_XLTE_m) is described in Table 1-31.

**Table 1-31. Medium Page Translated Address (CACHE_MMU_MED_XLTE_m) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:17 | ADDRESS | Logical Destination address | RW | 0 |
| 16:0 | Reserved | Reserved | R | 0 |

### 1.2.7.25 Medium Page Policy (CACHE_MMU_MED_POLY_m)

The Medium Page Policy (CACHE_MMU_MED_POLY_m) is described in Table 1-32.

**Table 1-32. Medium Page Policy (CACHE_MMU_MED_POLY_m) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:24 | Reserved | Reserved | R | 0 |
| 23 | L2_WR_POLICY | L2 write policy | RW | 0 |
| | | 0x0: write through | | |
| | | 0x1: write back | | |
| 22 | L2_ALLOCATE | L2 allocate policy | RW | 0 |
| | | 0x0: no writes are allocated | | |
| | | 0x1: follow sideband | | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-32. Medium Page Policy (CACHE_MMU_MED_POLY_m) Field Descriptions (continued)**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 21 | L2_POSTED | L2 posted policy | RW | 0 |
| | | 0x0: not posted | | |
| | | 0x1: posted | | |
| 20 | L2_CACHEABLE | L2 cache policy | RW | 0 |
| | | 0x0: non-cacheable | | |
| | | 0x1: cacheable | | |
| 19 | L1_WR_POLICY | L1 write policy | RW | 0 |
| | | 0x0: write through | | |
| | | 0x1: write back | | |
| 18 | L1_ALLOCATE | L1 allocate policy | RW | 0 |
| | | 0x0: no writes are allocated | | |
| | | 0x1: follow sideband | | |
| 17 | L1_POSTED | L1 posted policy | RW | 0 |
| | | 0x0: not posted | | |
| | | 0x1: posted | | |
| 16 | L1_CACHEABLE | L1 cache policy | RW | 0 |
| | | 0x0: non-cacheable | | |
| | | 0x1: cacheable | | |
| 15:8 | Reserved | Reserved | R | 0 |
| 7 | EXCLUSION | cache exclusion | RW | 0 |
| | | 0x0: do not send exclusion sideband | | |
| | | 0x1: send exclusion sideband | | |
| 6 | PRELOAD | preload region | RW | 0 |
| | | 0x0: do not preload | | |
| | | 0x1: preload | | |
| 5 | READ | preload region | RW | 0 |
| | | 0x0: do not preload | | |
| | | 0x1: preload | | |
| 4 | EXECUTE | execute only | RW | 0 |
| 3 | VOLATILE | volatile qualifier, | RW | 0 |
| | | 0x0: do not follow volatile qualifier | | |
| | | 0x1: follow volatile qualifier | | |
| 2 | Reserved | Reserved | R | 0 |
| 1 | SIZE | size of page | RW | 0 |
| | | 0: 128KB | | |
| | | 1: 256 KB | | |
| 0 | ENABLE | enable page | RW | 0 |
| | | 0x0: page not enabled | | |
| | | 0x1: page enabled | | |

### 1.2.7.26 Small Page Address (CACHE_MMU_SMALL_ADDR_n)

The Small Page Address (CACHE_MMU_SMALL_ADDR_n) is described in Table 1-33.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-33. Small Page Address (CACHE_MMU_SMALL_ADDR_n) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:12 | ADDRESS | logical source address | RW | 0 |
| 11:0 | Reserved | Reserved | R | 0 |

### 1.2.7.27 Small Page Translated Address (CACHE_MMU_SMALL_XLTE_n)

The Small Page Translated Address (CACHE_MMU_SMALL_XLTE_n) is described in Table 1-34.

**Table 1-34. Small Page Translated Address (CACHE_MMU_SMALL_XLTE_n) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:12 | ADDRESS | logical Destination or translated address | RW | 0 |
| 11:0 | Reserved | Reserved | R | 0 |

### 1.2.7.28 Small Page Policy (CACHE_MMU_SMALL_POLY_n)

The Small Page Policy (CACHE_MMU_SMALL_POLY_n) is described in Table 1-35.

**Table 1-35. Small Page Policy (CACHE_MMU_SMALL_POLY_n) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:24 | Reserved | Reserved | R | 0 |
| 23 | L2_WR_POLICY | L2 write policy | RW | 0 |
| | | 0x0: write through | | |
| | | 0x1: write back | | |
| 22 | L2_ALLOCATE | L2 allocate policy | RW | 0 |
| | | 0x0: no writes are allocated | | |
| | | 0x1: follow sideband | | |
| 21 | L2_POSTED | L2 posted policy | RW | 0 |
| | | 0x0: not posted | | |
| | | 0x1: posted | | |
| 20 | L2_CACHEABLE | L2 cache policy | RW | 0 |
| | | 0x0: non-cacheable | | |
| | | 0x1: cacheable | | |
| 19 | L1_WR_POLICY | L1 write policy | RW | 0 |
| | | 0x0: write through | | |
| | | 0x1: write back | | |
| 18 | L1_ALLOCATE | L1 allocate policy | RW | 0 |
| | | 0x0: no writes are allocated | | |
| | | 0x1: follow sideband | | |
| 17 | L1_POSTED | L1 posted policy | RW | 0 |
| | | 0x0: not posted | | |
| | | 0x1: posted | | |
| 16 | L1_CACHEABLE | L1 cache policy | RW | 0 |
| | | 0x0: non-cacheable | | |
| | | 0x1: cacheable | | |
| 15:8 | Reserved | Reserved | R | 0 |
| 7 | EXCLUSION | cache exclusion | RW | 0 |
| | | 0x0: do not send exclusion sideband | | |
| | | 0x1: send exclusion sideband | | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-35. Small Page Policy (CACHE_MMU_SMALL_POLY_n) Field Descriptions (continued)**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 6 | PRELOAD | preload region | RW | 0 |
| | | 0x0: do not preload | | |
| | | 0x1: preload | | |
| 5 | READ | preload region | RW | 0 |
| | | 0x0: do not preload | | |
| | | 0x1: preload | | |
| 4 | EXECUTE | execute only | RW | 0 |
| 3 | VOLATILE | volatile qualifier, | RW | 0 |
| | | 0x0: do not follow volatile qualifier | | |
| | | 0x1: follow volatile qualifier | | |
| 2 | RESERVED | Reserved | R | 0 |
| 1 | SIZE | size of page | RW | 0 |
| | | 0: 128KB | | |
| | | 1: 256 KB | | |
| 0 | ENABLE | enable page | RW | 0 |
| | | 0x0: page not enabled | | |
| | | 0x1: page enabled | | |

### 1.2.7.29 Small Page Maintenance Configuration (CACHE_MMU_SMALL_MAINT_n)

The Small Page Maintenance Configuration (CACHE_MMU_SMALL_MAINT_n) is described in Table 1-36.

**Table 1-36. Small Page Maintenance Configuration (CACHE_MMU_SMALL_MAINT_n) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:5 | Reserved | Reserved | R | 0 |
| 4 | INTERRUPT | generate interrupt when maintenance operation is complete | RW | 0 |
| 3 | INVALIDATE | invalidate page | RW | 0 |
| 2 | CLEAN | evict page | RW | 0 |
| 1 | LOCK | lock page | RW | 0 |
| 0 | PRELOAD | preload page | RW | 0 |

### 1.2.7.30 MMU Start/End Maintenance Config Register (CACHE_MMU_MAINT)

The MMU Start/End Maintenance Config Register (CACHE_MMU_MAINT) is described in Table 1-37.

**Table 1-37. MMU Start/End Maintenance Config Register (CACHE_MMU_MAINT) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:11 | Reserved | Reserved | R | 0 |
| 10 | G_FLUSH | Global flush bit | RW | 0 |
| | | 0x0: do nothing, global flush done | | |
| | | 0x1: invalidate L1+L2 | | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-37. MMU Start/End Maintenance Config Register (CACHE_MMU_MAINT) Field Descriptions (continued)**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 9 | L2_CACHE | do maintenance operation in L2 Cache. | RW | 0 |
| | | Note: Hardware ensures that the maintenance operations are done in L1 first and then in L2 | | |
| | | 0x0: do nothing | | |
| | | 0x1: perform maintenance operation selected by the maintenance bits | | |
| 8 | L1_CACHE2 | do maintenance operation in L1 Cache 2 | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: perform maintenance operation selected by the maintenance bits | | |
| 7 | L1_CACHE1 | do maintenance operation in L1 Cache1 | RW | 0 |
| | | 0x0: do not perform maintenance operation | | |
| | | 0x1: perform maintenance operation selected by the maintenance bits | | |
| 6 | CPU_INTERRUPT | generate interrupt to cpu when maintenance operation initiated by CPU is complete | RW | 0 |
| | | 0x0: do not generate interrupt | | |
| | | 0x1: generate interrupt | | |
| 5 | HOST_INTERRUPT | generate interrupt when maintenance operation is complete | RW | 0 |
| | | 0x0: do not generate interrupt | | |
| | | 0x1: generate interrupt | | |
| 4 | INVALIDATE | invalidate lines in region defined by maintenance start/end addresses | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: invalidate | | |
| 3 | CLEAN | evict dirty lines in region defined by maintenance start/end addresses | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: clean | | |
| 2 | UNLOCK | unlock region defined by maintenance start/end addresses | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: unlock | | |
| 1 | LOCK | lock region defined by maintenance start/end addresses | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: lock | | |
| 0 | PRELOAD | preload region defined by maintenance start/end addresses | RW | 0 |
| | | 0x0: do nothing | | |
| | | 0x1: preload | | |

### 1.2.7.31 Maintenance Start Address Register (CACHE_MMU_MTSTART)

The Maintenance Start Address Register (CACHE_MMU_MTSTART) is described in Table 1-38.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-38. Maintenance Start Address Register (CACHE_MMU_MTSTART) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | BEGIN_ADDRESS | Start address of maintenance operations, resets to 0 when finished | RW | 0 |

### 1.2.7.32 Maintenance End Address Register (CACHE_MMU_MTEND)

The Maintenance End Address Register (CACHE_MMU_MTEND) is described in Table 1-39.

**Table 1-39. Maintenance End Address Register (CACHE_MMU_MTEND) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | END_ADDRESS | End address of maintenance operations, resets to 0 when finished | RW | 0 |

### 1.2.7.33 Maintenance Status Register (CACHE_MMU_MAINTST)

The Maintenance Status Register (CACHE_MMU_MAINTST) is described in Table 1-40.

**Table 1-40. Maintenance Status Register (CACHE_MMU_MAINTST) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | Reserved | Reserved | R | 0 |
| 0 | STATUS | status bit | R | 0 |
| | | Read 0x0: do nothing, maintenance completed | | |
| | | Read 0x1: maintenance ongoing | | |

### 1.2.7.34 MMU Configuration Register (CACHE_MMU_MMUCONFIG)

The MMU Configuration Register (CACHE_MMU_MMUCONFIG) is described in Table 1-41.

**Table 1-41. MMU Configuration Register (CACHE_MMU_MMUCONFIG) Field Descriptions**

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | Reserved | Reserved | R | 0 |
| 1 | PRIVILEDGE | Privilege bit. once this bit is set, only global flush, debugger, or hardware reset can clear | RW | 0 |
| | | 0x0: CPU can access everything | | |
| | | 0x1: CPU can only access maintenance, and DMA can | | |
| | | not access MMU at all | | |
| 0 | Reserved | Reserved | R | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.3 C674x DSP

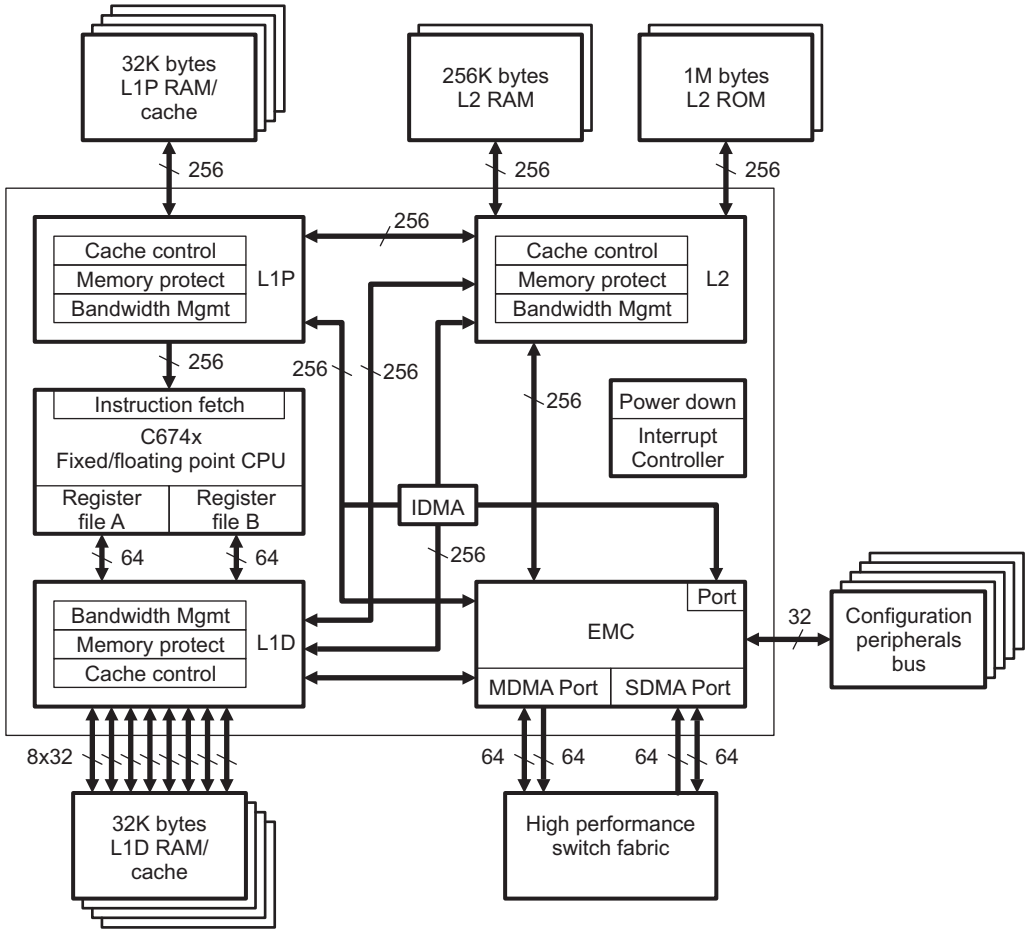
### 1.3.1 Introduction

The DSP subsystem (Figure 1-7) includes TI's standard TMS320C674x megamodule and several blocks of internal memory (L1P, L1D, and L2). This chapter provides an overview of the DSP subsystem and the following considerations associated with it:

- Memory mapping
- Interrupts
- Power management

For more information on the TMS320C674x megamodule, see the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5), the *TMS320C674x DSP CPU and Instruction Set Reference Guide* (SPRUFE8), and the *TMS320C674x DSP Cache User's Guide* (SPRUG82).

**Figure 1-7. TMS320C674x Megamodule Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

### 1.3.2 TMS320C674x Megamodule

The C674x megamodule (Figure 1-7) consists of the following components:

- TMS320C674x CPU
- Internal memory controllers:
  - Level 1 program memory controller (PMC)
  - Level 1 data memory controller (DMC)
  - Level 2 unified memory controller (UMC)
  - Extended memory controller (EMC)
  - Internal direct memory access (IDMA) controller
- Internal peripherals:
  - Interrupt controller (INTC)
  - Power-down controller (PDC)
  - Bandwidth manager (BWM)
- Advanced event triggering (AET)

For more information about each of these controllers, see the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

#### 1.3.2.1 Internal Memory Controllers

The C674x megamodule implements a two-level internal cache-based memory architecture with external memory support. Level 1 memory (L1) is split into separate program memory (L1P memory) and data memory (L1D memory). L1 memory is accessible to the CPU without stalls. Level 2 memory (L2) can also be split into L2 RAM (normal addressable on-chip memory) and L2 cache for caching external memory locations. The internal direct memory access controller (IDMA) manages DMA among the L1P, L1D, and L2 memories.

#### 1.3.2.2 Internal Peripherals

The C674x megamodule includes the following internal peripherals:

- DSP interrupt controller (INTC)
- DSP power-down controller (PDC)
- Bandwidth manager (BWM)
- Internal DMA (IDMA) controller

This section briefly describes the INTC, PDC, BWM, and IDMA controller. For more information on these internal peripherals, see the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

##### 1.3.2.2.1 Interrupt Controller (INTC)

The C674x megamodule includes an interrupt controller (INTC) to manage CPU interrupts. The INTC maps DSP device events to 12 CPU interrupts. All DSP device events are listed in Section 1.7.3. The INTC is fully described in the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

###### 1.3.2.2.1.1 Interrupt Controller Registers

For more information on the DSP interrupt controller (INTC) registers, see the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

##### 1.3.2.2.2 Power-Down Controller (PDC)

The C674x megamodule includes a power-down controller (PDC). The PDC can power-down all of the following components of the C674x megamodule and internal memories of the DSP subsystem:


Copyright © 2011–2016, Texas Instruments Incorporated

- C674x CPU
- Level 1 program memory controller (PMC)
- Level 1 data memory controller (DMC)
- Level 2 unified memory controller (UMC)
- Extended memory controller (EMC)
- Internal direct memory access (IDMA) controller
- L1P memory
- L1D memory
- L2 memory

This device supports the static power-down feature from the C674x megamodule. The *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5) describes the power-down control in more detail.

- Static power-down: The PDC initiates power-down (clock gating) of the entire C674x megamodule and all internal memories immediately upon command from software.

Static power-down (clock gating) affects all components of the C674x megamodule and all internal memories. Software can initiate static power-down by way of a register bit in the power-down controller command register (PDCCMD) of the PDC. For more information on the PDC, see the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

#### *1.3.2.2.3 Bandwidth Manager (BWM)*

The bandwidth manager (BWM) provides a programmable interface for optimizing bandwidth among the requesters for resources, which include the following:

- EDMA3-initiated DMA transfers (and resulting coherency operations)
- DSP subsystem IDMA-initiated transfers (and resulting coherency operations)
- Programmable cache coherency operations
  - Block based coherency operations
  - Global coherency operations
- CPU direct-initiated transfers
  - Data access (load/store)
  - Program access

The resources include the following:

- L1P memory
- L1D memory
- L2 memory
- Resources outside of the C674x megamodule: external memory, on-chip peripherals, registers

Since any given requestor could potentially block a resource for extended periods of time, the bandwidth manager is implemented to assure fairness for all requesters.

The bandwidth manager implements a weighted-priority-driven bandwidth allocation. Each requestor (EDMA3, DSP subsystem IDMA, CPU, etc.) is assigned a priority level on a per-transfer basis. The programmable priority level has a single meaning throughout the system. There are a total of nine priority levels, where priority zero is the highest priority and priority eight is the lowest priority. When requests for a single resource contend, access is granted to the highest-priority requestor. When the contention occurs for multiple successive cycles, a contention counter assures that the lower-priority requestor gets access to the resource every 1 out of *n* arbitration cycles, where *n* is programmable. A priority level of -1 represents a transfer whose priority has been increased due to expiration of the contention counter or a transfer that is fixed as the highest-priority transfer to a given resource.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.3.2.2.4 Internal DMA (IDMA) Controller

The IDMA controller performs fast block transfers between any two memory locations local to the C674x megamodule. Local memory locations are defined as those in Level 1 program (L1P), Level 1 data (L1D), and Level 2 (L2) memories, or in the external peripheral configuration (CFG) port. The IDMA cannot transfer data to or from the internal DSP memory-mapped register space. The IDMA is fully described in the *TMS320C674x DSP Megamodule Reference Guide* (SPRUFK5).

### 1.3.3 Memory Map

Refer to your device-specific data manual for the addresses of the memory-map registers.

#### 1.3.3.1 DSP Internal Memory

See the *System Memory* chapter for a description of the DSP internal memory.

#### 1.3.3.2 External Memory

See the *System Interconnect* chapter and the *System Memory* chapter for a description of the additional memory and peripherals that the DSP has access to.

### 1.3.4 Advanced Event Triggering (AET)

The C674x megamodule supports advanced event triggering (AET). This capability can be used to debug complex problems as well as understand performance characteristics of user applications. AET provides the following capabilities:

- Hardware Program Breakpoints: specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- Data Watchpoints: specify data variable addresses, address ranges, or data values that can generate events such as halting the processor or triggering the trace capture.
- Counters: count the occurrence of an event or cycles for performance monitoring.
- State Sequencing: allows combinations of hardware program breakpoints and data watchpoints to precisely generate events for complex sequences.

## 1.4 HD Video Coprocessor Subsystem

### 1.4.1 HDVICP2 Overview

The HDVICP2 is the image and video imaging hardware accelerator subsystem.

The HDVICP2 supports resolutions up to 1080 p/i with full performance of 60 fps (or 120 fields). The HDVICP2 subsystem supports the following codec standards natively; that is, all functions of standards are accelerated (without any intervention of the digital signal processor [Media Controller]):

- H.264: BP/MP/HP Encode and Decode
- H.264: Fast Profile/RCDO Encode and Decode
- MPEG-4: SP/ASP Encode/Decode (No support for GMC)
- DivX 5.x & higher Encode/Decode (No lower version; for example, 3.11 and 4.x)
- H.263: Profile 0 and 3 for Decode, Profile 0 for Encode
- Sorenson Spark: V0 and V1 Decode (No encode support)
- MPEG-2 SP/MP Encode/Decode
- MPEG-1 Encode/Decode
- VC1/WMV9/RTV : SP/MP/AP Encode and Decode
- ON2 VP6/VP7 Decode
- RV 8/9/10 Decode
- AVS 1.0 Encode and Decode


Copyright © 2011–2016, Texas Instruments Incorporated

- JPEG (also MJPEG) Baseline Encode/Decode
- H264 Annex H (MVC)

The HDVICP2 subsystem is composed of:

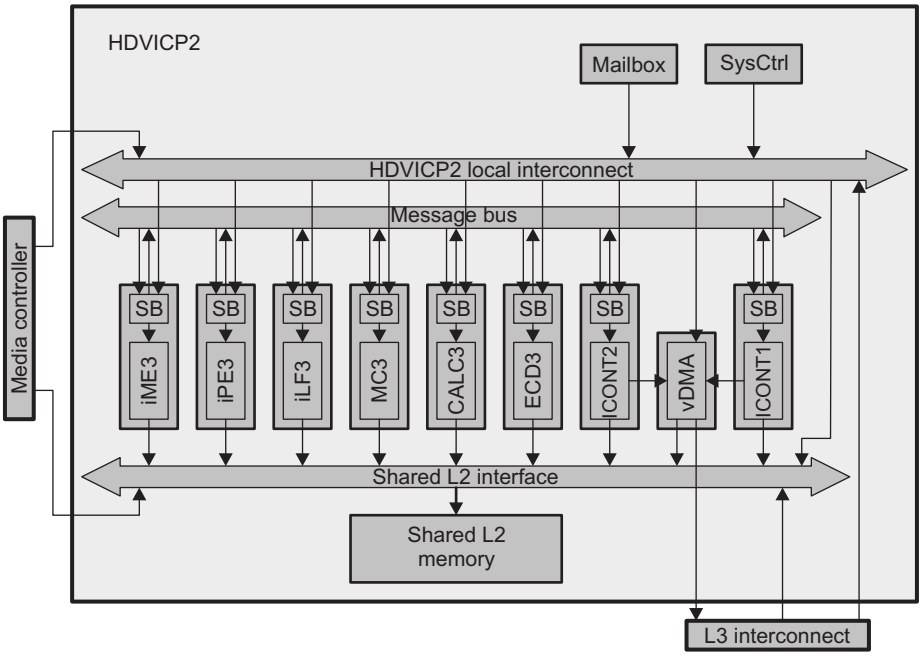
- A primary and a secondary sequencer: ICONT1 & ICONT2, which include its memories and an interrupt controller. Both these sequencers, ICONT1 and ICONT2, are identical.
- A video DMA engine: vDMA
- An entropy coder/decoder: ECD3
- A motion compensation engine: MC3
- A transform and quantization calculation engine: CALC3
- A loop filter acceleration engine: iLF3
- A motion estimation acceleration engine: iME3
- An intraprediction estimation engine: iPE3
- Shared level 2 (L2) interface and memory (of size 256 Kbytes)
- Local interconnect
- A message interface for communication between SyncBoxes
- Mailbox
- A debug module for trace event and software instrumentation: SMSET

TI has chosen eXpress DSP Digital Media (xDM) standard as the principle software interface to HDVICP2. The xDM standard defines application programming interfaces (APIs) through which an application invokes a particular class of codec, such as video, imaging, speech, and audio (aka VISA). The video and imaging classes for encode and decode apply to HDVICP2.

#### 1.4.1.1 HDVICP2 Functional Description

Figure 1-8 shows the block diagram of the HDVICP2 subsystem.

**Figure 1-8. HDVICP2 Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.4.1.2 SyncBox

The SyncBox is a configurable module that is in charge of scheduling all embedded hardware modules within the HDVICP2 subsystem. It handles all aspects of synchronization, data sharing, and parameters passing between accelerators. It also offers the possibilities to use asynchronous messages.

### *1.4.2 ICONTs*

The ICONT module is an ARM968E-S based microcontroller with 32KB of instruction tightly coupled memory (TCM) and 16KB of data tightly coupled memory. It includes an interrupt controller (INTC), a local data mover, its own SyncBox module for synchronizing tasks with other modules and its associated SyncBox handler.

Two identical instances of the ICONT, ICONT1 and ICONT2, are present in the HDVICP2 subsystem. They can typically be used to perform high-level processing (at frame and slice level), control at macroblock-level bounding box computation and other vDMA processing tasks.

Software can map processing equally on either ICONT.

### *1.4.3 vDMA*

The vDMA is a DMA engine to perform data transfer between external memories and shared L2 memory. The VDMA can also perform memory copies inside SL2 and inside external memory.

### *1.4.4 iME3*

The iME3 accelerator is used to perform motion estimation for Video encoding. iME3 has its own embedded SyncBox module for synchronizing tasks with other modules.

The iME3 compares a current macroblock to a reference area and provides an area in the reference region (in terms of offsets) which is least different to the current macroblock. It can also interpolate a block with half or quarter pixel precision, thus producing 4 (half pixel) or 16 (quarter pixel) interpolated blocks from the original block. Additionally, the IME3 supports searching of the best matching block within the interpolated planes.

### *1.4.5 iPE3*

The iPE3 accelerator is used to perform the intraprediction estimation for video encoding. The iPE3 has its own embedded SyncBox module for synchronizing tasks with other modules and an LSE to transfer data from internal memories to shared L2 memory.

The iPE3 supports two modes, depending on the video standard:

- Spatial intraprediction estimation for H264 and AVS. It creates intraprediction macroblocks with given intramodes from the original macroblock and provides a cost estimation between the original macroblock and each pseudo intraprediction macroblock, and then chooses a mode with the smallest cost to recommend it as an optimal intraprediction mode.
- Spatial activity for MPEG-1/2/4 and VC-1. It calculates the spatial activity of the original luminance samples with the specified block size. This mode is designed to provide information about the original Luma pixels. The values can be used to decide the coding parameters of the macroblock such as the coding mode and the quantization parameter.

### *1.4.6 MC3*

The MC3 accelerator is used to perform the motion compensation. The MC3 has its own embedded SyncBox module for synchronizing tasks with other HWAs and an LSE to transfer data from internal memories to shared L2 memory.

It creates an interprediction macroblock with given motion vectors and modes from the reference data.


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.4.7 CALC3

The CALC3 accelerator is used to perform forward and inverse transform and quantization calculation. CALC3 has its own embedded SyncBox module for synchronizing tasks with other HWAs and an LSE to transfer data from internal memories to shared L2 memory.

It can perform transform/inverse transform, Q/iQ and DC/AC prediction.

### 1.4.8 iLF3

The iLF3 accelerator is used to perform deblocking filtering and boundary strength computation. The iLF3 has its own embedded SyncBox module for synchronizing tasks with other HWAs.

### 1.4.9 ECD3

The ECD3 accelerator is designed to encode and decode the stream. ECD3 has its own embedded SyncBox module for synchronizing tasks with other HWAs and an LSE to transfer data from internal memories to shared L2 memory.

It supports Huffman codes and arithmetic codes.

For encode, the ECD3 encodes the macroblock information and residual data into a bitstream. For decode, the ECD3 decodes the bitstream and recovers the macroblock information and residual data.

### 1.4.10 SL2 Interface

The shared L2 interface, SL2IF, is an arbitrator that allows 18 initiators to access to an interleaved set of eight memory banks.

The SL2IF has two sets of interfaces:

- Eighteen 128-bit interfaces for accesses from module's to shared L2 memories.
- Eight 128-bit memory interfaces for direct accesses to the memory banks.

### 1.4.11 Message Bus

The message bus is an arbitrator that allows eight initiators to access eight targets. It is used to dispatch messages generated by the SyncBox of the different IPs.

### 1.4.12 HDVICP2 Local Interconnect

The HDVICP2 local interconnect provides connectivity between two external host interconnects (Media Controller and L3), two local sequencers (ICONT1 and ICONT2) and video hardware accelerators (iME3, iLF3, ECD3, CALC3, MC3, iPE3), video DMA engine (vDMA), and local modules (mailbox and sysctrl).

### 1.4.13 MailBox

The function of the mailbox is to support a 2-way communication between two hosts through interrupts. It allows the software to establish a communication channel between processors through a set of registers and associated interrupt signals by sending and receiving messages.

The mailbox embedded inside the HDVICP2 subsystem implements a 2-way communication between three external users and one internal user. This communication is ensured through three pairs of mailboxes and four-message FIFO depth for each message queue.

**NOTE:** The internal user is one of the two ICONTs. ICONT1 and ICONT2 are connected on a shared interrupt line. The choice between ICONT1 and ICONT2 is done by masking mailbox interrupt on ICONT1 or ICONT2. ICONT1 and ICONT2 can access the mailbox through the HDVICP2 local interconnect.

### 1.4.14 HDVICP2 System Control

The SYSCTRL module of HDVICP2 is in charge of:


Copyright © 2011–2016, Texas Instruments Incorporated

- Controlling clocks to the modules via software control and the hardware power handshaking state.
- Controlling power, reset, and clock management (PRCM) module power handshaking.
- Providing status of the above operation.
- Supporting synchronization through external event.


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.5 System MMU

### 1.5.1 MMU Overview

A memory management unit (MMU) is a hardware component responsible for handling accesses to memory requested by a processing unit, DMA controller, or other bus requestor. MMU functions include translation of virtual addresses to physical addresses (that is, virtual memory management).

The device contains the following MMUs:

- System MMU for: C674x™ (via its MDMA Port), EDMA Transfer Controller 0 (TPTC0), and EDMA Transfer Controller 1 (TPTC1).
- L1 Shared cache MMU and L2 MMU in Media Controller Subsystem
- ARM® Cortex™-A8 MMU
- SGX530™ MMU

This user guide provides a detailed description of System MMU and L2 MMU present in the Media Controller Subsystem. For more details on Cortex A8 MMU, see the ARM® Cortex™-A8 Core Technical Reference Manual (contact your TI representative). For more details on the L1 Shared cache MMU present in the Media Controller subsystem, please refer to the *Media Controller Subsystem* section.

### 1.5.2 MMU Integration

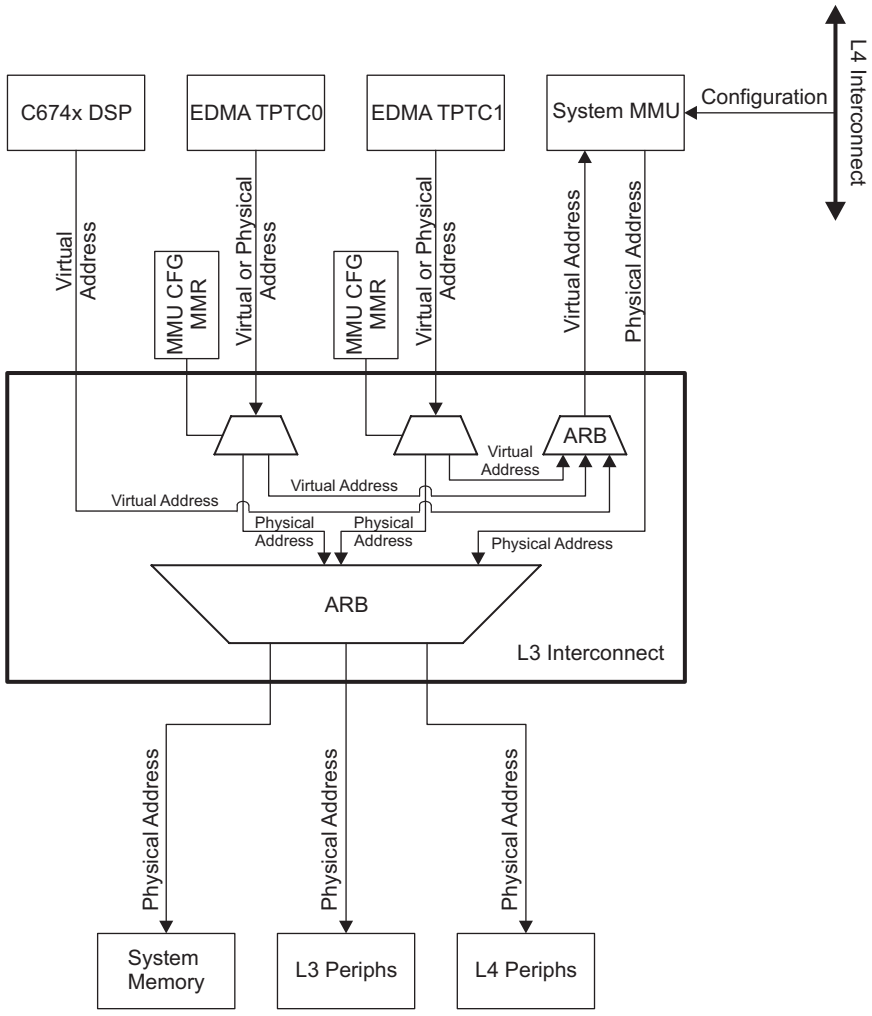
The MMU communicates accesses from the requestor (e.g., C674x MDMA bus requests) to the L3 main interconnect, performing virtual to physical address translation. The MMU will assert an interrupt when a translation fault occurs; or for other error conditions. For MMU interrupt details, refer to the device datasheet.

Figure 1-9 shows the System MMU integration. Requests initiated by the EDMA Transfer Controllers 0 and 1 (TPTC0, and TPTC1) can optionally be routed through the System MMU. These requestors' use (or not) of the MMU is independently controllable via the Control Module MMU_CFG register bitfields (TC0MMU and TC1MMU, respectively). The C674x DSP MDMA port accesses will always be routed through the MMU so no configuration bit is instantiated in the MMU_CFG register for C674x DSP. It is recommended that this register is set during system initialization and remain static.

If the MMU loopback path is enabled for a given requestor, requests will be routed via the L3 interconnect to the MMU and will again go through the L3 interconnect to the requested physical address. If the MMU loopback path is disabled for a given requestor, those bus requests go directly through the L3 interconnect to the requested physical address, thus minimizing bus request latency as illustrated in Figure 1-9.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-9. MMU Intergration**



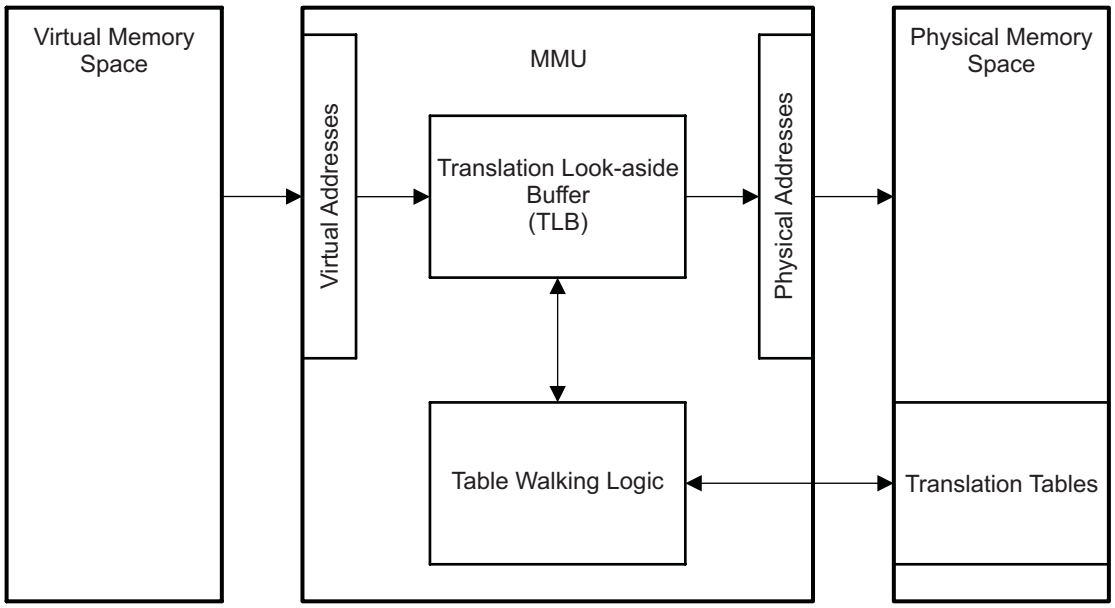

## *1.5.3 MMU Functional Description*

### 1.5.3.1 MMU Block Diagram

The MMU manages the virtual to physical address translation for external addresses. The MMU can be programmed through the L4 interconnect.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-10. MMU Block Diagram**




Each table entry describes the translation of one contiguous memory region. For a description of the structure of these tables, see Translation Tables.

Two major functional units exist in the MMU to provide address translation automatically based on the table entries:

- The table walker automatically retrieves the correct translation table entry for a requested translation. If two-level translation is used (for the translation of small memory pages), the table walker also automatically reads the required second-level translation table entry. The two-level translation is described later in the chapter.
- The translation look-aside buffer (TLB) stores recently used translation entries, acting like a cache of the translation table.

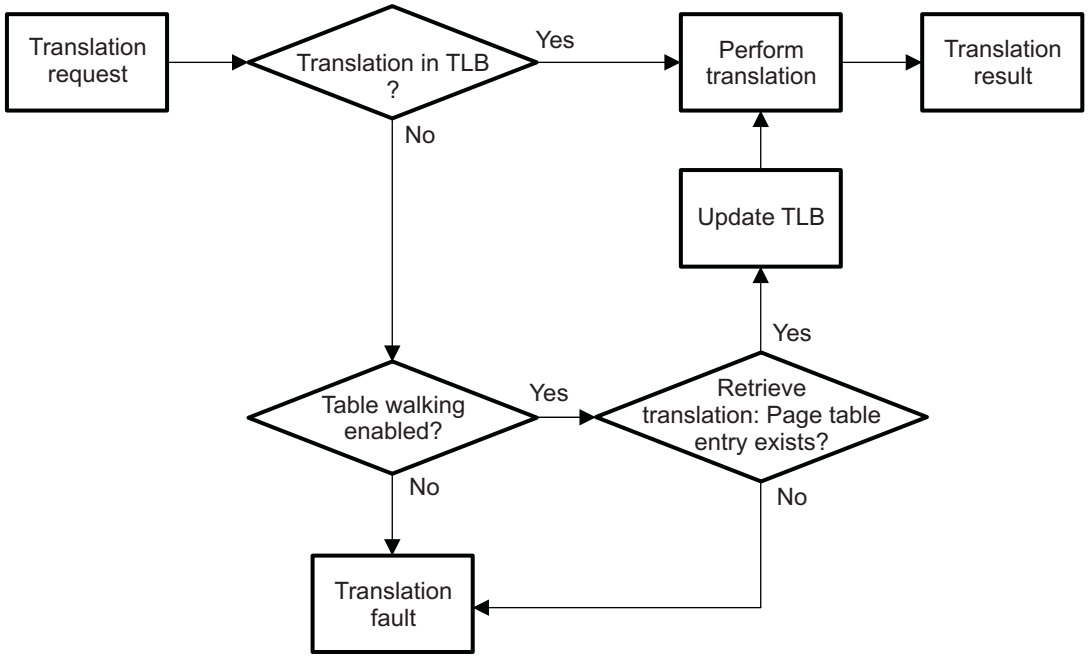
#### *1.5.3.1.1 MMU Address Translation Process*

Whenever an address translation is requested (that is, for every access with the MMU enabled), the MMU first checks whether the translation is contained in the TLB, which acts like a cache storing recent translations. The TLB can also be programmed manually to ensure that time-critical data can be translated without delay.

If the requested translation is not in the TLB, the table-walking logic retrieves this translation from the translation table(s), and then updates the TLB. The address translation is then performed. Figure 1-11 summarizes the process.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-11. MMU Address Translation Process**




#### 1.5.3.1.2 Translation Tables

The translation of virtual to physical addresses is based on entries in translation tables that define the following properties:

- Address translation, that is, the correspondence between virtual and physical addresses
- Size of the memory region the entry translates

The virtual addresses index the translation tables. Each virtual address corresponds to exactly one entry in the translation table.

##### 1.5.3.1.2.1 Translation Table Hierarchy

When developing a table-based address translation scheme, one of the most important design parameters is the memory page size described by each translation table entry. MMU instances support 4KB and 64KB pages, a 1MB section, and a 16MB supersection. Using bigger page sizes means a smaller translation table.

Using a smaller page size greatly increases the efficiency of dynamic memory allocation and defragmentation. That is why many operating systems (OSs) can operate on memory blocks as small as 4KB; however, the smaller size implies a more complex table structure.

A quick calculation shows that using 4KB memory pages with one translation table would require one million entries to span the entire 4GB address range. The table itself would be 32MB, a size that is not feasible.

However, using bigger pages reduces the flexibility of the typical OS memory management. Implementing a two-level hierarchy reconciles these two requirements. Within this hierarchy, one first-level translation table describes the translation properties based on 1MB memory regions.

Each of the entries in this first-level translation table can specify the following:

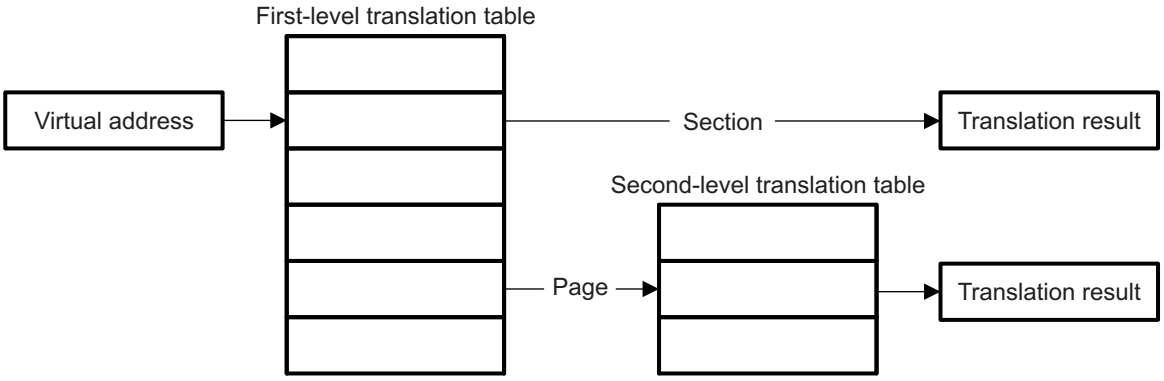
- The translation properties for a big memory section. This memory section can be either 1MB (section) or 16MB (supersection). In this case, all translation parameters are specified in the first-level translation table entry.
- A pointer to a second-level translation table that specifies individual translation properties based on


Copyright © 2011–2016, Texas Instruments Incorporated

smaller pages within the 1MB page of memory. These pages can be either 64KB (large page) or 4KB (small page). In this case, the actual translation parameters are specified in the second-level translation table entry. The first-level translation table entry specifies only the base address of the second-level translation table.

This hierarchical approach means that additional translation information for smaller pages must be provided only when the pages are actually used.

**Figure 1-12. Translation Hierarchy**




The structure of the first and second-level translation tables and their entries are described in more detail in First-Level Translation Table and Two-Level Translation.

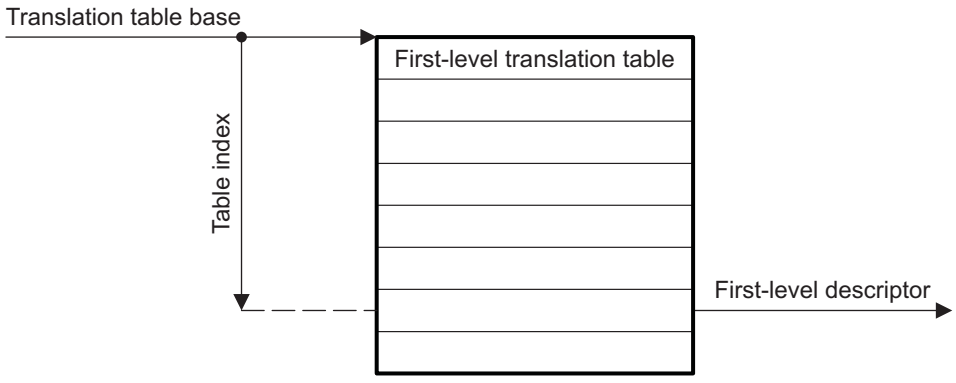
#### *1.5.3.1.2.2 First-Level Translation Table*

The first-level translation table describes the translation properties for 1MB sections. To describe a 4GB address range requires 4096 32-bit entries (so-called first-level descriptors).

The first-level translation table start address must be aligned on a multiple of the table size with a 128-byte minimum. Consequently, an alignment of at least 16K bytes is required for a complete 4096-entry table; that is, at least the last fourteen address bits must be zero.

The start address of the first-level translation table is specified by the so-called translation table base. The table is indexed by the upper 12-bits of the virtual address.

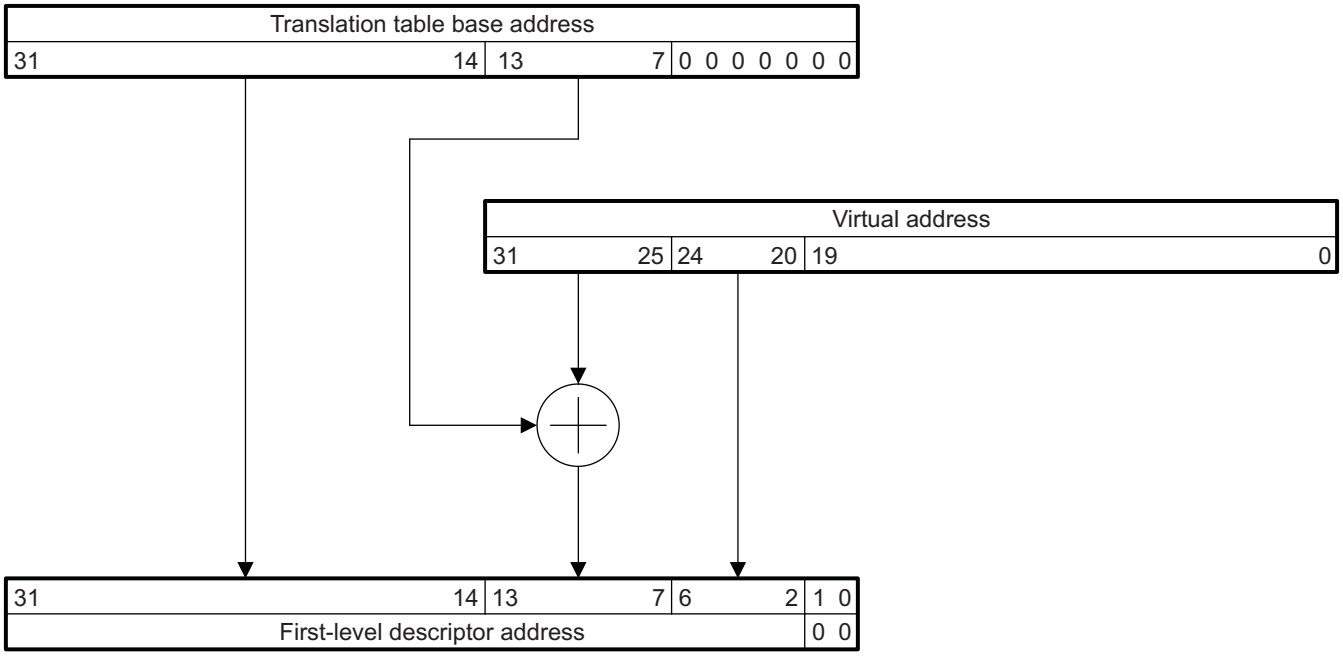
**Figure 1-13. First-Level Descriptor Address Calculation**




To summarize, the translation table base and the translation table index together define the first-level descriptor address. The precise mechanism used to calculate this address is shown below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-14. Detailed First-Level Descriptor Address Calculation**




As an example of this mechanism, consider a translation table base address of 0x8000:0000 and a virtual address of 0x1234:5678. In this case, the first-level descriptor address is:

0x8000:0000 + (0x123 <<2) = 0x8000:048C.

#### *1.5.3.1.2.3 First-Level Descriptor Format*

Each first-level descriptor provides either the complete address translation for 1MB or 16MB sections or provides a pointer to a second-level translation table for 4KB or 64KB pages.

**Table 1-42. First-Level Descriptor Format**

| 31:24 | 23:20 | 19 | 18 | 17 | 16 | 15 | 14:12 | 11:10 | 9:2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | | | | | | | | | | 0 | 0 | Fault |
| Second-level Translation Table Base Address | | | | | | | | | X | 0 | 1 | Page |
| Section Base Address | | X | 0 | M | X | E | X | ES | X | 1 | 0 | Section |
| Super Section base Address | | X | 1 | M | X | E | X | ES | X | 1 | 0 | Supersection |
| X | | | | | | | | | | 1 | 1 | Fault |

**M = Mixed region:** Don't care when E=0. Set to 0 to ensure future compatibility.

**E = Endianness:** Set to 0 for little-endian. Big-endian mode is not supported.

**ES = Element Size:** Don't care when E=0. Set to 00 to ensure future compatibility.

**X = Don't care.** Set to 0 to ensure future compatibility.

#### *1.5.3.1.2.4 First-Level Page Descriptor Format*

If a translation granularity smaller than 1MB is required, a two-level translation process is used. In this case, the first-level block descriptor specifies only the start address of a second-level translation table. The second-level translation table entries specify the actual translation properties.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.3.1.2.5 First-Level Section Descriptor Format

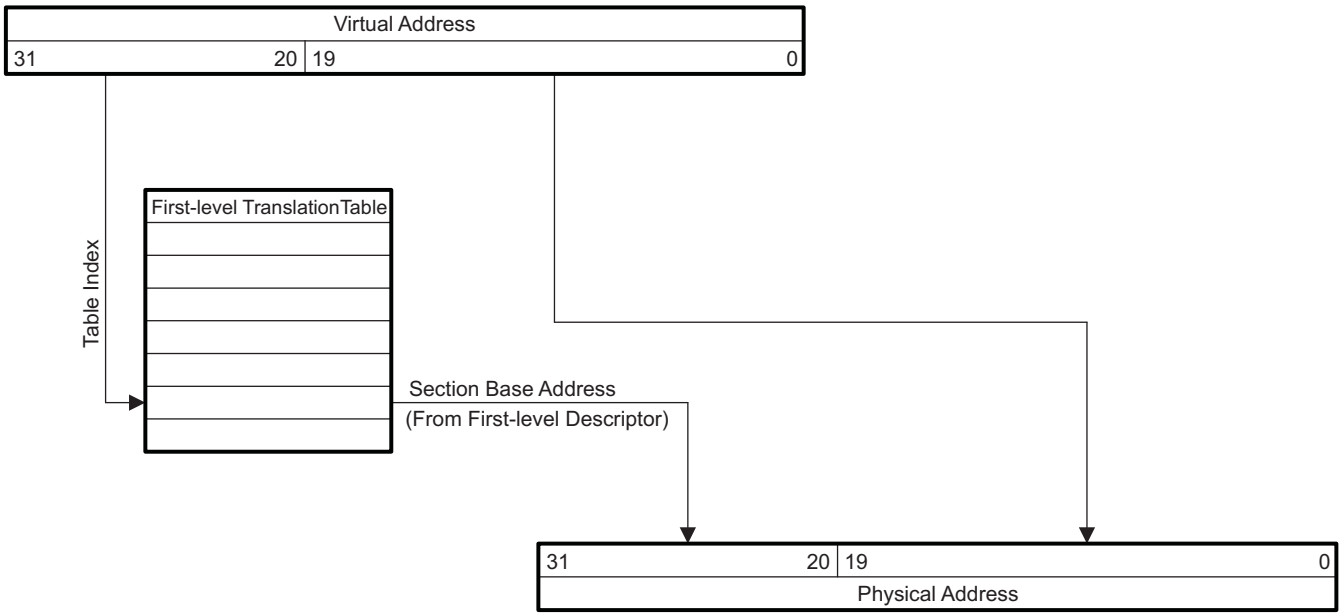
Each section descriptor in the first-level translation table specifies the complete translation properties for a 1MB section or a 16MB supersection.

**NOTE:** Supersection descriptors must be repeated 16 times, because each descriptor in the first-level translation table describes 1MB of memory. If an access points to a descriptor that is not initialized, the MMU will behave in an unpredictable way.

#### 1.5.3.1.2.6 Section Translation Summary

Sections and supersections can be translated based solely on the information in the first-level translation table.

**Figure 1-15. Section Translation Summary**



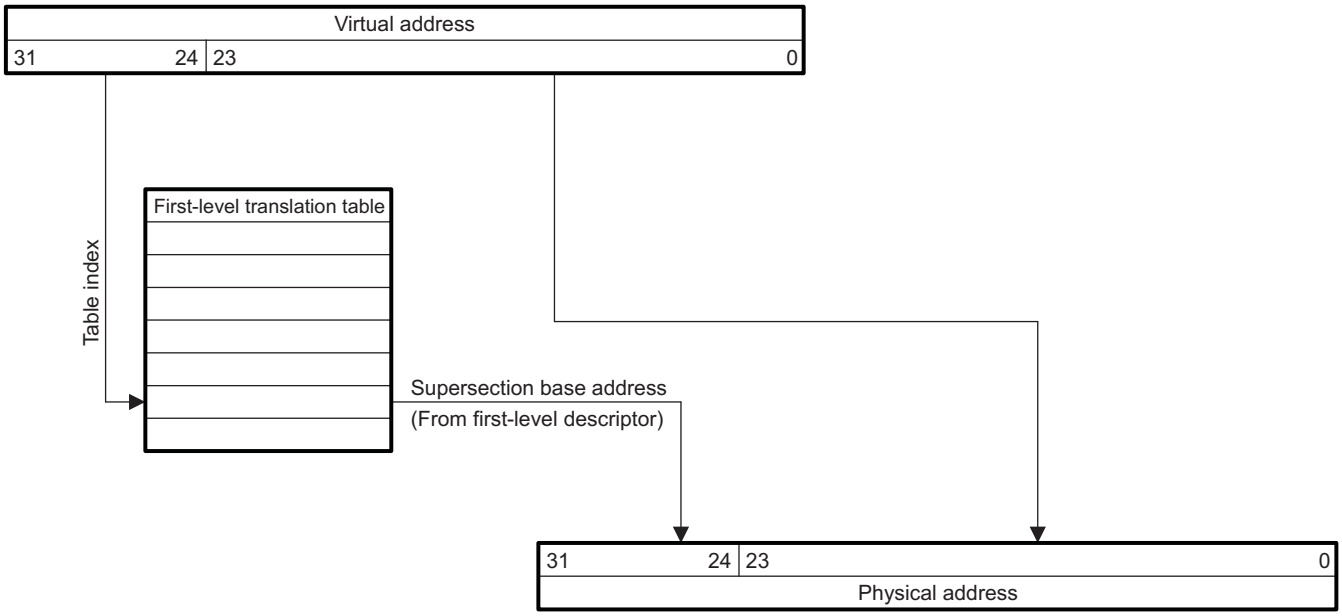


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.3.1.2.7 Supersection Translation Summary

The translation of a supersection is similar to the translation of a section. The difference is that for a supersection only bits 31 to 24 index into the first-level translation table. The last four bits of the table index are implicitly assumed to be zero as there are 16 identical consecutive entries for a supersection.

**Figure 1-16. Supersection Translation Summary**



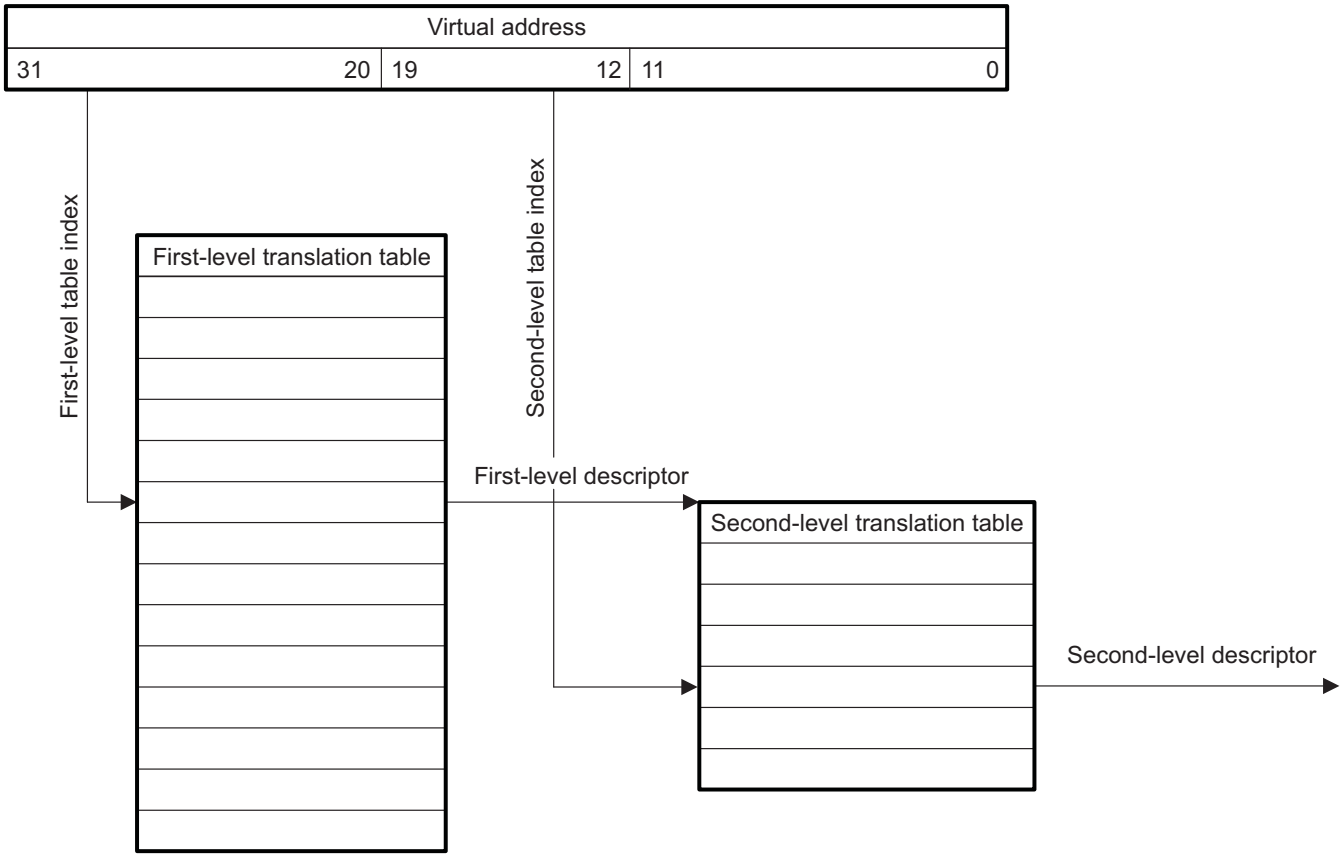


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.3.1.2.8 Two-Level Translation

Two-level translation is used when fine-grain granularity is required, that is, when memory sections smaller than 1MB are needed. In this case, the first-level descriptor provides a pointer to the base address of a second-level translation table. This second-level table is indexed by bits 19 to 12 of the virtual address.

**Figure 1-17. Two-Level Translation**




Each second-level translation table describes the translation of 1MB of address space in pages of 64KB (large page) or 4KB (small page). It consists of 256 second-level descriptors describing 4KB each.

**NOTE:** In the case of a large page, the same descriptor must be repeated 16 times. If an access points to a descriptor that is not initialized, the MMU will behave in an unpredictable way."


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.3.1.2.9 Second-Level Descriptor Format

Similar to first-level section descriptors, second-level descriptors provide all of the necessary information for the translation of a large or small page. The translation parameters (endianness, element size, and mixed region) have the same meaning as those for sections.

**Table 1-43. Second-Level Descriptor Format**

| 31:16 | 15:12 | 11 | 10 | 9 | 8:6 | 5:4 | 3:2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|
| X | | | | | | | | 0 | 0 | Fault |
| Large Page Base Address | X | M | X | E | X | ES | X | 0 | 1 | Large Page |
| Small Page Base Address | | M | X | E | X | ES | X | 1 | X | Small Page |

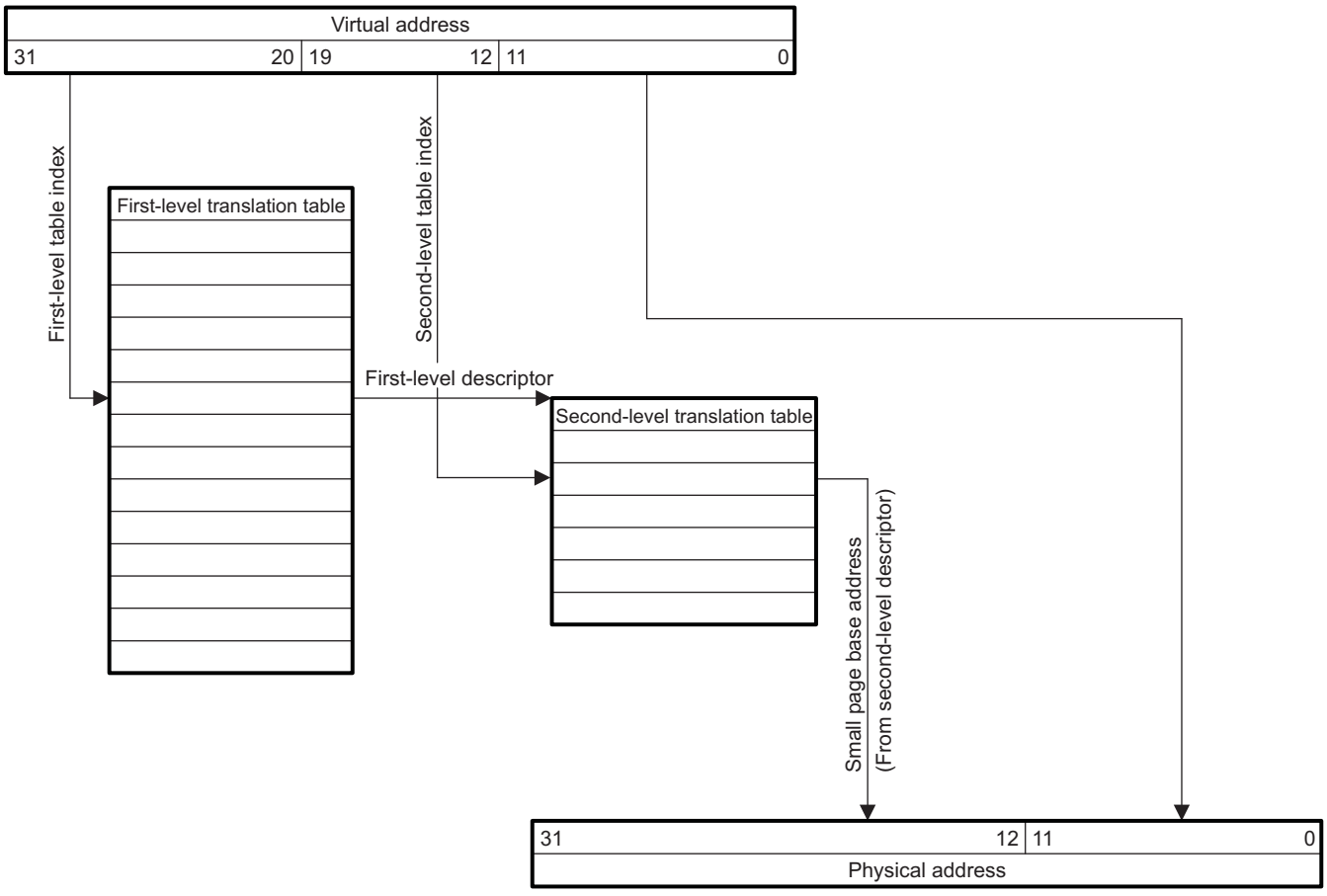
**M = Mixed region:** Don't care when E=0. Set to 0 to ensure future compatibility.

**E = Endianness:** Set to 0 for little-endian. Big-endian mode is not supported.

**ES = Element Size:** Don't care when E=0. Set to 00 to ensure future compatibility.

**X = Don't care.** Set to 0 to ensure future compatibility.

**Figure 1-18. Small Page Translation Summary**



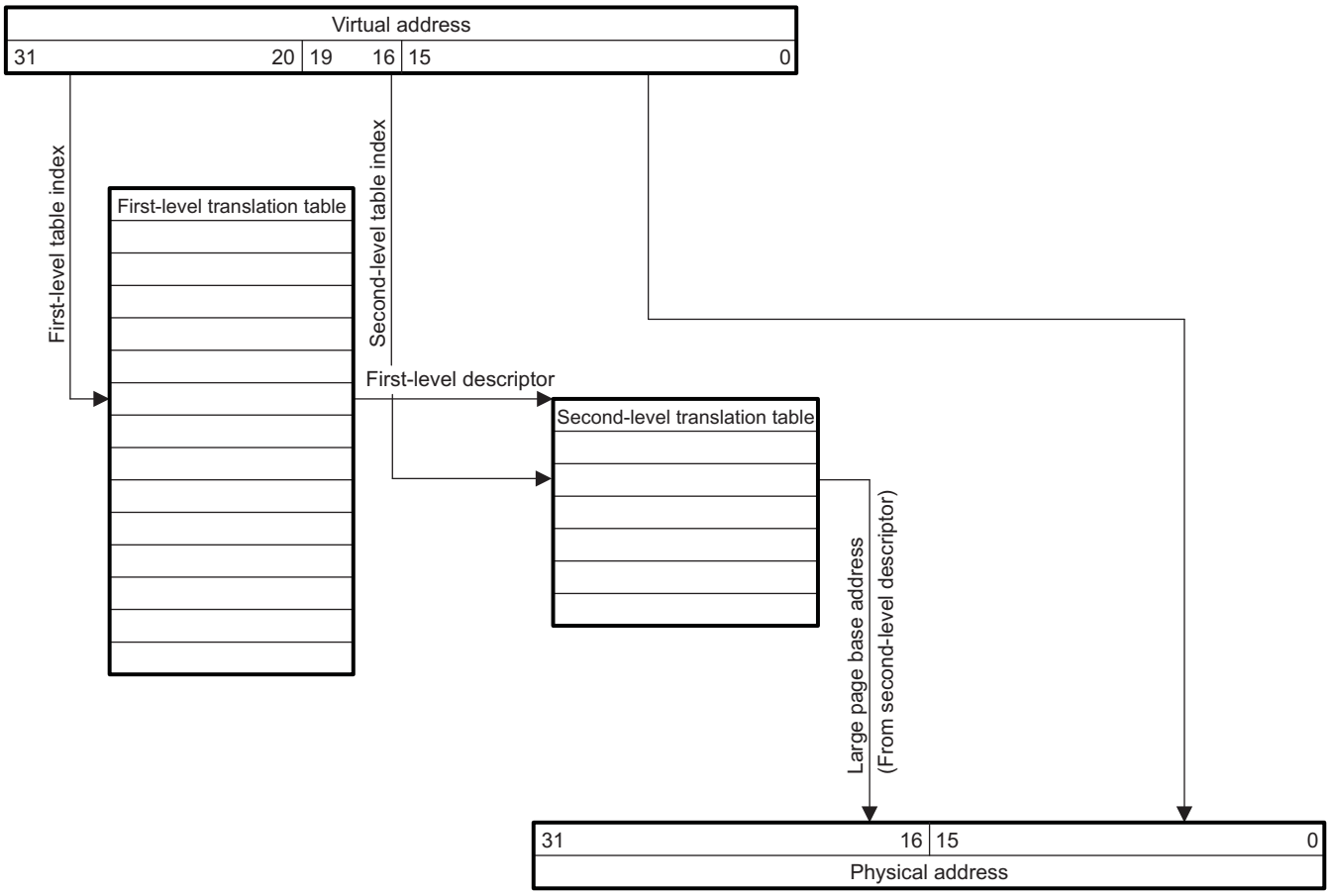


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.3.1.2.10 Large Page Translation Summary

The translation of a large page is similar to the translation of a small page. The difference is that, for a large page, only bits 19 to 16 index into the second-level translation table. The last four bits of the table index are implicitly assumed to be zero as there are 16 identical consecutive entries for a large page.

**Figure 1-19. Large Page Translation Summary**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.3.1.3 Translation Lookaside Buffer

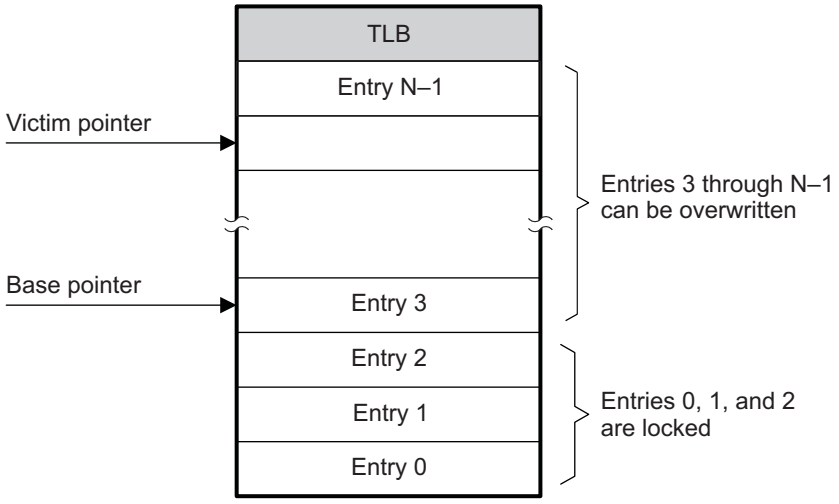
Translating virtual addresses to physical addresses is required for each memory access in systems using an MMU. To accelerate this translation process, a cache, or TLB, holds the result of recent translations.

For every translation, the MMU internal logic first checks whether the requested translation is already cached in the TLB. If the translation is cached, this translation is used; otherwise the translation is retrieved from the translation tables and the TLB is updated. If the TLB is full, one of its entries must be replaced. This entry is selected on a random basis.

The first n TLB entries, where n < Total Number N of TLB Entries , can be protected (locked) against being overwritten by setting the TLB base pointer to n . When this mechanism is used, only unprotected entries can be overwritten. The victim pointer indicates the next TLB entry to be written. Figure 1-20 shows an example of the TLB with N TLB entries (ranging from 0 to N-1). The base pointer contains the value "3" protecting Entry 0, Entry 1, and Entry 2 and the victim pointer points to the next TLB entry to be updated.

**NOTE:** The last TLB entry (Entry N-1) always remains unprotected.

**Figure 1-20. TLB-Entry Lock Mechanism**




The table walking logic automatically writes the TLB entries. The entries can also be manually written, which is done typically to ensure that the translation of time-critical data accesses is already present in the TLB so that they execute as fast as possible. The entries must be locked to prevent them from being overwritten.

##### 1.5.3.1.3.1 TLB Entry Format

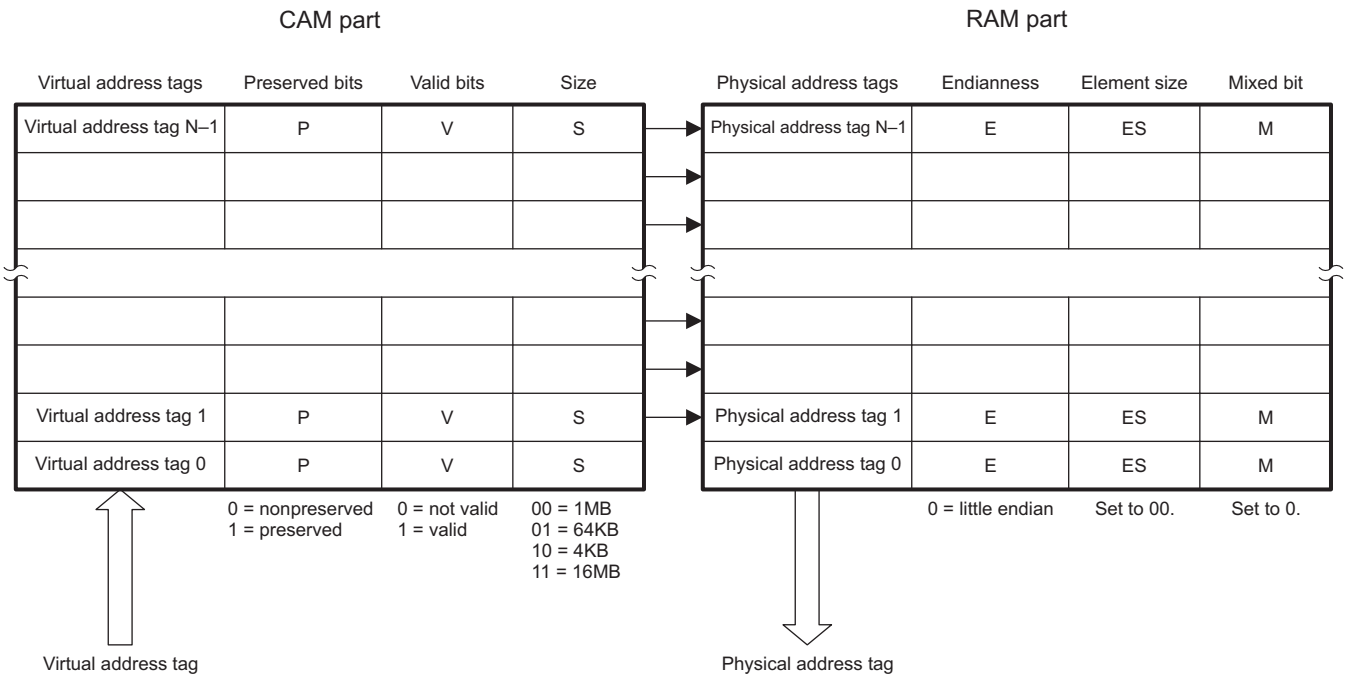
TLB entries consist of two parts:

- The CAM part contains the virtual address tag used to determine if a virtual address translation is in the TLB. The TLB acts like a fully associative cache addressed by the virtual address tag. The CAM part also contains the section/page size, as well as the preserved and the valid parameters. See the register table for more details.
- The RAM part contains the address translation that belongs to the virtual address tag as well as the endianness, element size, and mixed parameters described in First-Level Translation Table. See the MMU_RAM register table for more details.

The valid parameter specifies whether an entry is valid or not. The preserved parameter determines the behavior of an entry in the event of a TLB flush. If an entry is set as preserved, it is not deleted when a TLB is flushed, that is, when [0] GLOBALFLUSH is set to 1. Preserved entries must be deleted manually.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-21. TLB-Entry Structure**




### 1.5.3.2 MMU Clock Configuration

There are two clock domains: The functional clock domain for the MMU, which is synchronous to the clock for the interconnect slave and master access ports; and the clock domain for the interconnect slave configuration port. As these clocks are matched, there is a single input clock with enables for each of the clock domains. If a clock domain should run at the same frequency as the input clock, that enable can be tied high.

Two clock enable signals exist, one to enable the interconnect data master and slave ports, and the other to enable the clock on the configuration L3 interconnect port. The clock signals are configured through the MMU_SYSCONFIG register. This is a system configuration register that controls the various parameters of the L3 interface.

### 1.5.3.3 MMU Software Reset

This section describes the software reset feature of the module. The MMU instances are reset together with their respective reset domains. See the Clocks and Resets tables for information about the reset domains of the different MMU instances.

To perform a software reset, write 1 in the MMU_SYSCONFIG[1] SOFTRESET bit. The MMU_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the MMU_SYSCONFIG[1] SOFTRESET bit is automatically reset. The software must ensure that the software reset completes before doing MMU operations. When an MMU instance is released from reset, its TLB is empty and the MMU is disabled.

### 1.5.3.4 MMU Power Management

As part of the device system-wide power management scheme, each MMU instance supports a communication protocol with the PRCM module that allows the PRCM module to request an MMU instance to enter a low-power state. When the MMU instance acknowledges a low-power mode request from the PRCM module, the clock to the instance is gated off at the PRCM clock generator. Because the clock is disabled at the source, the low-power mode offers lower power consumption than the internal clock gating method in the local power management.


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.5.3.5 MMU Local Power Management Features

**Table 1-44. MMU Local Power Management Features**

| Feature | Register |
|---|---|
| Idle modes | MMU_SYSCONFIG[4:3] IDLEMODE |
| Clock activity | MMU_SYSCONFIG[9:8] CLOCKACTIVITY |
| Clock autogating | MMU_SYSCONFIG[0] AUTOIDLE |

**NOTE:** The MMU_SYSCONFIG[9:8] CLOCKACTIVITY bits are read only.

### 1.5.3.6 MMU Interrupt Requests

**Table 1-45. Events**

| Event Flag | Event Mask | Description |
|---|---|---|
| MMU_IRQSTATUS[4] MULTIHITFAULT | MMU_IRQENABLE[4] MULTIHITFAULT | Error due to multiple matches in the TLB |
| MMU_IRQSTATUS[3] TABLEWALKFAULT | MMU_IRQENABLE[3] TABLEWALKFAULT | Error due to error response received during a Table Walk |
| MMU_IRQSTATUS[2] EMUMISS | MMU_IRQENABLE[2] EMUMISS | |
| MMU_IRQSTATUS[1] TRANSLATIONFAULT | MMU_IRQENABLE[1] TRANSLATIONFAULT | Error due to invalid descriptor in the translation tables (translation fault) |
| MMU_IRQSTATUS[0] TLBMISS | MMU_IRQENABLE[0] TLBMISS | Error due to unrecoverable TLB miss (hardware TWL disabled) |

### 1.5.3.7 MMU Error Handling

**Table 1-46. Error Handling**

| Item | Condition | Action |
|---|---|---|
| 1 | Table-walk read has an error response. | Treat generally the same as a translation fault, but set the TableWalkFault interrupt status bit to aid in diagnosis |
| 2 | MMU is disabled during table-walk. | Not permitted; can result in loss of the current transaction but must not deadlock the MMU.<br>Avoid this condition by first disabling the table-walk logic and then polling the TWLRunning bit to ensure that no table walk is pending. |
| 3 | MMU is disabled during an address translation. | Not permitted; can result in access to an unintended location, but must not deadlock MMU.<br>This condition should be avoided by ensuring that no accesses are pending. |
| 4 | TLB is accessed during an address translation or a table walk. | Reading permitted; write should be done with care to ensure that the TLB is self-consistent at all times that a translation can occur. |
| 5 | TLB is flushed during address translation or a table walk. | Permitted; the flush is processed first, followed by the TWL update. |
| 6 | MMU is disabled while an interrupt is pending. | Not permitted; all pending interrupts should be processed before disabling the MMU. |


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.5.4 MMU Low-level Programming Models

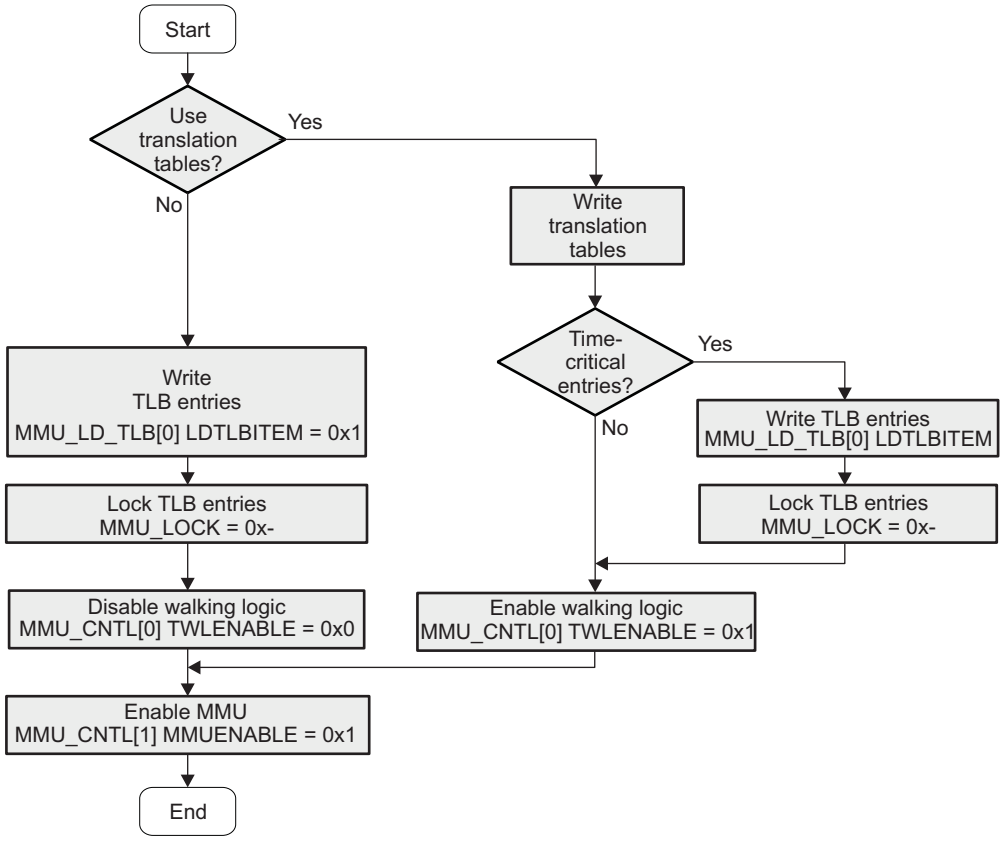
This section covers the low-level hardware programming sequences for configuration and usage of the module.

### 1.5.4.1 MMU Global Initialization

#### 1.5.4.1.1 Main Sequence - MMU Global Initialization

The System MMU can be enabled or disabled in MMU_CFG register (refer to the *Control Module* chapter). Figure 1-22 shows the procedure to initialize the MMU after a power-on or software reset.

**Figure 1-22. MMU Global Initialization**




#### 1.5.4.1.2 Subsequence - Configure a TLB entry

**Table 1-47. Configure a TLB Entry**

| Step | Registe/Bitfield/Programming Model | Value |
|---|---|---|
| Load the Virtual Address Tag | MMU_CAM[31:12] VATAG | 0x- |
| Protect the TLB entry against flush | MMU_CAM[3] P | 0x1 |
| Validate the TLB entry | MMU_CAM[2] V | 0x1 |
| Define the page size | MMU_CAM[1:0] PAGESIZE | 0x- |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.5.4.2 Operational Modes Configuration

#### *1.5.4.2.1 Main Sequence - Writing TLB Entries Statically*

Writing TLB entries statically avoids the need to write translation tables in memory and is commonly used for relatively small address spaces. This method ensures that the translation of time-critical data accesses execute as fast as possible with entries already present in the TLB. These entries must be locked to prevent them from being overwritten.

**Table 1-48. MMU Writing TLB Entries Statically**

| Step | Register/ Bitfield / Programming Model | Value |
|---|---|---|
| Execute software reset | MMU_SYSCONFIG[1] SOFTRESET | 0x1 |
| Wait for reset to complete | MMU_SYSSTATUS[0] RESETDONE | =0x1 |
| Enable power saving via automatic interface clock gating | MMU_SYSCONFIG[0] AUTOIDLE | 0x1 |
| Configure TLB entries | Refer to table Configure a TLB Entry | |
| Load the physical Address of the page | MMU_RAM[31:12] PHYSICALADDRESS | 0x- |
| Define the endianness of the page (little endian or big endian) | MMU_RAM[9] ENDIANNESS | 0x- |
| Select the element size | MMU_RAM[8:7] ELEMENTSIZE | 0x- |
| Define mixed page attribute | MMU_RAM[6] MIXED | 0x- |
| Specify the TLB entry you want to write | MMU_LOCK[8:4] CURRENTVICTIM | 0x- |
| Load the specified entry in the TLB | MMU_LD_TLB[0] LDTLBITEM | 0x1 |
| Enable multihit fault and TLB miss | MMU_IRQENABLE[4] MULTIHITFAULT | 0x1 |
| | MMU_IRQENABLE[0] TLBMISS | 0x1 |
| Enable memory translations | MMU_CNTL[1] MMUENABLE | 0x1 |

#### *1.5.4.2.2 Main Sequence - Protecting TLB Entries*

The first n TLB entries (with n < total number of TLB entries) can be protected from being overwritten with new translations. This is useful to ensure that certain commonly used or time-critical translations are always in the TLB and do not require retrieval using the table walking process.

**Table 1-49. Protecting TLB Entries**

| Step | Register/Bitfield/Programming Model | Value |
|---|---|---|
| Locks the TLB entries | MMU_LOCK[14:10] BASEVALUE | 0x- |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.4.2.3 Main Sequence - Deleting TLB Entries

Two mechanisms exist to delete TLB entries. All unpreserved TLB entries, that is, TLB entries written with the preserved bit set to zero, can be deleted by invoking a TLB flush. The preserved bit should only be used on protected TLB entries, as it does not prevent replacement by the table walking logic.

**Table 1-50. Deleting TLB Entries**

| Step | Registe/Bitfield/Programming Model | Value |
|---|---|---|
| Flush all nonprotected TLB entries | MMU_GFLUSH[0] GLOBALFLUSH | 0x1 |
| Flush all TLB entries specified by the CAM register | MMU_FLUSH_ENTRY[0] FLUSHENTRY | 0x1 |

#### 1.5.4.2.4 Main Sequence - Read TLB Entries

TLB entries can be read by the programmer to determine the TLB content at runtime.

**Table 1-51. Read TLB Entries**

| Step | Register/Bitfield/Programming Model | Value |
|---|---|---|
| Set the current victim pointer | MMU_LOCK[8:4] CURRENTVICTIM | 0x- |
| Read RAM parts of the TLB entry | MMU_READ_RAM | |
| Read CAM parts of the TLB entry | MMU_READ_CAM | |


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.5.5 MMU Registers

### 1.5.5.1 MMU Instance Summary

**Table 1-52. MMU Instance Summary**

| Module Name | Size |
|---|---|
| System MMU | 4KB |

### 1.5.5.2 MMU Registers

**Table 1-53. MMU Registers**

| Register Name | Type | Register Width (Bits) | Address Offset |
|---|---|---|---|
| MMU_REVISION | R | 32 | 00h |
| MMU_SYSCONFIG | RW | 32 | 10h |
| MMU_SYSSTATUS | R | 32 | 14h |
| MMU_IRQSTATUS | RW | 32 | 18h |
| MMU_IRQENABLE | RW | 32 | 1Ch |
| MMU_WALKING_ST | R | 32 | 40h |
| MMU_CNTL | RW | 32 | 44h |
| MMU_FAULT_AD | R | 32 | 48h |
| MMU_TTB | RW | 32 | 4Ch |
| MMU_LOCK | RW | 32 | 50h |
| MMU_LD_TLB | RW | 32 | 54h |
| MMU_CAM | RW | 32 | 58h |
| MMU_RAM | RW | 32 | 5Ch |
| MMU_GFLUSH | RW | 32 | 60h |
| MMU_FLUSH_ENTRY | RW | 32 | 64h |
| MMU_READ_CAM | R | 32 | 68h |
| MMU_READ_RAM | R | 32 | 6Ch |
| MMU_EMU_FAULT_AD | R | 32 | 70h |
| MMU_FAULT_PC | R | 32 | 80h |

#### 1.5.5.2.1 MMU_REVISION

The MMU_REVISION register is shown in Figure 1-23 and described in Table 1-54.

**Figure 1-23. MMU_REVISION**

| 31 | 0 |
|---|---|
| REVISION | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-54. MMU_REVISION Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | REVISION | 0 | IP Revision. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.2 MMU_SYSCONFIG

The MMU_SYSCONFIG register is shown in Figure 1-24 and described in Table 1-55.

**Figure 1-24. MMU_SYSCONFIG**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 | 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | CLOCKACTIVITY | | Reserved | | IDLEMODE | | Reserved | SOFTRESET | AUTOIDLE |
| R-0 | | R/W-0 | | R-0 | | R/W-0 | | R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-55. MMU_SYSCONFIG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Write 0's for future compatibility. Reads returns 0. |
| 9-8 | CLOCKACTIVITY | 0 | Clock activity during wake-up mode 00 Functional and Interconnect clocks can be switched off. |
| 7-5 | Reserved | 0 | Write 0's for future compatibility. Reads returns 0 |
| 4-3 | IDLEMODE | | Idle Mode |
| | | 0 | Force-idle. An idle request is acknowledged unconditionally. |
| | | 1h | No-idle. An idle request is never acknowledged. |
| | | 2h | Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module. |
| | | 3h | Reserved do not use. |
| 2 | Reserved | 0 | Write 0's for future compatibility. Reads returns 0. |
| 1 | SOFTRESET | 1-0 | Software reset. This bit is automatically reset by the hardware. During reads, it always return 0. |
| | | | Read 0x0: always return 0 |
| | | | Write 0x0: no functional effect |
| | | | Write 0x1: The module is reset |
| | | | Read 0x1: never happens |
| 0 | AUTOIDLE | | Internal interconnect clock gating strategy. |
| | | 0 | Interconnect clock is free-running. |
| | | 1 | Automatic interconnect clock gating strategy is applied, based on the interconnect interface activity. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.3 MMU_SYSSTATUS

The MMU_SYSSTATUS register is shown in Figure 1-25 and described in Table 1-56.

**Figure 1-25. MMU_SYSSTATUS**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | RESETDONE |
| R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-56. MMU_SYSSTATUS Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reads returns 0. |
| 0 | RESETDONE | | Internal reset monitoring. |
| | | 0 | Internal module reset in on-going. |
| | | 1 | Reset completed. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.4 MMU_IRQSTATUS

The MMU_IRQSTATUS register is shown in Figure 1-26 and described in Table 1-57.

**Figure 1-26. MMU_IRQSTATUS**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | MULTIHITFAULT | TABLEWALKFAULT | EMUMISS | TRANSLATIONFAULT | TLBMISS |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-57. MMU_IRQSTATUS Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Write 0's for future compatibility read returns 0. |
| 4 | MULTIHITFAULT | 1-0 | Error due to multiple matches in the TLB. |
| | | | Read 0x0: MultiHitFault false. |
| | | | Write 0x0: MultiHitFault status bit unchanged. |
| | | | Write 0x1: MultiHitFault status bit is reset. |
| | | | Read 0x1: MultiHitFault is true ("pending"). |
| 3 | TABLEWALKFAULT | 1-0 | Error response received during a Table Walk. |
| | | | Read 0x0: TableWalkFault false. |
| | | | Write 0x0: TableWalkFault status bit unchanged. |
| | | | Write 0x1: TableWalkFault status bit is reset. |
| | | | Read 0x1: TableWalkFault is true ("pending"). |
| 2 | EMUMISS | 1-0 | Unrecoverable TLB miss during debug (hardware TWL disabled). |
| | | | Read 0x0: EMUMiss false. |
| | | | Write 0x0: EMUMiss status bit unchanged. |
| | | | Write 0x1: EMUMiss status bit is reset. |
| | | | Read 0x1: EMUMiss is true ("pending"). |
| 1 | TRANSLATIONFAULT | 1-0 | Invalid descriptor in translation tables (translation fault). |
| | | | Read 0x0: TranslationFault false. |
| | | | Write 0x0: TranslationFault status bit unchanged. |
| | | | Write 0x1: TranslationFault status bit is reset. |
| | | | Read 0x1: TranslationFault is true ("pending"). |
| 0 | TLBMISS | 1-0 | Unrecoverable TLB miss (hardware TWL disabled). |
| | | | Read 0x0: TLBMiss false. |
| | | | Write 0x0: TLBMiss status bit unchanged. |
| | | | Write 0x1: TLBMiss status bit is reset. |
| | | | Read 0x1: TLBMiss is true ("pending"). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.5 MMU_IRQENABLE

The MMU_IRQENABLE register is shown in Figure 1-27 and described in Table 1-58.

**Figure 1-27. MMU_IRQENABLE**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | MULTIHITFAULT | TABLEWALKFAULT | EMUMISS | TRANSLATIONFAULT | TLBMISS |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-58. MMU_IRQENABLE Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Write 0's for future compatibility read returns 0. |
| 4 | MULTIHITFAULT | | Error due to multiple matches in the TLB. |
| | | 0 | MultiHitFault is masked. |
| | | 1 | MultiHitFault event generates an interrupt if occurs. |
| 3 | TABLEWALKFAULT | | Error response received during a Table Walk. |
| | | 0 | TableWalkFault is masked. |
| | | 1 | TableWalkFault event generates an interrupt if occurs. |
| 2 | EMUMISS | | Unrecoverable TLB miss during debug (hardware TWL disabled). |
| | | 0 | EMUMiss interrupt is masked. |
| | | 1 | EMUMiss event generates an interrupt when it occurs. |
| 1 | TRANSLATIONFAULT | | Invalid descriptor in translation tables (translation fault). |
| | | 0 | TranslationFault is masked. |
| | | 1 | TranslationFault event generates an interrupt if occurs. |
| 0 | TLBMISS | | Unrecoverable TLB miss (hardware TWL disabled). |
| | | 0 | TLBMiss interrupt is masked. |
| | | 1 | TLBMiss event generates an interrupt when if occurs. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.6 MMU_WALKING_ST

The MMU_WALKING_ST register is shown in Figure 1-28 and described in Table 1-59.

**Figure 1-28. MMU_WALKING_ST**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TWLRUNNING |
| R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-59. MMU_WALKING_ST Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reads return 0. |
| 0 | TWLRUNNING | | Table Walking Logic is running. |
| | | 0 | TWL Completed. |
| | | 1 | TWL Running. |

#### 1.5.5.2.7 MMU_CNTL

The MMU_CNTL register is shown in Figure 1-29 and described in Table 1-60.

**Figure 1-29. MMU_CNTL**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | EMUTLBUPDATE | TWLENABLE | MMUENABLE | Reserved |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-60. MMU_CNTL Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |
| 3 | EMUTLBUPDATE | | Enable TLB update on emulator table walk. |
| | | 0 | Emulator TLB update disabled. |
| | | 1 | Emulator TLB update enabled. |
| 2 | TWLENABLE | | Table Walking Logic enable. |
| | | 0 | TWL disabled. |
| | | 1 | TWL enabled. |
| 1 | MMUENABLE | | MMU enable. |
| | | 0 | MMU disabled. |
| | | 1 | MMU enabled. |
| 0 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.8 MMU_FAULT_AD

The MMU_FAULT_AD register is shown in Figure 1-30 and described in Table 1-61.

**Figure 1-30. MMU_FAULT_AD**

| 31 | 0 |
|---|---|
| FAULTADDRESS | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-61. MMU_FAULT_AD Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | FAULTADDRESS | 0-FFFF FFFFh | Virtual address of the access that generated a fault. |

#### 1.5.5.2.9 MMU_TTB

The MMU_TTB register is shown in Figure 1-31 and described in Table 1-62.

**Figure 1-31. MMU_TTB**

| 31 | 7 | 6 | 0 |
|---|---|---|---|
| TTBADDRESS | | Reserved | |
| R/W-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-62. MMU_TTB Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | TTBADDRESS | 0-3FF FFFFh | Translation Table Base Address. |
| 6-0 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |

#### 1.5.5.2.10 MMU_LOCK

The MMU_LOCK register is shown in Figure 1-32 and described in Table 1-63.

**Figure 1-32. MMU_LOCK**

| 31 | 15 | 14 | 10 | 9 | 8 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | BASEVALUE | | Reserved | CURRENTVICTIM | | Reserved | |
| R-0 | | R/W-0 | | R-0 | R/W-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-63. MMU_LOCK Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |
| 14-10 | BASEVALUE | 0-1Fh | Locked entries base value. |
| 9 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |
| 8-4 | CURRENTVICTIM | 0-1Fh | Current entry to be updated either by the TWL or by the SW Write value : TLB entry to be updated by software, Read value : TLB entry that will be updated by table walk logic. |
| 3-0 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.11 MMU_LD_TLB

The MMU_LD_TLB register is shown in Figure 1-33 and described in Table 1-64.

**Figure 1-33. MMU_LD_TLB**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | LDTLBITEM |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-64. MMU_LD_TLB Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |
| 0 | LDTLBITEM | 1-0 | Write (load) data in the TLB. |
| | | | Read 0x0: Always return 0. |
| | | | Write 0x0: No functional effect. |
| | | | Write 0x1: Load TLB data. |
| | | | Read 0x1: Never happens. |

#### 1.5.5.2.12 MMU_CAM

The MMU_CAM register is shown in Figure 1-34 and described in Table 1-65.

**Figure 1-34. MMU_CAM**

| 31 | 12 | 11 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VATAG | | Reserved | | P | V | PAGESIZE | |
| R/W-0 | | R-0 | | R/W-0 | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-65. MMU_CAM Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | VATAG | 0-F FFFFh | Virtual address tag. |
| 11-4 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |
| 3 | P | | Preserved bit. |
| | | 0 | TLB entry may be flushed. |
| | | 1 | TLB entry is protected against flush. |
| 2 | V | | Valid bit. |
| | | 0 | TLB entry is invalid. |
| | | 1 | TLB entry is valid. |
| 1-0 | PAGESIZE | | Page size. |
| | | 0 | Section (1MB). |
| | | 1 | Large page (64KB). |
| | | 2h | Small page (4KB). |
| | | 3h | Supersection (16MB). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.13 MMU_RAM

The MMU_RAM register is shown in Figure 1-35 and described in Table 1-66.

**Figure 1-35. MMU_RAM**

| 31 12 | 11 10 | 9 | 8 7 | 6 | 5 0 |
|---|---|---|---|---|---|
| PHYSICALADDRESS | Reserved | ENDIANNESS | ELEMENTSIZE | MIXED | Reserved |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-66. MMU_RAM Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | PHYSICALADDRESS | 0-F FFFFh | Physical address of the page. |
| 11-10 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |
| 9 | ENDIANNESS | | Endianness of the page. |
| | | 0 | Little Endian. |
| | | 1 | Reserved. |
| 8-7 | ELEMENTSIZE | | Element size of the page (8, 16, 32, no translation) |
| | | 0 | Value is don't care when ENDIANNESS is set to 0 (Little Endian). Should set to 0 to ensure future compatibility. |
| | | 1 | Though value is don't care, not recommended. Value should be set to 0x0 to ensure future compatibility. |
| | | 2h | Though value is don't care, not recommended. Value should be set to 0x0 to ensure future compatibility. |
| | | 3h | Though value is don't care, not recommended. Value should be set to 0x0 to ensure future compatibility. |
| 6 | MIXED | | Mixed page attribute (use CPU element size). |
| | | 0 | Value is don't care when ENDIANNESS is set to 0 (Little Endian). Should set to 0 to ensure future compatibility. |
| | | 1 | Though value is don't care, not recommended. Value should be set to 0 to ensure future compatibility. |
| 5-0 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |

#### 1.5.5.2.14 MMU_GFLUSH

The MMU_GFLUSH register is shown in Figure 1-36 and described in Table 1-67.

**Figure 1-36. MMU_GFLUSH**

| 31 1 | 0 |
|---|---|
| Reserved | GLOBALFLUSH |
| R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-67. MMU_GFLUSH Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |
| 0 | GLOBALFLUSH | 1-0 | Flush all the non-protected TLB entries when set. |
| | | | Read 0x0: always return 0. |
| | | | Write 0x0: no functional effect. |
| | | | Write 0x1: flush all the non-protected TLB entries. |
| | | | Read 0x1: never happens. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.15 MMU_FLUSH_ENTRY

The MMU_FLUSH_ENTRY register is shown in Figure 1-37 and described in Table 1-68.

**Figure 1-37. MMU_FLUSH_ENTRY**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | FLUSHENTRY |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-68. MMU_FLUSH_ENTRY Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Write 0's for future compatibility Reads return 0. |
| 0 | FLUSHENTRY | 1-0 | Flush the TLB entry pointed by the virtual address (VATag) in MMU_CAM register, even if this entry is set protected. |
| | | | Read 0x0: always return 0. |
| | | | Write 0x0: no functional effect. |
| | | | Write 0x1: flush all the TLB entries specified by the CAM register. |
| | | | Read 0x1: never happens. |

#### 1.5.5.2.16 MMU_READ_CAM

The MMU_READ_CAM register is shown in Figure 1-38 and described in Table 1-69.

**Figure 1-38. MMU_READ_CAM**

| 31 | 12 | 11 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VATAG | | Reserved | | P | V | PAGESIZE | |
| R-0 | | R-0 | | R-0 | R-0 | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-69. MMU_READ_CAM Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | VATAG | 0-F FFFFh | Virtual address tag. |
| 11-4 | Reserved | 0 | Reads return 0. |
| 3 | P | | Preserved bit. |
| | | 0 | TLB entry may be flushed. |
| | | 1 | TLB entry is protected against flush. |
| 2 | V | | Valid bit. |
| | | 0 | TLB entry is invalid. |
| | | 1 | TLB entry is valid. |
| 1-0 | PAGESIZE | | Page size. |
| | | 0 | Section (1MB). |
| | | 1 | Large page (64KB). |
| | | 2h | Small page (4KB). |
| | | 3h | Supersection (16MB). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.17 MMU_READ_RAM

The MMU_READ_RAM register is shown in Figure 1-39 and described in Table 1-70.

**Figure 1-39. MMU_READ_RAM**

| 31 12 | 11 10 | 9 | 8 7 | 6 | 5 0 |
|---|---|---|---|---|---|
| PHYSICALADDRESS | Reserved | ENDIANNESS | ELEMENTSIZE | MIXED | Reserved |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-70. MMU_READ_RAM Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | PHYSICALADDRESS | 0-F FFFFh | Physical address of the page. |
| 11-10 | Reserved | 0 | Reads return 0. |
| 9 | ENDIANNESS | | Endianness of the page. |
| | | 0 | Little Endian. |
| | | 1 | Big endian. |
| 8-7 | ELEMENTSIZE | | Element size of the page (8, 16, 32, no translation) |
| | | 0 | 8-bits |
| | | 1 | 16-bits |
| | | 2h | 32-bits |
| | | 3h | No translation |
| 6 | MIXED | | Mixed page attribute (use CPU element size). |
| | | 0 | Use TLB element size. |
| | | 1 | Use CPU element size. |
| 5-0 | Reserved | 0 | Reads return 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.18 MMU_EMU_FAULT_AD

The MMU_EMU_FAULT_AD register is shown in Figure 1-40 and described in Table 1-71.

**Figure 1-40. MMU_EMU_FAULT_AD**

| 31 | 0 |
|---|---|
| EMUFAULTADDRESS | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-71. MMU_EMU_FAULT_AD Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | EMUFAULTADDRESS | 0-FFFFFFFFh | Virtual address of the last emulator access that generated a fault. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.5.5.2.19 MMU_FAULT_PC

The MMU_FAULT_PC register is shown in Figure 1-41 and described in Table 1-72.

**Figure 1-41. MMU_FAULT_PC**

| 31 | 0 |
|---|---|
| PC | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-72. MMU_FAULT_PC Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | PC | 0-FFFF FFFFh | CPU program counter value where cause MMU fault. |
| | | | The address value is captured at MMU_EMU_FAULT_AD[31:0] EMUFAULTADDRESS bit field. |
| | | | Data-Read-access : corresponding PC. |
| | | | Data-write-access : not perfect accuracy due to Posted-write. |


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.6 SGX530 Graphics Subsystem

This chapter describes the 2D/3D graphics accelerator (SGX) for the device.

**NOTE:** The SGX subsystem is a Texas Instruments instantiation of the POWERVR® SGX530 core from Imagination Technologies Ltd.

This document contains materials that are ©2003-2007 Imagination Technologies Ltd.

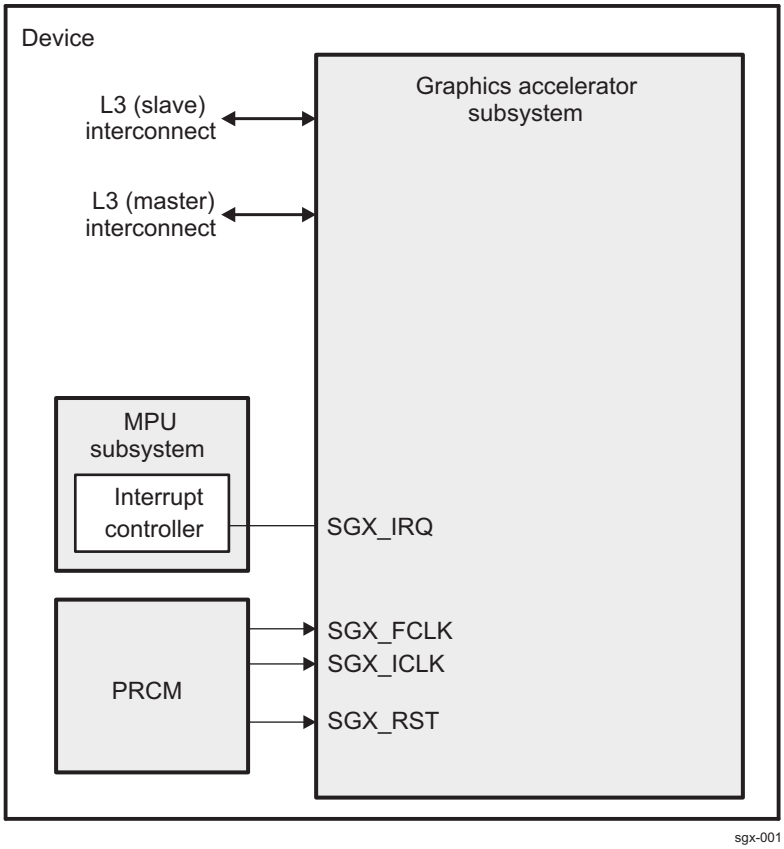
POWERVR® and USSE™ are trademarks or registered trademarks of Imagination Technologies Ltd.

### *1.6.1 SGX Overview*

The 2D/3D graphics accelerator (SGX) subsystem accelerates 2-dimensional (2D) and 3-dimensional (3D) graphics applications. The SGX subsystem is based on the POWERVR® SGX core from Imagination Technologies. SGX is a new generation of programmable POWERVR graphic cores. The POWERVR SGX530 v1.2.5 architecture is scalable and can target all market segments from mainstream mobile devices to high-end desktop graphics. Targeted applications include feature phone, PDA, and hand-held games.

Figure 1-42 shows the SGX subsystem in the device.

**Figure 1-42. Graphics Accelerator Highlight**




The SGX graphics accelerator can simultaneously process various multimedia data types:

- Pixel data
- Vertex data
- Video data
- General-purpose processing


Copyright © 2011–2016, Texas Instruments Incorporated

This is achieved through a multithreaded architecture using two levels of scheduling and data partitioning enabling zero-overhead task switching.

The SGX subsystem is connected to the L3 interconnect by a 128-bit master and a 32-bit slave interface.

#### 1.6.1.1 POWERVR SGX Main Features

- 2D graphics, 3D graphics, vector graphics, and programming support for GP-GPU functions
- Tile-based architecture
- Universal scalable shader engine (USSE™) – multithreaded engine incorporating pixel and vertex shader functionality
- Advanced shader feature set – in excess of Microsoft VS3.0, PS3.0, and OpenGL2.0
- Industry-standard API support – Direct3D Mobile, OpenGL ES 1.1 and 2.0, OpenVG v1.0.1
- Fine-grained task switching, load balancing, and power management
- Advanced geometry direct memory access (DMA) driven operation for minimum CPU interaction
- Programmable high-quality image anti-aliasing
- POWERVR SGX core MMU for address translation from the core virtual address to the external physical address (up to 4GB address range)
- Fully virtualized memory addressing for OS operation in a unified memory architecture
- Advanced and standard 2D operations [e.g., vector graphics, BLTs (block level transfers), ROPs (raster operations)]
- 32K stride support

#### 1.6.1.2 SGX 3D Features

- Deferred pixel shading
- On-chip tile floating point depth buffer
- 8-bit stencil with on-chip tile stencil buffer
- 8 parallel depth/stencil tests per clock
- Scissor test
- Texture support:
  - Cube map
  - Projected textures
  - 2D textures
  - Nonsquare textures
- Texture formats:
  - RGBA 8888, 565, 1555
  - Monochromatic 8, 16, 16f, 32f, 32int
  - Dual channel, 8:8, 16:16, 16f:16f
  - Compressed textures PVR-TC1, PVR-TC2, ETC1
  - Programmable support for all YUV formats
- Resolution support:
  - Frame buffer maximum size = 2048 x 2048
  - Texture maximum size = 2048 x 2048
- Texture filtering:
  - Bilinear, trilinear, anisotropic
  - Independent minimum and maximum control
- Antialiasing:
  - 4x multisampling
  - Up to 16x full scene anti-aliasing


Copyright © 2011–2016, Texas Instruments Incorporated

  - Programmable sample positions
- Indexed primitive list support
  - Bus mastered
- Programmable vertex DMA
- Render to texture:
  - Including twiddled formats
  - Auto MipMap generation
- Multiple on-chip render targets (MRT).
  **Note**: Performance is limited when the on-chip memory is not available.

#### 1.6.1.3 Universal Scalable Shader Engine (USSE) – Key Features

The USSE is the engine core of the POWERVR SGX architecture and supports a broad range of instructions.

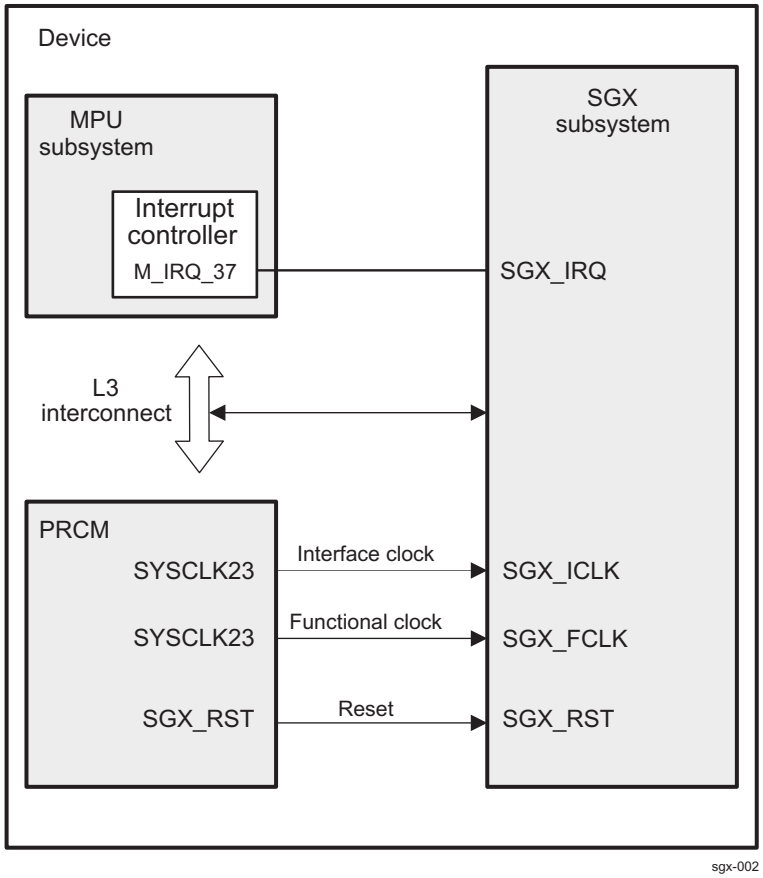
- Single programming model:
  - Multithreaded with 16 simultaneous execution threads and up to 64 simultaneous data instances
  - Zero-cost swapping in, and out, of threads
  - Cached program execution model
  - Dedicated pixel processing instructions
  - Dedicated video encode/decode instructions
- SIMD execution unit supporting operations in:
  - 32-bit IEEE float
  - 2-way 16-bit fixed point
  - 4-way 8-bit integer
  - 32-bit bit-wise (logical only)
- Static and dynamic flow control:
  - Subroutine calls
  - Loops
  - Conditional branches
  - Zero-cost instruction predication
- Procedural geometry:
  - Allows generation of primitives
  - Effective geometry compression
  - High-order surface support
- External data access:
  - Permits reads from main memory using cache
  - Permits writes to main memory
  - Data fence facility
  - Dependent texture reads


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.6.2 SGX Integration

Figure 1-43 highlights the SGX subsystem integration in the device.

**Figure 1-43. SGX Subsystem Integration**




#### 1.6.2.1 Clocking, Reset, and Power-Management Scheme

##### 1.6.2.1.1 Clocks

The SGX subsystem operates from two clocks: an interface clock (SGX_ICLK) and a functional clock (SYSCLK23). The power, reset, and clock management (PRCM) module generates and distributes both clocks inside the device.

**Table 1-73. Clock Descriptions**

| Functional Name | Signal Name | I/O[1] | Description |
|---|---|---|---|
| SGX_FCLK | SYSCLK23 | I | Functional clock → Functional clock domain |
| SGX_ICLK | SYSCLK23 | I | Interface clock → Interface clock domain |

(1) I =Input; O=Output

- SGX_ICLK interface clock manages the data transfer on the L3 master and slave ports.
- SGX_FCLK is the functional clock and is used inside the SGX subsystem to generate SGX 2D and 3D domain clock signals.

**To Enable the SGX Functional and Interface Clock**

Program the PRCM.CM_SGX_CLKSTCTRL.CTLKTRCTRL to 0x2
Program the PRCM.CM_SGX_SGX_CLKCTRL.MODULEMODE to 0x2
Poll for the PRCM.CM_SGX_SGX_CLKCTRL IDLEST to be 0x0


Copyright © 2011–2016, Texas Instruments Incorporated

**To Disable the SGX Functional and Interface Clock**

Program the PRCM.CM_SGX_SGX_CLKCTRL.MODULEMODE to 0x0
Program the PRCM.CM_SGX_CLKSTCTRL.CTLKTRCTRL to 0x0

The SGX_ICLK and SGX_FLCK are sourced from SYSCLK23 Depending on the DPLL_SGX and the PRCM.CM_SYSCLK23_CLKSEL[CLKSEL] settings, SGX can support a frequency up to 200 MHz. SGX530 Clocks can be divided from the original DPLL_SGX frequency. The divider can be programmed from /1../8 using the PRCM.CM_SYSCLK23_CLKSEL[CLKSEL] bits

#### *1.6.2.1.2 Resets*

The SGX subsystem has its own reset domain. Global reset of the SGX is performed by activating the SGX_RST signal in the SGX_RST domain. Software controls the release of SGX_RST using the PRCM.RM_SGX_RSTCTRL[0] SGX_RST bit.

#### *1.6.2.1.3 Power Management*

The SGX subsystem has its own power domain (SGX power domain). See the *Power, Reset, and Clock Management* chapter for additional information about the SGX power domain.

As described in the *SGX Integration* chapter, the SGX subsystem receives two clock signals from the PRCM module. The functional clock is used inside the SGX to generate clock signals to the multiple internal module SGX clock domains. The division ratio depends on the PRCM registers setting. For more information, see the *Power, Reset, and Clock Management* chapter.

Three power-management modes are defined:

- Deep power sleep (All clocks are gated.)
- Idle (2D and 3D clocks are gated.)
- 3D (No clock is gated.)

The SGX handles the automatic clock gating performed on the multiple internal module clock domains. SGX is in the same voltage domain as that of the Core logic. For more information refer to the Power, Reset, Clock Management chapter.

### 1.6.2.2 Hardware Requests

#### *1.6.2.2.1 Interrupt Request*

The SGX subsystem can generate one interrupt (SGX_IRQ) to the MPU subsystem interrupt controller mapped on M_IRQ_37.


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.6.3 SGX Functional Description

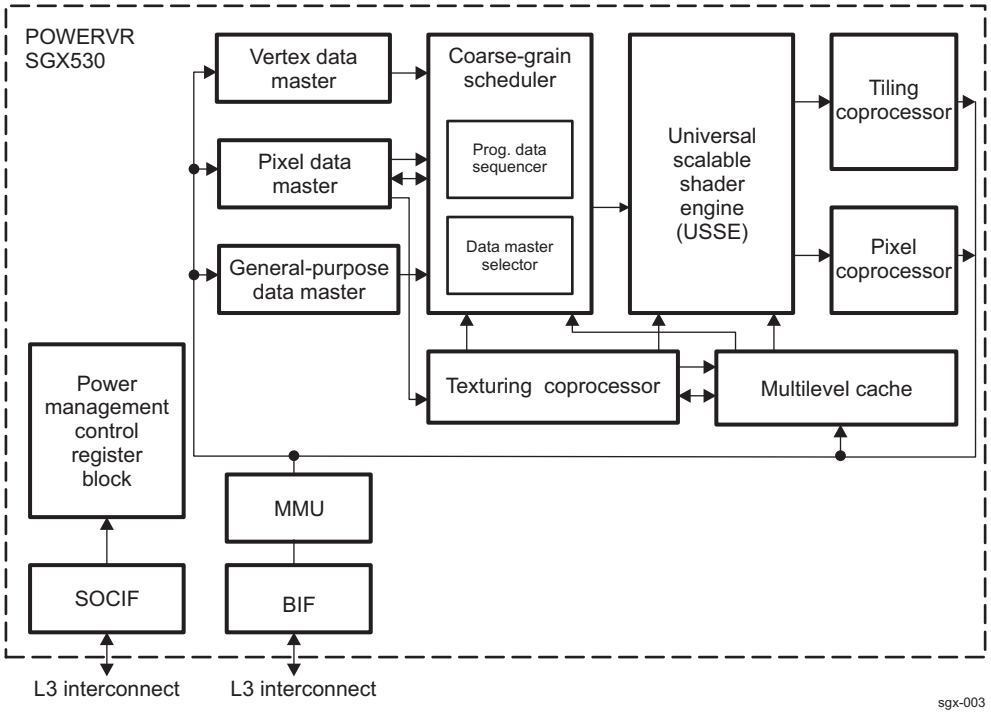
### 1.6.3.1 SGX Block Diagram

The SGX subsystem is based on the POWERVR® SGX530 core from Imagination Technologies. The architecture uses programmable and hard coded pipelines to perform various processing tasks required in 2D, 3D, and video processing. The SGX architecture comprises the following elements:

- Coarse grain scheduler
  - Programmable data sequencer (PDS)
  - Data master selector (DMS)
- Vertex data master (VDM)
- Pixel data master (PDM)
- General-purpose data master
- USSE
- Tiling coprocessor
- Pixel coprocessor
- Texturing coprocessor
- Multilevel cache

Figure 1-44 shows a block diagram of the SGX cores.

**Figure 1-44. SGX Block Diagram**




### 1.6.3.2 SGX Elements Description

The coarse grain scheduler (CGS) is the main system controller for the POWERVR SGX architecture. It consists of two stages, the DMS and the PDS. The DMS processes requests from the data masters and determines which tasks can be executed given the resource requirements. The PDS then controls the loading and processing of data on the USSE.

There are three data masters in the SGX core:

- The VDM is the initiator of transform and lighting processing within the system. The VDM reads an


Copyright © 2011–2016, Texas Instruments Incorporated

input control stream, which contains triangle index data and state data. The state data indicates the PDS program, size of the vertices, and the amount of USSE output buffer resource available to the VDM. The triangle data is parsed to determine unique indices that must be processed by the USSE. These are grouped together according to the configuration provided by the driver and presented to the DMS.

- The PDM is the initiator of rasterization processing within the system. Each pixel pipeline processes pixels for a different half of a given tile, which allows for optimum efficiency within each pipe due to locality of data. It determines the amount of resource required within the USSE for each task. It merges this with the state address and issues a request to the DMS for execution on the USSE.
- The general-purpose data master responds to events within the system (such as end of a pass of triangles from the ISP, end of a tile from the ISP, end of render, or parameter stream breakpoint event). Each event causes either an interrupt to the host or synchronized execution of a program on the PDS. The program may, or may not cause a subsequent task to be executed on the USSE.

The USSE is a user-programmable processing unit. Although general in nature, its instructions and features are optimized for three types of task: processing vertices (vertex shading), processing pixels (pixel shading), and video/imaging processing.

The multilevel cache is a 2-level cache consisting of two modules: the main cache and the mux/arbiter/demux/decompression unit (MADD). The MADD is a wrapper around the main cache module designed to manage and format requests to and from the cache, as well as providing Level 0 caching for texture and USSE requests. The MADD can accept requests from the PDS, USSE, and texture address generator modules. Arbitration, as well as any required texture decompression, are performed between the three data streams.

The texturing coprocessor performs texture address generation and formatting of texture data. It receives requests from either the iterators or USSE modules and translates these into requests in the multilevel cache. Data returned from the cache are then formatted according to the texture format selected, and sent to the USSE for pixel-shading operations.

To process pixels in a tiled manner, the screen is divided into tiles and arranged as groups of tiles by the tiling coprocessor. An inherent advantage of tiling architecture is that a large amount of vertex data can be rejected at this stage, thus reducing the memory storage requirements and the amount of pixel processing to be performed.

The pixel coprocessor is the final stage of the pixel-processing pipeline and controls the format of the final pixel data sent to the memory. It supplies the USSE with an address into the output buffer and then USSE returns the relevant pixel data. The address order is determined by the frame buffer mode. The pixel coprocessor contains a dithering and packing function.


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.6.4 SGX Register Manual

**CAUTION**

All SGX registers are limited to 32-bit data accesses, 8- and 16-bit accesses are not allowed because they can corrupt register content.

#### 1.6.4.1 SGX OCP Register Summary

**Table 1-74. SGX Registers Mapping Summary**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| FE00h | OCP_REVISION | OCP Revision Register | Section 1.6.4.1.1 |
| FE04h | OCP_HWINFO | Hardware Implementation Information Register | Section 1.6.4.1.2 |
| FE10h | OCP_SYSCONFIG | System Configuration Register | Section 1.6.4.1.3 |
| FE24h | OCP_IRQSTATUS_RAW_0 | Raw IRQ 0 Status Register | Section 1.6.4.1.4 |
| FE28h | OCP_IRQSTATUS_RAW_1 | Raw IRQ 1 Status Register | Section 1.6.4.1.5 |
| FE2Ch | OCP_IRQSTATUS_RAW_2 | Raw IRQ 2 Status Register | Section 1.6.4.1.6 |
| FE30h | OCP_IRQSTATUS_0 | Interrupt 0 Status Event Register | Section 1.6.4.1.7 |
| FE34h | OCP_IRQSTATUS_1 | Interrupt 1 Status Event Register | Section 1.6.4.1.8 |
| FE38h | OCP_IRQSTATUS_2 | Interrupt 2 Status Event Register | Section 1.6.4.1.9 |
| FE3Ch | OCP_IRQENABLE_SET_0 | Enable Interrupt 0 Register | Section 1.6.4.1.10 |
| FE40h | OCP_IRQENABLE_SET_1 | Enable Interrupt 1 Register | Section 1.6.4.1.11 |
| FE44h | OCP_IRQENABLE_SET_2 | Enable Interrupt 2 Register | Section 1.6.4.1.12 |
| FE48h | OCP_IRQENABLE_CLR_0 | Disable Interrupt 0 Register | Section 1.6.4.1.13 |
| FE4Ch | OCP_IRQENABLE_CLR_1 | Disable Interrupt 1 Register | Section 1.6.4.1.14 |
| FE50h | OCP_IRQENABLE_CLR_2 | Disable Interrupt 2 Register | Section 1.6.4.1.15 |
| FF00h | OCP_PAGE_CONFIG | Configure Memory Page Register | Section 1.6.4.1.16 |
| FF04h | OCP_INTERRUPT_EVENT | Interrupt Events Register | Section 1.6.4.1.17 |
| FF08h | OCP_DEBUG_CONFIG | Configuration of Debug Modes Register | Section 1.6.4.1.18 |
| FF0Ch | OCP_DEBUG_STATUS | Debug Status Register | Section 1.6.4.1.19 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.1 OCP Revision Register (OCP_REVISION)

The OCP Revision Register (OCP_REVISION) is shown in Figure 1-45 and described in Table 1-75.

**Figure 1-45. OCP Revision Register (OCP_REVISION)**

| 31 | 0 |
|---|---|
| REVISIONID | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-75. OCP Revision Register (OCP_REVISION) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | REVISIONID | 0-FFFF FFFFh | Revision value. |

#### 1.6.4.1.2 Hardware Implementation Information Register (OCP_HWINFO)

The Hardware Implementation Information Register (OCP_HWINFO) is shown in Figure 1-46 and described inTable 1-76.

**Figure 1-46. Hardware Implementation Information Register (OCP_HWINFO)**

| 31 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | MEM_BUS_WIDTH | SYS_BUS_WIDTH | |
| R-0 | | R-0 | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-76. Hardware Implementation Information Register (OCP_HWINFO) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved. |
| 2 | MEM_BUS_WIDTH | | Memory bus width. |
| | | 0 | Memory bus width is 64 bits. |
| | | 1 | Memory bus width is 128 bits |
| 1-0 | SYS_BUS_WIDTH | | System bus width. |
| | | 0 | System bus width is 32 bits. |
| | | 1 | System bus width is 64 bits. |
| | | 2h | System bus width is 128 bits. |
| | | 3h | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.3 System Configuration Register (OCP_SYSCONFIG)

The System Configuration Register (OCP_SYSCONFIG) is shown in Figure 1-47 and described in Table 1-77.

**Figure 1-47. System Configuration Register (OCP_SYSCONFIG)**

| 31 6 | 5 4 | 3 2 | 1 0 |
|---|---|---|---|
| Reserved | STANDBY_MODE | IDLE_MODE | Reserved |
| R-0 | R/W-2 | R/W-2 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-77. System Configuration Register (OCP_SYSCONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved. |
| 5-4 | STANDBY_MODE | | Clock standby mode. |
| | | 0 | Force Standby mode. |
| | | 1h | No Standby mode. |
| | | 2h | Smart Standby mode. |
| | | 3h | Smart Standby mode. |
| 3-2 | IDLE_MODE | | Clock Idle mode. |
| | | 0 | Force Idle mode. |
| | | 1h | No idle mode. |
| | | 2h | Smart Idle mode. |
| | | 3h | Smart Idle mode. |
| 1-0 | Reserved | 0 | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.4 Raw IRQ 0 Status Register (OCP_IRQSTATUS_RAW_0)

The Raw IRQ 0 Status Register (OCP_IRQSTATUS_RAW_0) is shown in Figure 1-48 and described in Table 1-78.

**Figure 1-48. Raw IRQ 0 Status Register (OCP_IRQSTATUS_RAW_0)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INIT_MINTERRUPT_RAW |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-78. Raw IRQ 0 Status Register (OCP_IRQSTATUS_RAW_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | INIT_MINTERRUPT_RAW | | Interrupt 0 - master port raw event. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: No action. |
| | | 1 | Write: Set event (Used for debug). |

#### 1.6.4.1.5 Raw IRQ 1 Status Register (OCP_IRQSTATUS_RAW_1)

The Raw IRQ 1 Status Register (OCP_IRQSTATUS_RAW_1) is shown in Figure 1-49 and described in Table 1-79.

**Figure 1-49. Raw IRQ 1 Status Register (OCP_IRQSTATUS_RAW_1)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TARGET_SINTERRUPT_RAW |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-79. Raw IRQ 1 Status Register (OCP_IRQSTATUS_RAW_1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | TARGET_SINTERRUPT_RAW | | Interrupt 1- slave port raw event. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: No action. |
| | | 1 | Write: Set event (Used for debug). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.6 Raw IRQ 2 Status Register (OCP_IRQSTATUS_RAW_2)

The Raw IRQ 2 Status Register (OCP_IRQSTATUS_RAW_2) is shown in Figure 1-50 and described in Table 1-80.

**Figure 1-50. Raw IRQ 2 Status Register (OCP_IRQSTATUS_RAW_2)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | THALIA_IRQ_RAW |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-80. Raw IRQ 2 Status Register (OCP_IRQSTATUS_RAW_2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | THALIA_IRQ_RAW | | Interrupt 2 - Thalia raw event. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: No action. |
| | | 1 | Write: Set event (Used for debug). |

#### 1.6.4.1.7 Interrupt 0 Status Event Register (OCP_IRQSTATUS_0)

The Interrupt 0 Status Event Register (OCP_IRQSTATUS_0) is shown in Figure 1-51 and described in Table 1-81.

**Figure 1-51. Interrupt 0 Status Event Register (OCP_IRQSTATUS_0)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INIT_MINTERRUPT_STATUS |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-81. Interrupt 0 Status Event Register (OCP_IRQSTATUS_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | INIT_MINTERRUPT_STATUS | | Interrupt 0 - Master port status event. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending and interrupt enabled. |
| | | 0 | Write: No action. |
| | | 1 | Write: Clear event. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.8 Interrupt 1 Status Event Register (OCP_IRQSTATUS_1)

The Interrupt 1 Status Event Register (OCP_IRQSTATUS_1) is shown in Figure 1-52 and described in Table 1-82.

**Figure 1-52. Interrupt 1 Status Event Register (OCP_IRQSTATUS_1)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TARGET_SINTERRUPT_STATUS |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-82. Interrupt 1 Status Event Register (OCP_IRQSTATUS_1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | TARGET_SINTERRUPT_STATUS | | Interrupt 1 - slave port status event. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending and interrupt enabled. |
| | | 0 | Write: No action. |
| | | 1 | Write: Clear event. |

#### 1.6.4.1.9 Interrupt 2 Status Event Register (OCP_IRQSTATUS_2)

The Interrupt 2 Status Event Register (OCP_IRQSTATUS_2) is shown in Figure 1-53 and described in Table 1-83.

**Figure 1-53. Interrupt 2 Status Event Register (OCP_IRQSTATUS_2)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TARGET_SINTERRUPT_STATUS |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-83. Interrupt 2 Status Event Register (OCP_IRQSTATUS_2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | TARGET_SINTERRUPT_STATUS | | Interrupt 1 - slave port status event. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending and interrupt enabled. |
| | | 0 | Write: No action. |
| | | 1 | Write: Clear event. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.10 Enable Interrupt 0 Register (OCP_IRQENABLE_SET_0)

The Enable Interrupt 0 Register (OCP_IRQENABLE_SET_0) is shown in Figure 1-54 and described in Table 1-84.

**Figure 1-54. Enable Interrupt 0 Register (OCP_IRQENABLE_SET_0)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INIT_MINTERRUPT_ENABLE |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-84. Enable Interrupt 0 Register (OCP_IRQENABLE_SET_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | INIT_MINTERRUPT_ENABLE | | Enable interrupt 0 - master port. |
| | | 0 | Read: Interrupt is enabled. |
| | | 1 | Read: Interrupt is disabled. |
| | | 0 | Write: No action. |
| | | 1 | Write: Enable interrupt. |

#### 1.6.4.1.11 Enable Interrupt 1 Register (OCP_IRQENABLE_SET_1)

The Enable Interrupt 1 Register (OCP_IRQENABLE_SET_1) is shown in Figure 1-55 and described in Table 1-85.

**Figure 1-55. Enable Interrupt 1 Register (OCP_IRQENABLE_SET_1)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TARGET_SINTERRUPT_ENABLE |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-85. Enable Interrupt 1 Register (OCP_IRQENABLE_SET_1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | TARGET_SINTERRUPT_ENABLE | | Enable interrupt 1 - slave port interrupt. |
| | | 0 | Read: Interrupt is enabled. |
| | | 1 | Read: Interrupt is disabled. |
| | | 0 | Write: No action. |
| | | 1 | Write: Enable interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.12 Enable Interrupt 2 Register (OCP_IRQENABLE_SET_2)

The Enable Interrupt 2 Register (OCP_IRQENABLE_SET_2) is shown in Figure 1-56 and described in Table 1-86.

**Figure 1-56. Enable Interrupt 2 Register (OCP_IRQENABLE_SET_2)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | THALIA_IRQ_ENABLE |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-86. Enable Interrupt 2 Register (OCP_IRQENABLE_SET_2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | THALIA_IRQ_ENABLE | | Enable interrupt 2 - Thalia (core) interrupt. |
| | | 0 | Read: Interrupt is enabled. |
| | | 1 | Read: Interrupt is disabled. |
| | | 0 | Write: No action. |
| | | 1 | Write: Enable interrupt. |

#### 1.6.4.1.13 Disable Interrupt 0 Register (OCP_IRQENABLE_CLR_0)

The Disable Interrupt 0 Register (OCP_IRQENABLE_CLR_0) is shown in Figure 1-57 and described in Table 1-87.

**Figure 1-57. Disable Interrupt 0 Register (OCP_IRQENABLE_CLR_0)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INIT_MINTERRUPT_DISABLE |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-87. Disable Interrupt 0 Register (OCP_IRQENABLE_CLR_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | INIT_MINTERRUPT_DISABLE | | Disable interrupt 0 - master port. |
| | | 0 | Read: Interrupt is enabled. |
| | | 1 | Read: Interrupt is disabled. |
| | | 0 | Write: No action. |
| | | 1 | Write: Disable interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.14 Disable Interrupt 1 Register (OCP_IRQENABLE_CLR_1)

The Disable Interrupt 1 Register (OCP_IRQENABLE_CLR_1) is shown in Figure 1-58 and described in Table 1-88.

**Figure 1-58. Disable Interrupt 1 Register (OCP_IRQENABLE_CLR_1)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TARGET_SINTERRUPT_DISABLE |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-88. Disable Interrupt 1 Register (OCP_IRQENABLE_CLR_1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | TARGET_SINTERRUPT_DISABLE | | Disable interrupt 1 - slave port. |
| | | 0 | Read: Interrupt is enabled. |
| | | 1 | Read: Interrupt is disabled. |
| | | 0 | Write: No action. |
| | | 1 | Write: Disable interrupt. |

#### 1.6.4.1.15 Disable Interrupt 2 Register (OCP_IRQENABLE_CLR_2)

The Disable Interrupt 2 Register (OCP_IRQENABLE_CLR_2) is shown in Figure 1-59 and described in Table 1-89.

**Figure 1-59. Disable Interrupt 2 Register (OCP_IRQENABLE_CLR_2)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | THALIA_IRQ_DISABLE |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-89. Disable Interrupt 2 Register (OCP_IRQENABLE_CLR_2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | THALIA_IRQ_DISABLE | | Disable interrupt 2 - Thalia (core) interrupt. |
| | | 0 | Read: Interrupt is enabled. |
| | | 1 | Read: Interrupt is disabled. |
| | | 0 | Write: No action. |
| | | 1 | Write: Disable interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.16 Configure Memory Page Register (OCP_PAGE_CONFIG)

The Configure Memory Page Register (OCP_PAGE_CONFIG) is shown in Figure 1-59 and described in Table 1-89.

**Figure 1-60. Configure Memory Page Register (OCP_PAGE_CONFIG)**

| 31 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | OCP_PAGE_SIZE | | MEM_PAGE_CHECK_EN | MEM_PAGE_SIZE | |
| R-0 | | R/W-2 | | R/W-1 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-90. Configure Memory Page Register (OCP_PAGE_CONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Reserved. |
| 4-3 | OCP_PAGE_SIZE | | Defines the page size on OCP memory interface. |
| | | 0 | Page size is 4 KB. |
| | | 1h | Page size is 2 KB. |
| | | 2h | Page size is 1 KB. |
| | | 3h | Page size is 520 B. |
| 2 | MEM_PAGE_CHECK_EN | | Enable page boundary checking. |
| | | 0 | Page boundary checking disabled. |
| | | 1 | Page boundary checking enabled. |
| 1-0 | MEM_PAGE_SIZE | | Defines the page size on internal memory interface. |
| | | 0 | Page size is 4 KB. |
| | | 1h | Page size is 2 KB. |
| | | 2h | Page size is 1 KB. |
| | | 3h | Page size is 520 B. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.17 Interrupt Events Register (OCP_INTERRUPT_EVENT)

The Interrupt Events Register (OCP_INTERRUPT_EVENT) is shown in Figure 1-61 and described in Table 1-91.

**Figure 1-61. Interrupt Events Register (OCP_INTERRUPT_EVENT)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | TARGET_INVALID_OCP_CMD | TARGET_CMD_FIFO_FULL | TARGET_RESP_FIFO_FULL |
| R-0 | | | | | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | INIT_MEM_REQ_FIFO_OVERRUN | INIT_READ_TAG_FIFO_OVERRUN | INIT_PAGE_CROSS_ERROR | INIT_RESP_ERROR | INIT_RESP_UNUSED_TAG | INIT_RESP_UNEXPECTED |
| | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-91. Interrupt Events Register (OCP_INTERRUPT_EVENT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Reserved | 0 | Reserved. |
| 10 | TARGET_INVALID_OCP_CMD | | Invalid command from OCP |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: Clear the event. |
| | | 1 | Write: Set event and interrupt if enabled (debug only). |
| 9 | TARGET_CMD_FIFO_FULL | | Command FIFO full. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: Clear the event. |
| | | 1 | Write: Set event and interrupt if enabled (debug only). |
| 7-6 | Reserved | 0 | Reserved. |
| 5 | INIT_MEM_REQ_FIFO_OVERRUN | | Memory request FIFO overrun. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: Clear the event. |
| | | 1 | Write: Set event and interrupt if enabled (debug only). |
| 4 | INIT_READ_TAG_FIFO_OVERRUN | | Read tag FIFO overrun. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: Clear the event. |
| | | 1 | Write: Set event and interrupt if enabled (debug only). |
| 3 | INIT_PAGE_CROSS_ERROR | | Memory page had been crossed during a burst. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: Clear the event. |
| | | 1 | Write: Set event and interrupt if enabled (debug only). |
| 2 | INIT_RESP_ERROR | | Receiving error response. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-91. Interrupt Events Register (OCP_INTERRUPT_EVENT) Field Descriptions (continued)**

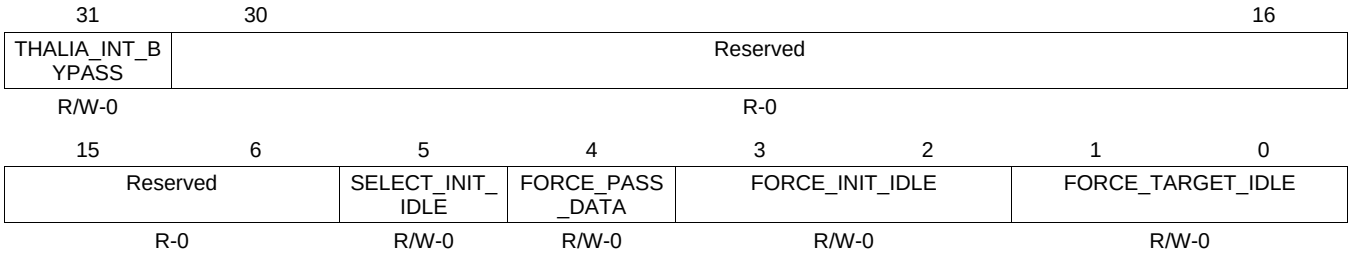
| Bit | Field | Value | Description |
|---|---|---|---|
| | | 0 | Write: Clear the event. |
| | | 1 | Write: Set event and interrupt if enabled (debug only). |
| 1 | INIT_RESP_UNUSED_TAG | | Receiving response on an unused tag. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: Clear the event. |
| | | 1 | Write: Set event and interrupt if enabled (debug only). |
| 0 | INIT_RESP_UNEXPECTED | | Receiving response when not expected. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: Clear the event. |
| | | 1 | Write: Set event and interrupt if enabled (debug only). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.18 Configuration of Debug Modes Register (OCP_DEBUG_CONFIG)

The Configuration of Debug Modes Register (OCP_DEBUG_CONFIG) is shown in Figure 1-62 and described in Table 1-92.

**Figure 1-62. Configuration of Debug Modes Register (OCP_DEBUG_CONFIG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-92. Configuration of Debug Modes Register (OCP_DEBUG_CONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | THALIA_INT_BYPASS | | Bypass OCP IPG interrupt logic. |
| | | 0 | Don't Bypass. |
| | | 1 | Bypass core interrupt to IO pin, ie disregard the interrupt enable setting in IPG register. |
| 30-6 | Reserved | 0 | Reserved. |
| 5 | SELECT_INIT_IDLE | | To select which idle the disconnect protocol should act on 0. |
| | | 0 | Whole SGX Idle. |
| | | 1 | OCP initiator idle only |
| 4 | FORCE_PASS_DATA | | Forces the initiator to pass data independent of disconnect protocol. |
| | | 0 | Normal mode. Don't force. |
| | | 1 | Never fence request to OCP. |
| 3-2 | FORCE_INIT_IDLE | | Forces the OCP master port to Idle. |
| | | 0 | Normal mode. No force. |
| | | 1h | Force port to be always Idle. |
| | | 2h | Forces target port to never be in Idle mode. |
| | | 3h | Normal mode. No force. |
| 1-0 | FORCE_TARGET_IDLE | | Forces the OCP target port to Idle. |
| | | 0 | Normal mode. No force. |
| | | 1h | Force port to be always Idle. |
| | | 2h | Forces target port to never be in Idle mode. |
| | | 3h | Normal mode. No force. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.6.4.1.19 Debug Status Register (OCP_DEBUG_STATUS)

The Debug Status Register (OCP_DEBUG_STATUS) is shown in Figure 1-63 and described in Table 1-93.

**Figure 1-63. Debug Status Register (OCP_DEBUG_STATUS)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CMD_DEBUG_STATE | CMD_RESP_DEBUG_STATE | TARGET_IDLE | RESP_FIFO_FULL | CMD_FIFO_FULL | RESP_ERROR | WHICH_TARGET_REGISTER | |
| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R/W-0 | |

| 23 | 21 | 20 | 18 | 17 | 16 |
|---|---|---|---|---|---|
| WHICH_TARGET_REGISTER | | TARGET_CMD_OUT | | INIT_MSTANDBY | INIT_MWAIT |
| R/W-0 | | R-0 | | R-0 | R-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| INIT_MDISCREQ | | INIT_MDISCACK | INIT_SCONNECT2 | INIT_SCONNECT1 | INIT_SCONNECT0 | INIT_MCONNECT | |
| R-0 | | R/W-0 | R-0 | R-0 | R-0 | R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TARGET_SIDLEACK | | TARGET_SDISCACK | | TARGET_SIDLEREQ | TARGET_SCONNECT | TARGET_MCONNECT | |
| R-0 | | R-0 | | R-0 | R-0 | | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-93. Debug Status Register (OCP_DEBUG_STATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | CMD_DEBUG_STATE | | Target command state machine |
| | | 0 | Idle. |
| | | 1 | Accept command. |
| 30 | CMD_RESP_DEBUG_STATE | | Target response state machine. |
| | | 0 | Send accept. |
| | | 1 | Wait accept. |
| 29 | TARGET_IDLE | 0-1 | Target idle. |
| 28 | RESP_FIFO_FULL | 0-1 | Target response FIFO full. |
| 27 | CMD_FIFO_FULL | 0-1 | Target command FIFO full. |
| 26 | RESP_ERROR | 0-1 | Respond to OCP with error, which could be caused by either address misalignment or invalid byte enable. |
| 25-21 | WHICH_TARGET_REGISTER | 0-1Fh | Indicates which OCP target registers to read. |
| 20-18 | TARGET_CMD_OUT | | Command received from OCP. |
| | | 0 | Command WRSYS received. |
| | | 1h | Command RDSYS received. |
| | | 2h | Command WR_ERROR received. |
| | | 3h | Command RD_ERROR received. |
| | | 4h | Command CHK_WRADDR_PAGE received. Not used. |
| | | 5h | Command CHK_RDADDR_PAGE received. Not used. |
| | | 6h | Command TARGET_REG_WRITE received. |
| | | 7h | Command TARGET_REG_READ received. |
| 17 | INIT_MSTANDBY | 1-0 | Status of init_MStandby signal. |
| 16 | INIT_MWAIT | 1-0 | Status of init_MWait signal. |
| 15-14 | INIT_MDISCREQ | | Disconnect status of the OCP interface. |
| | | 0 | State is FUNCT. |
| | | 1h | State is SLEEP TRANS |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-93. Debug Status Register (OCP_DEBUG_STATUS) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| | | 2h | Reserved. |
| | | 3h | State is IDLE. |
| 13 | INIT_MDISCACK | | Memory request FIFO full. |
| | | 0 | Read: No event pending. |
| | | 1 | Read: Event pending. |
| | | 0 | Write: Clear the event. |
| | | 1 | Write: Set the event and interrupt if enabled (debug only) |
| 12 | INIT_SCONNECT2 | | Defines whether to wait in M_WAIT state for MConnect FSM. |
| | | 0 | Skip M_WAIT state. |
| | | 1 | Wait in M_WAIT state. |
| 11 | INIT_SCONNECT1 | | Defines the busy-ness state of the slave. |
| | | 0 | Slave is drained. |
| | | 1 | Slave is loaded. |
| 10 | INIT_SCONNECT0 | | Disconnect from slave. |
| | | 0 | Disconnect request from slave. |
| | | 1 | Connect request from slave. |
| 9-8 | INIT_MCONNECT | | Initiator MConnect state. |
| | | 0 | State is M_OFF. |
| | | 1h | State is M_WAIT. |
| | | 2h | State is M_DISC. |
| | | 3h | State is M_CON. |
| 7-6 | TARGET_SIDLEACK | | Acknowledge the SIdleAck state machine. |
| | | 0 | State is FUNCT. |
| | | 1h | State is SLEEP TRANS. |
| | | 2h | Reserved. |
| | | 3h | State is IDLE. |
| 5-4 | TARGET_SDISCACK | | Acknowledge the SDiscAck state machine. |
| | | 0 | State is FUNCT. |
| | | 1h | State is SLEEP TRANS. |
| | | 2h | Reserved. |
| | | 3h | State is IDLE. |
| 3 | TARGET_SIDLEREQ | | Request the target to go idle. |
| | | 0 | Don't go idle, or go active. |
| | | 1 | Go idle. |
| 2 | TARGET_SCONNECT | | Target SConnect state. |
| | | 0 | Disconnect interface. |
| | | 1 | Connect OCP interface. |
| 1-0 | TARGET_MCONNECT | | Target MConnect state. |
| | | 0 | Target is in M_OFF state. |
| | | 1h | Target is in M_WAIT disconnect state. |
| | | 2h | Target is in M_DISC state. |
| | | 3h | Target is in M_CON state. |


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.7 Device Interrupts and EDMA Events

### 1.7.1 Interrupt Controllers

The device has a large number of interrupts to service the needs of its many peripherals and subsystems. The ARM Cortex-A8, C674x DSP, and Media Controller are capable of servicing these interrupts via their integrated interrupt controllers. In addition, the C674x DSP and Media Controller interrupt controllers are preceded by an Interrupt Controller Crossbar that provides flexibility in mapping the device interrupts to processor interrupts. For more information about crossbars, see the *Control Module* chapter.

**Cortex™-A8 MPU Subsystem Interrupt Controller**

The ARM Cortex-A8 Interrupt Controller (AINTC) is responsible for prioritizing all service requests from the system peripherals and generating either IRQs or FIQs to the Cortex-A8. The AINTC has the capability to handle up to 128 requests, and the priority of the interrupt inputs is programmable. Table 1-94 lists the interrupt sources for the AINTC.

For detailed information about this module, see the *ARM Interrupt Controller (AINTC)* chapter.

**DSP subsystem (DSPSS) Interrupt Controller**

The C674x DSPx interrupt controller combines up to 128 interrupts into 12 prioritized interrupts presented to the CPU. The default sources of the 128 interrupts are shown in Table 1-95. In addition, device interrupts 15 through 95 can alternatively be sourced through the DSP's Interrupt Crossbar from one of the 31 multiplexed device interrupts shown in Table 1-96. The DSP_INTMUX_x registers in the Control Module are used to select between the default interrupt and the multiplexed interrupt. The interrupt controller also controls the generation of the CPU exceptions, NMI, and emulation interrupts.

For detailed information about this module, refer to Chapter 7 of the TMS320C674x DSP Megamodule.

**Media Controller Interrupt Controllers**

A Nested Vectored Interrupt Controller (NVIC) is integrated within each Cortex-M3. The interrupt mapping is the same for the two cores to facilitate parallel processing. The default sources of the 79 interrupts are shown in Table 1-97. In addition, device interrupts 23 through 79 can alternatively be sourced through the Media Controller's Interrupt Crossbar from one of the 63 multiplexed device interrupts shown in Table 1-98. The Media_Controller_INTMUX_x registers in the Control Module are used to select between the default interrupt and the multiplexed interrupt. The NVIC supports:

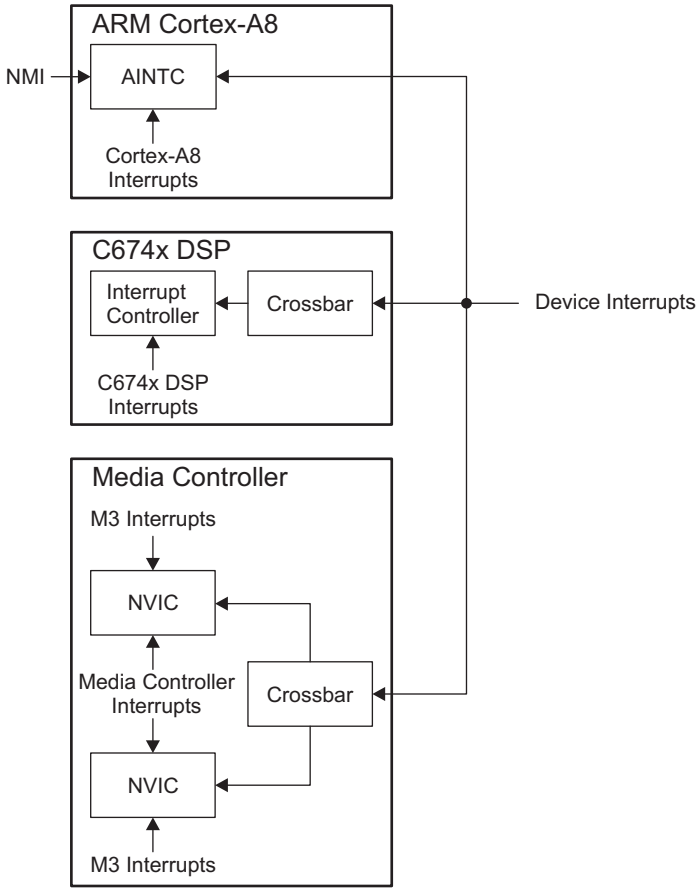
- 64 external interrupts (in addition to 16 Cortex-M3 internal interrupts), which are dynamically prioritized with 16 levels of priority defined for each core
- Low-latency exception and interrupt handling
- Prioritization and handling of exceptions
- Control of the local power management
- Debug accesses to the processor core

For detailed information about this module, refer to ARM *Cortex-M3 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

Figure 1-64 shows the top-level block diagram of the interrupt controllers in this device.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-64. Device Interrupt Controllers**




### 1.7.2 Interrupt Requests to Cortex-A8 MPU INTC

Table 1-94 shows the Cortex-A8 MPU INTC interrupt mapping.

**Table 1-94. Cortex-A8 MPU INTC Interrupt Mapping**

| Interrupt Number | Interrupt Event | Interrupt Source | Description |
|---|---|---|---|
| 0 | EMUINT | CORTEX-A8 | Cortex-A8 Emulation Interrupt |
| 1 | COMMTX | CORTEX-A8 | Cortex-A8 Emulation Interrupt |
| 2 | COMMRX | CORTEX-A8 | Cortex-A8 Emulation Interrupt |
| 3 | BENCH | CORTEX-A8 | Cortex-A8 Emulation Interrupt |
| 4 | ELM_IRQ | ELM | ELM Error Location Process Completion Interrupt |
| 5 | Reserved | Reserved | Reserved |
| 6 | Reserved | Reserved | Reserved |
| 7 | NMI | External Pin | NMIn Pin Interrupt |
| 8 | Reserved | Reserved | Reserved |
| 9 | L3DEBUG | L3 | L3 Debug Error Interrupt |
| 10 | L3APPINT | L3 | L3 Application Error Interrupt |
| 11 | TINT8 | Timer8 | Timer8 Interrupt |
| 12 | EDMA3CC_INT0 | EDMA TPCC (EDMA3CC) | EDMA3CC Region 0 DMA Completion Interrupt |
| 13 | EDMA3CC_MPINT | EDMA3CC | EDMA3CC Memory Protection Error Interrupt |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-94. Cortex-A8 MPU INTC Interrupt Mapping (continued)**

| Interrupt Number | Interrupt Event | Interrupt Source | Description |
|---|---|---|---|
| 14 | EDMA3CC_ERRINT | EDMA3CC | EDMA3CC Error Interrupt |
| 15 | WDTINT | WDTIMER0 | Watchdog Timer 0 Interrupt |
| 16 | SATAINT | SATA | SATA Module Interrupt |
| 17 | USBSSINT | USBSS | USBSS Queue MGR or CPPI Completion Interrupt |
| 18 | USBINT0 | USBSS | USB0 RX/TX DMA, Endpoint Ready/Error Interrupt |
| 19 | USBINT1 | USBSS | USB1 RX/TX DMA, Endpoint Ready/Error Interrupt |
| 20 | Reserved | Reserved | Reserved |
| 21 | Reserved | Reserved | Reserved |
| 22 | Reserved | Reserved | Reserved |
| 23 | Reserved | Reserved | Reserved |
| 24 | Reserved | Reserved | Reserved |
| 25 | Reserved | Reserved | Reserved |
| 26 | Reserved | Reserved | Reserved |
| 27 | Reserved | Reserved | Reserved |
| 28 | SDINT1 | SD1/SDIO1 | SDIO1 Interrupt |
| 29 | SDINT2 | SD2/SDIO2 | SDIO2 Interrupt |
| 30 | I2CINT2 | I2C2 | I2C2 Interrupt |
| 31 | I2CINT3 | I2C3 | I2C3 Interrupt |
| 32 | GPIO2INT0 | GPIO2 | GPIO2 Interrupt 0 |
| 33 | GPIO2INT1 | GPIO2 | GPIO2 Interrupt 1 |
| 34 | USBWAKEUP | USBSS | USBSS Wakeup Interrupt |
| 35 | PCIeWAKEUP | PCIe | PCIe Wakeup Interrupt |
| 36 | HDVPSSINT0 | HDVPSS | HDVPSS Interrupt 0 |
| 37 | GFXINT | SGX530 | SGX530 Error in the IMG Bus Interrupt |
| 38 | HDMIINT | HDMI | HDMI Interrupt |
| 39 | ISSINT5 | ISS | ISS Interrupt 5 |
| 40 | EMACSWRXTHR | EMACSW | EMACSW Receive Threshold Interrupt |
| 41 | EMACSWRXINT | EMACSW | EMACSW Receive Pending Interrupt |
| 42 | EMACSWTXINT | EMACSW | EMACSW Transmit Pending Interrupt |
| 43 | EMACSWMISC | EMACSW | EMACSW Stat, Host, MDIO LINKINT or MDIO USERINT |
| 44 | UARTINT3 | UART3 | UART3 Interrupt |
| 45 | UARTINT4 | UART4 | UART4 Interrupt |
| 46 | UARTINT5 | UART5 | UART5 Interrupt |
| 47 | FDIFINT3 | FDIF | Face Detect Interrupt 3 |
| 48 | PCIINT0 | PCIe | PCIe (Legacy) Interrupt 0 |
| 49 | PCIINT1 | PCIe | PCIe (MSI) Interrupt 1 |
| 50 | PCIINT2 | PCIe | PCIe (Error) Interrupt 2 |
| 51 | PCIINT3 | PCIe | PCIe (Power Management) Interrupt 3 |
| 52 | DCAN0INT0 | DCAN0 | DCAN0 Interrupt 0 |
| 53 | DCAN0INT1 | DCAN0 | DCAN0 Interrupt 1 |
| 54 | DCAN0PARIT | DCAN0 | DCAN0 Parity Interrupt |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-94. Cortex-A8 MPU INTC Interrupt Mapping (continued)**

| Interrupt Number | Interrupt Event | Interrupt Source | Description |
|---|---|---|---|
| 55 | DCAN1INT0 | DCAN1 | DCAN1 Interrupt 0 |
| 56 | DCAN1INT1 | DCAN1 | DCAN1 Interrupt 1 |
| 57 | DCAN1PARIT | DCAN1 | DCAN1 Parity Interrupt |
| 58 | MLBSYSINT0 | MLB | MLB Sys Interrupt 0 |
| 59 | MLBSYSINT1 | MLB | MLB Sys Interrupt 1 |
| 60 | MLBINT | MLB | MLB Interrupt |
| 61 | VCPINT | VCP | VCP Interrupt |
| 62 | GPIO3INT0 | GPIO3 | GPIO3 Interrupt 0 |
| 63 | GPIO3INT1 | GPIO3 | GPIO3 Interrupt 1 |
| 64 | SDINT0 | SD0/SDIO0 | SDIO0 Interrupt |
| 65 | SPIINT0 | SPI0 | SPI0 Interrupt |
| 66 | Reserved | Reserved | Reserved |
| 67 | TINT1 | Timer1 | Timer1 Interrupt |
| 68 | TINT2 | Timer2 | Timer2 Interrupt |
| 69 | TINT3 | Timer3 | Timer3 Interrupt |
| 70 | I2CINT0 | I2C0 | I2C0 Interrupt |
| 71 | I2CINT1 | I2C1 | I2C1 Interrupt |
| 72 | UARTINT0 | UART0 | UART0 Interrupt |
| 73 | UARTINT1 | UART1 | UART1 Interrupt |
| 74 | UARTINT2 | UART2 | UART2 Interrupt |
| 75 | RTCINT | RTC | RTC Timer Interrupt |
| 76 | RTCALARMINT | RTC | RTC Alarm Interrupt |
| 77 | MB1INT0 | Mailbox1 | Mailbox1 Interrupt 0 |
| 78 | IDEINTR | PATA | PATA Interrupt |
| 79 | PLLRECAL | PLL | PLL Recalculation Interrupt |
| 80 | MCATXINT0 | McASP0 | McASP0 Transmit Interrupt |
| 81 | MCARXINT0 | McASP0 | McASP0 Receive Interrupt |
| 82 | MCATXINT1 | McASP1 | McASP1 Transmit Interrupt |
| 83 | MCARXINT1 | McASP1 | McASP1 Receive Interrupt |
| 84 | MCATXINT2 | McASP2 | McASP2 Transmit Interrupt |
| 85 | MCARXINT2 | McASP2 | McASP2 Receive Interrupt |
| 86 | MCBSPINT | McBSP | McBSP Common (TX/RX) Interrupt |
| 87 | Reserved | Reserved | Reserved |
| 88 | Reserved | Reserved | Reserved |
| 89 | SR_DSP_IRQ | SmartReflex DSP (SR_DSP) | SmartReflex ACTIVE (DSP) Domain Interrupt |
| 90 | SR_HDVICP_IRQ | SmartReflex HDVICP (SR_HDVICP) | SmartReflex HDVICP Domain Interrupt |
| 91 | Reserved | Reserved | Reserved |
| 92 | TINT4 | Timer4 | Timer4 Interrupt |
| 93 | TINT5 | Timer5 | Timer5 Interrupt |
| 94 | TINT6 | Timer6 | Timer6 Interrupt |
| 95 | TINT7 | Timer7 | Timer7 Interrupt |
| 96 | GPIO0INT0 | GPIO0 | GPIO0 Interrupt 0 |
| 97 | GPIO0INT1 | GPIO0 | GPIO0 Interrupt 1 |
| 98 | GPIO1INT0 | GPIO1 | GPIO1 Interrupt 0 |
| 99 | GPIO1INT1 | GPIO1 | GPIO1 Interrupt 1 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-94. Cortex-A8 MPU INTC Interrupt Mapping (continued)**

| Interrupt Number | Interrupt Event | Interrupt Source | Description |
|---|---|---|---|
| 100 | GPMCINT | GPMC | GPMC Interrupt |
| 101 | DDRERR0 | DDR EMIF0 | DDR EMIF0 Error Interrupt |
| 102 | DDRERR1 | DDR EMIF1 | DDR EMIF1 Error Interrupt |
| 103 | HDVICPCONT1SYNC | HDVICP | HDVICP iCONT1 Sync Interrupt |
| 104 | HDVICPCONT2SYNC | HDVICP | HDVICP iCONT2 Sync Interrupt |
| 105 | MCATXINT3 | McASP3 | McASP3 Transmit Interrupt |
| 106 | MCARXINT3 | McASP3 | McASP3 Receive Interrupt |
| 107 | HDVICPMBOXINT0 | HDVICP | HDVICP Mailbox Interrupt 0 |
| 108 | MCATXINT4 | McASP4 | McASP4 Transmit Interrupt |
| 109 | MCARXINT4 | McASP4 | McASP4 Receive Interrupt |
| 110 | MCATXINT5 | McASP5 | McASP5 Transmit Interrupt |
| 111 | MCARXINT5 | McASP5 | McASP5 Receive Interrupt |
| 112 | EDMA3TC0_ERRINT | EDMA TPTC0 (EDMA3TC0) | EDMA3TC0 Error Interrupt |
| 113 | EDMA3TC1_ERRINT | EDMA TPTC1 (EDMA3TC1) | EDMA3TC1 Error Interrupt |
| 114 | EDMA3TC2_ERRINT | EDMA TPTC2 (EDMA3TC2) | EDMA3TC2 Error Interrupt |
| 115 | EDMA3TC3_ERRINT | EDMA TPTC3 (EDMA3TC3) | EDMA3TC3 Error Interrupt |
| 116 | Reserved | Reserved | Reserved |
| 117 | Reserved | Reserved | Reserved |
| 118 | Reserved | Reserved | Reserved |
| 119 | Reserved | Reserved | Reserved |
| 120 | SR_ARM_IRQ | SmartReflex ARM (SR_ARM) | SmartReflex ARM Domain Interrupt |
| 121 | SR_CORE_IRQ | SmartReflex CORE (SR_CORE) | SmartReflex CORE Domain Interrupt |
| 122 | SYSMMUINT | System MMU | System MMU Table Walk Abort Interrupt |
| 123 | MCMMUINT | Media Controller | Media Controller MMU Fault Interrupt |
| 124 | DMMINT | DMM | PAT Fault Interrupt |
| 125 | SPIINT1 | SPI1 | SPI1 Interrupt |
| 126 | SPIINT2 | SPI2 | SPI2 Interrupt |
| 127 | SPIINT3 | SPI3 | SPI3 Interrupt |

### 1.7.3 Interrupt Requests to C674x INTC

Table 1-95 shows the C674x INTC default interrupt mapping.

**Table 1-95. C674x INTC Default Interrupt Mapping**

| Interrupt Number | Default Interrupt Name | Default Interrupt Source | Description |
|---|---|---|---|
| 0 | EVT0 | C674x INTC | Output of event combiner 0, for events 1 – 31 |
| 1 | EVT1 | C674x INTC | Output of event combiner 0, for events 32 – 63 |
| 2 | EVT2 | C674x INTC | Output of event combiner 0, for events 64 – 95 |
| 3 | EVT3 | C674x INTC | Output of event combiner 0, for events 96 – 127 |
| 4 | Reserved | Reserved | Reserved |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-95. C674x INTC Default Interrupt Mapping (continued)**

| Interrupt Number | Default Interrupt Name | Default Interrupt Source | Description |
|---|---|---|---|
| 5 | Reserved | Reserved | Reserved |
| 6 | Reserved | Reserved | Reserved |
| 7 | Reserved | Reserved | Reserved |
| 8 | Reserved | Reserved | Reserved |
| 9 | EMU_DTDMA | C674x EMU | C674x EMU DTDMA Interrupt |
| 10 | Reserved | Reserved | Reserved |
| 11 | EMU_RTDXRX | C674x EMU | RTDX Receive Complete Interrupt |
| 12 | EMU_RTDXTX | C674x EMU | RTDX Transmit Complete Interrupt |
| 13 | IDMAINT0 | C674x IDMA | IDMA Channel 0 Interrupt |
| 14 | IDMAINT1 | C674x IDMA | IDMA Channel 1 Interrupt |
| 15 | SDINT0 | SD0/SDIO0 | SD/SDIO0 Interrupt |
| 16 | SPIINT0 | SPI0 | SPI0 Interrupt |
| 17 | VCPINT | VCP | VCP Interrupt |
| 18 | ELM_IRQ | ELM | ELM Error Location Procession Completion Interrupt |
| 19 | IDEINTR | PATA | PATA Interrupt |
| 20 | EDMA3CC_INT1 | EDMA3CC | EDMA3CC Region 1 DMA Completion Interrupt |
| 21 | EDMA3CC_ERRINT | EDMA3CC | EDMA3CC Error Interrupt |
| 22 | EDMA3TC0_ERRINT | EDMA3TC0 | EDMA3TC0 Error Interrupt |
| 23 | ISSINT4 | ISS | ISS Interrupt 4 |
| 24 | Reserved | Reserved | Reserved |
| 25 | Reserved | Reserved | Reserved |
| 26 | Reserved | Reserved | Reserved |
| 27 | EDMA3TC1_ERRINT | EDMA3TC1 | EDMA3TC1 Error Interrupt |
| 28 | EDMA3TC2_ERRINT | EDMA3TC2 | EDMA3TC2 Error Interrupt |
| 29 | EDMA3TC3_ERRINT | EDMA3TC3 | EDMA3TC3 Error Interrupt |
| 30 | SDINT1 | SD1/SDIO1 | SD1/SDIO1 Interrupt |
| 31 | SDINT2 | SD2/SDIO2 | SD2/SDIO2 Interrupt |
| 32 | EMACSWRXTHR | EMACSW | EMACSW Receive Threshold Interrupt |
| 33 | EMACSWRXINT | EMACSW | EMACSW Receive Pending Interrupt |
| 34 | EMACSWTXINT | EMACSW | EMACSW Transmit Pending Interrupt |
| 35 | EMACSWMISC | EMACSW | EMACSW Stat, Host, MDIO LINKINT or MDIO USERINT |
| 36 | PCIINT0 | PCIe | PCIe (Legacy) Interrupt 0 |
| 37 | PCIINT1 | PCIe | PCIe (MSI) Interrupt 1 |
| 38 | PCIINT2 | PCIe | PCIe (Error) Interrupt 2 |
| 39 | PCIINT3 | PCIe | PCIe (Power Management) Interrupt 3 |
| 40 | HDVPSSINT1 | HDVPSS | HDVPSS Interrupt 1 |
| 41 | HDMIINT | HDMI | HDMI Interrupt |
| 42 | SATAINT | SATA | SATA Interrupt |
| 43 | GFXINT | SGX530 | SGX530 Error in the IMG Bus Interrupt |
| 44 | Reserved | Reserved | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-95. C674x INTC Default Interrupt Mapping (continued)**

| Interrupt Number | Default Interrupt Name | Default Interrupt Source | Description |
|---|---|---|---|
| 45 | Reserved | Reserved | Reserved |
| 46 | FDIFINT2 | FDIF | Face Detect Interrupt 2 |
| 47 | Reserved | Reserved | Reserved |
| 48 | Reserved | Reserved | Reserved |
| 49 | TINT1 | Timer1 | Timer1 Interrupt |
| 50 | TINT2 | Timer2 | Timer2 Interrupt |
| 51 | TINT3 | Timer3 | Timer3 Interrupt |
| 52 | TINT4 | Timer4 | Timer4 Interrupt |
| 53 | TINT5 | Timer5 | Timer5 Interrupt |
| 54 | TINT6 | Timer6 | Timer6 Interrupt |
| 55 | TINT7 | Timer7 | Timer7 Interrupt |
| 56 | MB1INT1 | Mailbox1 | Mailbox1 Interrupt 1 |
| 57 | GPIO3INT0 | GPIO3 | GPIO3 Interrupt 0 |
| 58 | I2CINT0 | I2C0 | I2C0 Interrupt |
| 59 | I2CINT1 | I2C1 | I2C1 Interrupt |
| 60 | UARTINT0 | UART0 | UART0 Interrupt |
| 61 | UARTINT1 | UART1 | UART1 Interrupt |
| 62 | UARTINT2 | UART2 | UART2 Interrupt |
| 63 | GPIO3INT1 | GPIO3 | GPIO3 Interrupt 1 |
| 64 | GPIO0INT0 | GPIO0 | GPIO0 Interrupt 0 |
| 65 | GPIO0INT1 | GPIO0 | GPIO0 Interrupt 1 |
| 66 | GPIO1INT0 | GPIO1 | GPIO1 Interrupt 0 |
| 67 | GPIO1INT1 | GPIO1 | GPIO1 Interrupt 1 |
| 68 | GPIO2INT0 | GPIO2 | GPIO2 Interrupt 0 |
| 69 | GPIO2INT1 | GPIO2 | GPIO2 Interrupt 1 |
| 70 | MCATXINT0 | McASP0 | McASP0 Transmit Interrupt |
| 71 | MCARXINT0 | McASP0 | McASP0 Receive Interrupt |
| 72 | MCATXINT1 | McASP1 | McASP1 Transmit Interrupt |
| 73 | MCARXINT1 | McASP1 | McASP1 Receive Interrupt |
| 74 | MCATXINT2 | McASP2 | McASP2 Transmit Interrupt |
| 75 | MCARXINT2 | McASP2 | McASP2 Receive Interrupt |
| 76 | MCBSPINT | McBSP | McBSP Common (TX/RX) Interrupt |
| 77 | UARTINT3 | UART3 | UART3 Interrupt |
| 78 | UARTINT4 | UART4 | UART4 Interrupt |
| 79 | UARTINT5 | UART5 | UART5 Interrupt |
| 80 | MCATXINT3 | McASP3 | McASP3 Transmit Interrupt |
| 81 | MCARXINT3 | McASP3 | McASP3 Receive Interrupt |
| 82 | MCATXINT4 | McASP4 | McASP4 Transmit Interrupt |
| 83 | MCARXINT4 | McASP4 | McASP4 Receive Interrupt |
| 84 | MCATXINT5 | McASP5 | McASP5 Transmit Interrupt |
| 85 | MCARXINT5 | McASP5 | McASP5 Receive Interrupt |
| 86 | SPIINT1 | SPI1 | SPI1 Interrupt |
| 87 | SPIINT2 | SPI2 | SPI2 Interrupt |
| 88 | SPIINT3 | SPI3 | SPI3 Interrupt |
| 89 | I2CINT2 | I2C2 | I2C2 Interrupt |
| 90 | HDVICPCONT1SYNC | HDVICP | HDVICP iCONT1 Sync Interrupt |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-95. C674x INTC Default Interrupt Mapping (continued)**

| Interrupt Number | Default Interrupt Name | Default Interrupt Source | Description |
|---|---|---|---|
| 91 | HDVICPCONT2SYNC | HDVICP | HDVICP iCONT2 Sync Interrupt |
| 92 | I2CINT3 | I2C3 | I2C3 Interrupt |
| 93 | SYSMMUINT | System MMU | System MMU Table Walk Abort Interrupt |
| 94 | HDVICPMBOXINT1 | HDVICP | HDVICP Mailbox Interrupt 1 |
| 95 | GPMCINT | GPMC | GPMC Interrupt |
| 96 | INTERR | C674x INTC | C674x Interrupt Controller Dropped CPU Interrupt |
| 97 | EMC_IDMAERR | C674x EMC | C674x EMC Invalid IDMA Parameters Interrupt |
| 98 | PBISTINT | C674x PBIST | PBIST Interrupt |
| 99 | Reserved | Reserved | Reserved |
| 100 | EFIINTA | C674x EFI A | C674x EFI Interrupt from Side A |
| 101 | EFIINTB | C674x EFI B | C674x EFI Interrupt from Side B |
| 102 | Reserved | Reserved | Reserved |
| 103 | Reserved | Reserved | Reserved |
| 104 | Reserved | Reserved | Reserved |
| 105 | Reserved | Reserved | Reserved |
| 106 | Reserved | Reserved | Reserved |
| 107 | Reserved | Reserved | Reserved |
| 108 | Reserved | Reserved | Reserved |
| 109 | Reserved | Reserved | Reserved |
| 110 | Reserved | Reserved | Reserved |
| 111 | Reserved | Reserved | Reserved |
| 112 | Reserved | Reserved | Reserved |
| 113 | PMC_ED | C674x PMC | C674x PMC Interrupt |
| 114 | Reserved | Reserved | Reserved |
| 115 | Reserved | Reserved | Reserved |
| 116 | UMC_ED1 | C674x UMC | C674x L2 Single Bit Error Detected Interrupt |
| 117 | UMC_ED2 | C674x UMC | C674x L2 Two Bit Error Detected Interrupt |
| 118 | PDC_INT | C674x PDC | Power Down sleep interrupt Interrupt |
| 119 | SYS_CMPA | SYS | SYS Interrupt |
| 120 | PMC_CMPA | C674x PMC | C674x L1P CPU Memory Protection Fault Interrupt |
| 121 | PMC_DMPA | C674x PMC | C674x L1P DMA Memory Protection Fault Interrupt |
| 122 | DMC_CMPA | C674x DMC | C674x L1D CPU Memory Protection Fault Interrupt |
| 123 | DMC_DMPA | C674x DMC | C674x L1D DMA Memory Protection Fault Interrupt |
| 124 | UMC_CMPA | C674x UMC | C674x L2 CPU Memory Protection Fault Interrupt |
| 125 | UMC_DMPA | C674x UMC | C674x L2 DMA Memory Protection Fault Interrupt |
| 126 | EMC_CMPA | C674x EMC | C674x IDMA CPU Memory Protection Fault Interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-95. C674x INTC Default Interrupt Mapping (continued)**

| Interrupt Number | Default Interrupt Name | Default Interrupt Source | Description |
|---|---|---|---|
| 127 | EMC_BUSERR | C674x EMC | C674x IDMA Bus Error Interrupt |

Table 1-96 shows the C674x INTC interrupt crossbar mapping.

**Table 1-96. C674x INTC Interrupt Crossbar Mapping**

| DSP_INTMUX_x Value | Multiplexed Interrupt Name | Multiplexed Interrupt Source | Description |
|---|---|---|---|
| 0 | Default Interrupt (See Table 1-95) | Default | Default |
| 1 | DCAN0INT0 | DCAN0 | DCAN0 Interrupt 0 |
| 2 | DCAN0INT1 | DCAN0 | DCAN0 Interrupt 1 |
| 3 | DCAN0PARIT | DCAN0 | DCAN0 Parity Interrupt |
| 4 | DCAN1INT0 | DCAN1 | DCAN1 Interrupt 0 |
| 5 | DCAN1INT1 | DCAN1 | DCAN1 Interrupt 1 |
| 6 | DCAN1PARIT | DCAN1 | DCAN1 Parity Interrupt |
| 7 | MLBSYSINT0 | MLB | MLB Sys Interrupt 0 |
| 8 | MLBSYSINT1 | MLB | MLB Sys Interrupt 1 |
| 9 | MLBINT | MLB | MLB Interrupt |
| 10 | Reserved | Reserved | Reserved |
| 11 | L3DEBUG | L3 | L3 Debug Error Interrupt |
| 12 | L3APPINT | L3 | L3 Application Error Interrupt |
| 13 | EDMA3CC_MPINT | EDMA3CC | EDMA3CC Memory Protection Error Interrupt |
| 14 | TINT8 | Timer8 | Timer8 Interrupt |
| 15 | WDTINT | WDTIMER0 | Watchdog Timer 0 Interrupt |
| 16 | USBSSINT | USBSS | USBSS Queue MGR or CPPI Completion Interrupt |
| 17 | USBINT0 | USBSS | USB0 RX/TX DMA, Endpoint Ready/Error Interrupt |
| 18 | USBINT1 | USBSS | USB1 RX/TX DMA, Endpoint Ready/Error Interrupt |
| 19 | RTCINT | RTC | RTC Timer Interrupt |
| 20 | RTCALARMINT | RTC | RTC Alarm Interrupt |
| 21 | Reserved | Reserved | Reserved |
| 22 | Reserved | Reserved | Reserved |
| 23 | DDRERR0 | DDR EMIF0 | DDR EMIF0 Error Interrupt |
| 24 | DDRERR1 | DDR EMIF1 | DDR EMIF1 Error Interrupt |
| 25-31 | Reserved | Reserved | Reserved |

### 1.7.4 Interrupt Requests to Media Controller INTC

Table 1-97 shows the Media Controller INTC default interrupt mapping.

**Table 1-97. Media Controller INTC Default Interrupt Mapping**

| Interrupt Number | Default Interrupt Name | Default Interrupt Source | Description |
|---|---|---|---|
| 0 | Reserved | Reserved | Reserved |
| 1 | RESET | M3 | M3 Reset Interrupt |
| 2 | NMI | M3 | M3 NMI Interrupt |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-97. Media Controller INTC Default Interrupt Mapping (continued)**

| Interrupt Number | Default Interrupt Name | Default Interrupt Source | Description |
|---|---|---|---|
| 3 | HARDFAULT | M3 | M3 Hard Fault Interrupt |
| 4 | MEMMANAGERFAULT | M3 | M3 Mem Manager Fault Interrupt |
| 5 | BUSFAULT | M3 | M3 Bus Fault Interrupt |
| 6 | USAGEFAULT | M3 | M3 Usage Fault Interrupt |
| 7 | Reserved | Reserved | Reserved |
| 8 | Reserved | Reserved | Reserved |
| 9 | Reserved | Reserved | Reserved |
| 10 | Reserved | Reserved | Reserved |
| 11 | SVCALL | M3 | M3 SVCall Interrupt |
| 12 | DBUGMON | M3 | M3 Debug Monitor Interrupt |
| 13 | Reserved | Reserved | Reserved |
| 14 | PENDSV | M3 | M3 PendSV Interrupt |
| 15 | SYSTICK | M3 | M3 SYSTICK Interrupt |
| 16 | MMUFAULT | Media controller | Media Controller MMU Fault Interrupt |
| 17 | UCACHEINT | Media controller | Media Controller Unicache Interrupt |
| 18 | CTMINT1 | Media controller | Media Controller CTM Timer Interrupt 1 |
| 19 | HWSEMINT | Media controller | Media Controller HWSEM Interrupt |
| 20 | EMUINT | Media controller | Media Controller EMU Interrupt |
| 21 | IMPFAULT | Media controller | Media Controller IMP Fault Interrupt |
| 22 | CTMINT2 | Media controller | Media Controller CTM Timer Interrupt 2 |
| 23 | HDMIINT | HDMI | HDMI Interrupt |
| 24 | Reserved | Reserved | Reserved |
| 25 | TINT1 | Timer1 | Timer1 Interrupt |
| 26 | TINT2 | Timer2 | Timer2 Interrupt |
| 27 | TINT3 | Timer3 | Timer3 Interrupt |
| 28 | TINT4 | Timer4 | Timer4 Interrupt |
| 29 | TINT5 | Timer5 | Timer5 Interrupt |
| 30 | TINT6 | Timer6 | Timer6 Interrupt |
| 31 | TINT7 | Timer7 | Timer7 Interrupt |
| 32 | ISSINT0 | ISS | ISS Interrupt 0 |
| 33 | ISSINT1 | ISS | ISS Interrupt 1 |
| 34 | ISSINT2 | ISS | ISS Interrupt 2 |
| 35 | ISSINT3 | ISS | ISS Interrupt 3 |
| 36 | ISSINT4 | ISS | ISS Interrupt 4 |
| 37 | ISSINT5 | ISS | ISS Interrupt 5 |
| 38 | HDVICPMBOXINT2 | HDVICP | HDVICP Mailbox Interrupt 2 |
| 39 | HDVICPCONT1SYNC | HDVICP | HDVICP iCONT1 Sync Interrupt |
| 40 | HDVICPCONT2SYNC | HDVICP | HDVICP iCONT2 Sync Interrupt |
| 41 | I2CINT0 | I2C0 | I2C0 Interrupt |
| 42 | I2CINT1 | I2C1 | I2C1 Interrupt |
| 43 | UARTINT0 | UART0 | UART0 Interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-97. Media Controller INTC Default Interrupt Mapping (continued)**

| Interrupt Number | Default Interrupt Name | Default Interrupt Source | Description |
|---|---|---|---|
| 44 | UARTINT1 | UART1 | UART1 Interrupt |
| 45 | UARTINT2 | UART2 | UART2 Interrupt |
| 46 | GPIO0INT0 | GPIO0 | GPIO0 Interrupt 0 |
| 47 | GPIO0INT1 | GPIO0 | GPIO0 Interrupt 1 |
| 48 | VCPINT | VCP | VCP Interrupt |
| 49 | USBSSINT | USBSS | USBSS Queue MGR or CPPI Completion Interrupt |
| 50 | USBINT0 | USBSS | USB0 RX/TX DMA, Endpoint Ready/Error Interrupt |
| 51 | USBINT1 | USBSS | USB1 RX/TX DMA, Endpoint Ready/Error Interrupt |
| 52 | Reserved | Reserved | Reserved |
| 53 | MB1INT2 | Mailbox1 | Mailbox1 Interrupt 2 |
| 54 | MB1INT3 | Mailbox1 | Mailbox1 Interrupt 3 |
| 55 | SR_CORE_IRQ | SmartReflex CORE (SR_CORE) | SmartReflex CORE Domain Interrupt |
| 56 | Reserved | Reserved | Reserved |
| 57 | Reserved | Reserved | Reserved |
| 58 | Reserved | Reserved | Reserved |
| 59 | SR_ARM_IRQ | SmartReflex ARM (SR_ARM) | SmartReflex ARM Domain Interrupt |
| 60 | HDVPSSINT2 | HDVPSS | HDVPSS Interrupt 2 |
| 61 | HDVPSSINT3 | HDVPSS | HDVPSS Interrupt 3 |
| 62 | EDMA3CC_INT4 | EDMA3CC | EDMA3CC Region 4 DMA Completion Interrupt |
| 63 | EDMA3CC_INT5 | EDMA3CC | EDMA3CC Region 5 DMA Completion Interrupt |
| 64 | EMACSWRXTHR | EMACSW | EMACSW Receive threshold Interrupt |
| 65 | EMACSWRXINT | EMACSW | EMACSW Receive pending Interrupt |
| 66 | EMACSWTXINT | EMACSW | EMACSW Transmit pending Interrupt |
| 67 | EMACSWMISC | EMACSW | EMACSW Stat, Host, MDIO LINKINT or MDIO USERINT |
| 68 | MCATXINT0 | McASP0 | McASP0 Transmit Interrupt |
| 69 | MCARXINT0 | McASP0 | McASP0 Receive Interrupt |
| 70 | MCATXINT1 | McASP1 | McASP1 Transmit Interrupt |
| 71 | MCARXINT1 | McASP1 | McASP1 Receive Interrupt |
| 72 | GPIO1INT0 | GPIO1 | GPIO1 Interrupt 0 |
| 73 | GPIO1INT1 | GPIO1 | GPIO1 Interrupt 1 |
| 74 | SDINT0 | SD0/SDIO0 | SDIO0 Interrupt |
| 75 | SPIINT0 | SPI0 | SPI0 Interrupt |
| 76 | Reserved | Reserved | Reserved |
| 77 | Reserved | Reserved | Reserved |
| 78 | GPIO2INT0 | GPIO2 | GPIO2 Interrupt 0 |
| 79 | GPIO2INT1 | GPIO2 | GPIO2 Interrupt 1 |

Table 1-98 shows the Media Controller INTC interrupt crossbar mapping.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-98. Media Controller INTC Interrupt Crossbar Mapping**

| Media_Controller_INTMUX_x Value | Multiplexed Interrupt Name | Multiplexed Interrupt Source | Description |
|---|---|---|---|
| 0 | Default Interrupt (See Table 1-97) | Default | Default |
| 1 | SATAINT | SATA | SATA Interrupt |
| 2 | SDINT1 | SD1/SDIO1 | SDIO1 Interrupt |
| 3 | SDINT2 | SD2/SDIO2 | SDIO2 Interrupt |
| 4 | I2CINT2 | I2C2 | I2C2 Interrupt |
| 5 | I2CINT3 | I2C3 | I2C3 Interrupt |
| 6 | GPIO3INT0 | GPIO3 | GPIO3 Interrupt 0 |
| 7 | GPIO3INT1 | GPIO3 | GPIO3 Interrupt 1 |
| 8 | GFXINT | SGX530 | SGX530 Error in the IMG Bus Interrupt |
| 9 | UARTINT0 | UART0 | UART0 Interrupt |
| 10 | UARTINT1 | UART1 | UART1 Interrupt |
| 11 | UARTINT2 | UART2 | UART2 Interrupt |
| 12 | FDIFINT1 | FDIF | Face Detect Interrupt 1 |
| 13 | PCIINT0 | PCIe | PCIe (Legacy) Interrupt 0 |
| 14 | PCIINT1 | PCIe | PCIe (MSI) Interrupt 1 |
| 15 | PCIINT2 | PCIe | PCIe (Error) Interrupt 2 |
| 16 | PCIINT3 | PCIe | PCIe (Power Management) Interrupt 3 |
| 17 | DCAN0INT0 | DCAN0 | DCAN0 Interrupt 0 |
| 18 | DCAN0INT1 | DCAN0 | DCAN0 Interrupt 1 |
| 19 | DCAN0PARIT | DCAN0 | DCAN0 Parity Interrupt |
| 20 | DCAN1INT0 | DCAN1 | DCAN1 Interrupt 0 |
| 21 | DCAN1INT1 | DCAN1 | DCAN1 Interrupt 1 |
| 22 | DCAN1PARIT | DCAN1 | DCAN1 Parity Interrupt |
| 23 | MLBSYSINT0 | MLB | MLB Sys Interrupt 0 |
| 24 | MLBSYSINT1 | MLB | MLB Sys Interrupt 1 |
| 25 | MLBINT | MLB | MLB Interrupt |
| 26 | RTCINT | RTC | RTC Timer Interrupt |
| 27 | RTCALARMINT | RTC | RTC Alarm Interrupt |
| 28 | IDEINTR | PATA | PATA Interrupt |
| 29 | MCATXINT2 | McASP2 | McASP2 Transmit Interrupt |
| 30 | MCARXINT2 | McASP2 | McASP2 Receive Interrupt |
| 31 | MCBSPINT | McBSP | McBSP Common (TX/RX) Interrupt |
| 32 | GPMCINT | GPMC | GPMC Interrupt |
| 33 | MCATXINT3 | McASP3 | McASP3 Transmit Interrupt |
| 34 | MCARXINT3 | McASP3 | McASP3 Receive Interrupt |
| 35 | MCATXINT4 | McASP4 | McASP4 Transmit Interrupt |
| 36 | MCARXINT4 | McASP4 | McASP4 Receive Interrupt |
| 37 | MCATXINT5 | McASP5 | McASP5 Transmit Interrupt |
| 38 | MCARXINT5 | McASP5 | McASP5 Receive Interrupt |
| 39 | EDMA3TC0_ERRINT | EDMA3TC0 | EDMA3TC0 Error Interrupt |
| 40 | EDMA3TC1_ERRINT | EDMA3TC1 | EDMA3TC1 Error Interrupt |
| 41 | EDMA3TC2_ERRINT | EDMA3TC2 | EDMA3TC2 Error Interrupt |
| 42 | EDMA3TC3_ERRINT | EDMA3TC3 | EDMA3TC3 Error Interrupt |
| 43 | SPIINT1 | SPI1 | SPI1 Interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-98. Media Controller INTC Interrupt Crossbar Mapping (continued)**

| Media_Controller_INTMUX_x Value | Multiplexed Interrupt Name | Multiplexed Interrupt Source | Description |
|---|---|---|---|
| 44 | SPIINT2 | SPI2 | SPI2 Interrupt |
| 45 | SPIINT3 | SPI3 | SPI3 Interrupt |
| 46 | Reserved | Reserved | Reserved |
| 47 | SR_HDVICP_IRQ | SmartReflex HDVICP (SR_HDVICP) | SmartReflex HDVICP Domain Interrupt |
| 48-63 | Reserved | Reserved | Reserved |

### 1.7.5 EDMA Channel Synchronization Events

The EDMA channel controller supports up to 64 channels which service peripherals and memory. Each EDMA channel is mapped to a default EDMA synchronization event as shown in Table 1-99. In addition, each EDMA channel can alternatively be mapped through a crossbar to one of the 31 multiplexed EDMA synchronization events shown in Table 1-100. The EDMA3CC_EVTMUX_x registers in the Control Module are used to select between the default event and the multiplexed events for each channel.

For more detailed information on the EDMA module and how EDMA events are enabled, captured, processed, linked, chained, cleared, and more, see the *Enhanced Direct Memory Access Controller* chapter.

Figure 1-65 shows the top-level block diagram of the EDMA channel synchronization events.

**Figure 1-65. EDMA Channel Synchronization Events**

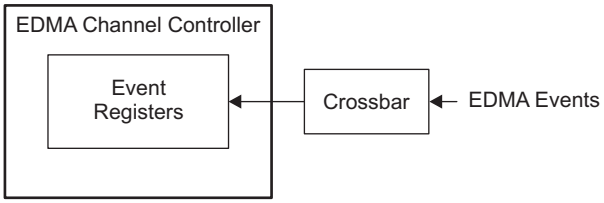



**Table 1-99. EDMA Default Synchronization Events**

| Event Number | Default Event Name | Default Event Source | Description |
|---|---|---|---|
| 0 | VCPREVT0 | VCP | VCP Receive Event 0 |
| 1 | VCPXEVT0 | VCP | VCP Transmit Event 0 |
| 2 | SDTXEVT1 | SD/SDIO1 | SD/SDIO1 Transmit Event |
| 3 | SDRXEVT1 | SD/SDIO1 | SD/SDIO1 Receive Event |
| 4-7 | Reserved | Reserved | Reserved |
| 8 | AXEVT0 | McASP0 | McASP0 Transmit Event |
| 9 | AREVT0 | McASP0 | McASP0 Receive Event |
| 10 | AXEVT1 | McASP1 | McASP1 Transmit Event |
| 11 | AREVT1 | McASP1 | McASP1 Receive Event |
| 12 | AXEVT2 | McASP2 | McASP2 Transmit Event |
| 13 | AREVT2 | McASP2 | McASP2 Receive Event |
| 14 | BXEVT | McBSP | McBSP Transmit Event |
| 15 | BREVT | McBSP | McBSP Receive Event |
| 16 | SPI0XEVT0 | SPI0 | SPI0 Transmit Event 0 |
| 17 | SPI0REVT0 | SPI0 | SPI0 Receive Event 0 |
| 18 | SPI0XEVT1 | SPI0 | SPI0 Transmit Event 1 |
| 19 | SPI0REVT1 | SPI0 | SPI0 Receive Event 1 |
| 20 | SPI0XEVT2 | SPI0 | SPI0 Transmit Event 2 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-99. EDMA Default Synchronization Events (continued)**

| Event Number | Default Event Name | Default Event Source | Description |
|---|---|---|---|
| 21 | SPI0REVT2 | SPI0 | SPI0 Receive Event 2 |
| 22 | SPI0XEVT3 | SPI0 | SPI0 Transmit Event 3 |
| 23 | SPI0REVT3 | SPI0 | SPI0 Receive Event 3 |
| 24 | SDTXEVT0 | SD/SDIO0 | SD/SDIO0 Transmit Event |
| 25 | SDRXEVT0 | SD/SDIO0 | SD/SDIO0 Receive Event |
| 26 | UTXEVT0 | UART0 | UART0 Transmit Event |
| 27 | URXEVT0 | UART0 | UART0 Receive Event |
| 28 | UTXEVT1 | UART1 | UART1 Transmit Event |
| 29 | URXEVT1 | UART1 | UART1 Receive Event |
| 30 | UTXEVT2 | UART2 | UART2 Transmit Event |
| 31 | URXEVT2 | UART2 | UART2 Receive Event |
| 32-35 | Reserved | Reserved | Reserved |
| 36 | ISS_DMA_REQ1 | ISS | ISS Event 1 |
| 37 | ISS_DMA_REQ2 | ISS | ISS Event 2 |
| 38 | ISS_DMA_REQ3 | ISS | ISS Event 3 |
| 39 | ISS_DMA_REQ4 | ISS | ISS Event 4 |
| 40 | CAN_IF1DMA | DCAN0 | DCAN0 IF1 Event |
| 41 | CAN_IF2DMA | DCAN0 | DCAN0 IF2 Event |
| 42 | SPI1XEVT0 | SPI1 | SPI1 Transmit Event 0 |
| 43 | SPI1REVT0 | SPI1 | SPI1 Receive Event 0 |
| 44 | SPI1XEVT1 | SPI1 | SPI1 Transmit Event 1 |
| 45 | SPI1REVT1 | SPI1 | SPI1 Receive Event 1 |
| 46 | Reserved | Reserved | Reserved |
| 47 | CAN_IF3DMA | DCAN0 | DCAN0 IF3 Event |
| 48 | TINT4 | Timer4 | Timer4 Event |
| 49 | TINT5 | Timer5 | Timer5 Event |
| 50 | TINT6 | Timer6 | Timer6 Event |
| 51 | TINT7 | Timer7 | Timer7 Event |
| 52 | GPMCEVT | GPMC | GPMC Event |
| 53 | HDMIEVT | HDMI | HDMI Event |
| 54 | PCIE_TX | PCIe | PCIe Transmit Event |
| 55 | PCIE_RX | PCIe | PCIe Receive Event |
| 56 | AXEVT3 | McASP3 | McASP3 Transmit Event |
| 57 | AREVT3 | McASP3 | McASP3 Receive Event |
| 58 | I2CTXEVT0 | I2C0 | I2C0 Transmit Event |
| 59 | I2CRXEVT0 | I2C0 | I2C0 Receive Event |
| 60 | I2CTXEVT1 | I2C1 | I2C1 Transmit Event |
| 61 | I2CRXEVT1 | I2C1 | I2C1 Receive Event |
| 62 | AXEVT4 | McASP4 | McASP4 Transmit Event |
| 63 | AREVT4 | McASP4 | McASP4 Receive Event |

**Table 1-100. EDMA Multiplexed Synchronization Events**

| EDMA3CC_EVTMUX_x Value | Multiplexed Event Name | Multiplexed Event Source | Description |
|---|---|---|---|
| 0 | Default Event (see Table 1-99) | Default | Default |
| 1 | SDTXEVT2 | SD/SDIO2 | SD/SDIO2 Transmit Event |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-100. EDMA Multiplexed Synchronization Events (continued)**

| EDMA3CC_EVTMUX_x Value | Multiplexed Event Name | Multiplexed Event Source | Description |
|---|---|---|---|
| 2 | SDRXEVT2 | SD/SDIO2 | SD/SDIO2 Receive Event |
| 3 | I2CTXEVT2 | I2C2 | I2C2 Transmit Event |
| 4 | I2CRXEVT2 | I2C2 | I2C2 Receive Event |
| 5 | I2CTXEVT3 | I2C3 | I2C3 Transmit Event |
| 6 | I2CRXEVT3 | I2C3 | I2C3 Receive Event |
| 7 | UTXEVT3 | UART3 | UART3 Transmit Event |
| 8 | URXEVT3 | UART3 | UART3 Receive Event |
| 9 | UTXEVT4 | UART4 | UART4 Transmit Event |
| 10 | URXEVT4 | UART4 | UART4 Receive Event |
| 11 | UTXEVT5 | UART5 | UART5 Transmit Event |
| 12 | URXEVT5 | UART5 | UART5 Receive Event |
| 13 | CAN_IF1DMA | DCAN1 | DCAN1 IF1 Event |
| 14 | CAN_IF2DMA | DCAN1 | DCAN1 IF2 Event |
| 15 | CAN_IF3DMA | DCAN1 | DCAN1 IF3 Event |
| 16 | SPI2XEVT0 | SPI2 | SPI2 Transmit Event 0 |
| 17 | SPI2REVT0 | SPI2 | SPI2 Receive Event 0 |
| 18 | SPI2XEVT1 | SPI2 | SPI2 Transmit Event 1 |
| 19 | SPI2REVT1 | SPI2 | SPI2 Receive Event 1 |
| 20 | SPI3XEVT0 | SPI3 | SPI3 Transmit Event 0 |
| 21 | SPI3REVT0 | SPI3 | SPI3 Receive Event 0 |
| 22 | Reserved | Reserved | Reserved |
| 23 | TINT1 | Timer1 | Timer1 Event |
| 24 | TINT2 | Timer2 | Timer2 Event |
| 25 | TINT3 | Timer3 | Timer3 Event |
| 26 | AXEVT5 | McASP5 | McASP5 Transmit Event |
| 27 | AREVT5 | McASP5 | McASP5 Receive Event |
| 28 | EDMAEVT0 | EDMA_EVT0 Pin | EDMA_EVT0 Pin Event |
| 29 | EDMAEVT1 | EDMA_EVT1 Pin | EDMA_EVT1 Pin Event |
| 30 | EDMAEVT2 | EDMA_EVT0 Pin | EDMA_EVT2 Pin Event |
| 31 | EDMAEVT3 | EDMA_EVT0 Pin | EDMA_EVT3 Pin Event |


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.8 Inter-Processor Communication

This SoC is a heterogeneous multi-core device, which requires software to efficiently manage and communicate between the cores. The following are the main features that need to be implemented by any such software:

1. Device management of the slave processors from the host processor.
2. Inter-processor communication between the cores for transfer and exchange of information between them.

In this SoC, the host processor is usually the Cortex A8. This processor is responsible for boot-loading the slave processors (Video-M3, VPSS-M3).

Boot-loading includes power management of the slaves (power-up/down and other power management), reset control (reset/release of the slave processor) and setting the entry point of the slave executable into the appropriate register.

For implementing efficient Inter-Processor Communication between the multiple cores on the device, certain hardware features are provided:

- Mailbox interrupts
- Hardware Spinlocks

### *1.8.1 Reset Requirements*

This SoC has a power on reset (POR) and warm reset. For the POR reset, the A8 is taken out of reset and it boots from its boot ROM. Once booted, the A8 will boot load the VPSS M3 and Video M3.

### *1.8.2 Features*

Different methods of IPC are used between the various processing elements in the SoC:

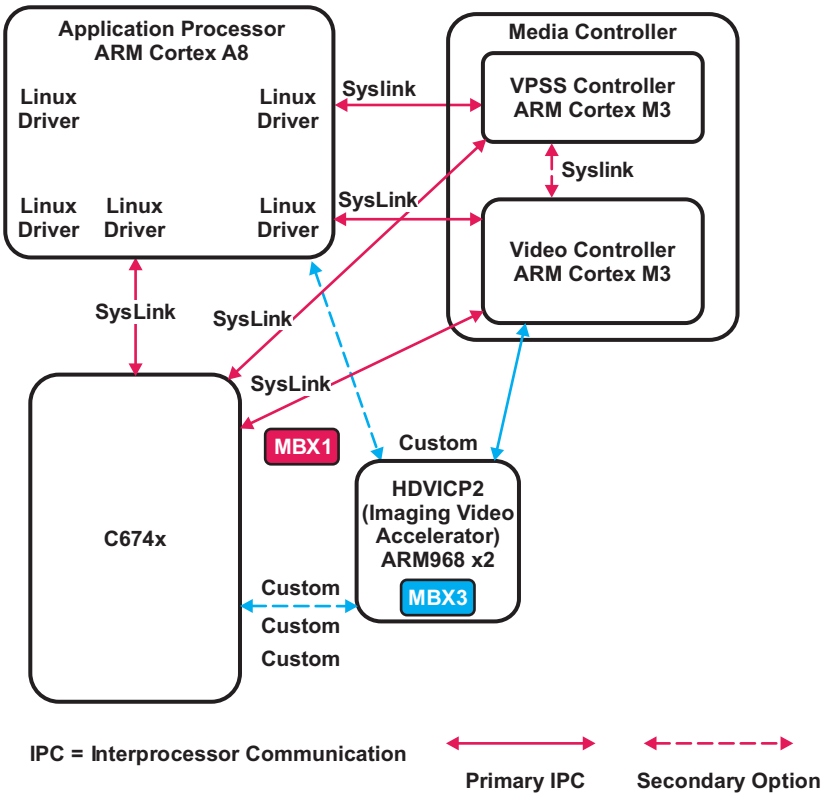
- Custom IPC – This is an IPC that doesn't use any standard IPC software.
- SysLink – This is a new implementation of the BIOS Link that allows communication between the processors in the SoC. SysLink includes ProcMgr module for device management of the slave cores (Video-M3, VPSS-M3, C674x™). The Notify method is the API for responding to interrupts. SysLink also incorporates full IPC features including MessageQ, RingIO, and FrameQ.


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.8.3 Overview and Strategy

Figure 1-66 depicts the processing elements that are part of the SoC.

**Figure 1-66. IPC Overview Diagram**




The SoC contains Mailbox and Spinlock hardware to facilitate the IPC mechanism.

Mailboxes provide a mechanism for one processor to write a value to a register and send an interrupt to another processor. One system level mailbox is provided for the IPCs between the A8, the C674x™, and the Media M3s. The HDVICP2 has its own mailbox within its IP module. The SoC shall provide Spinlocks to facilitate Mutexes for access to shared resources in the system.

Mailboxes and Spinlocks will be discussed in more detail later in this document.

It is intended that the following IPC mechanisms shall be used between various processors:

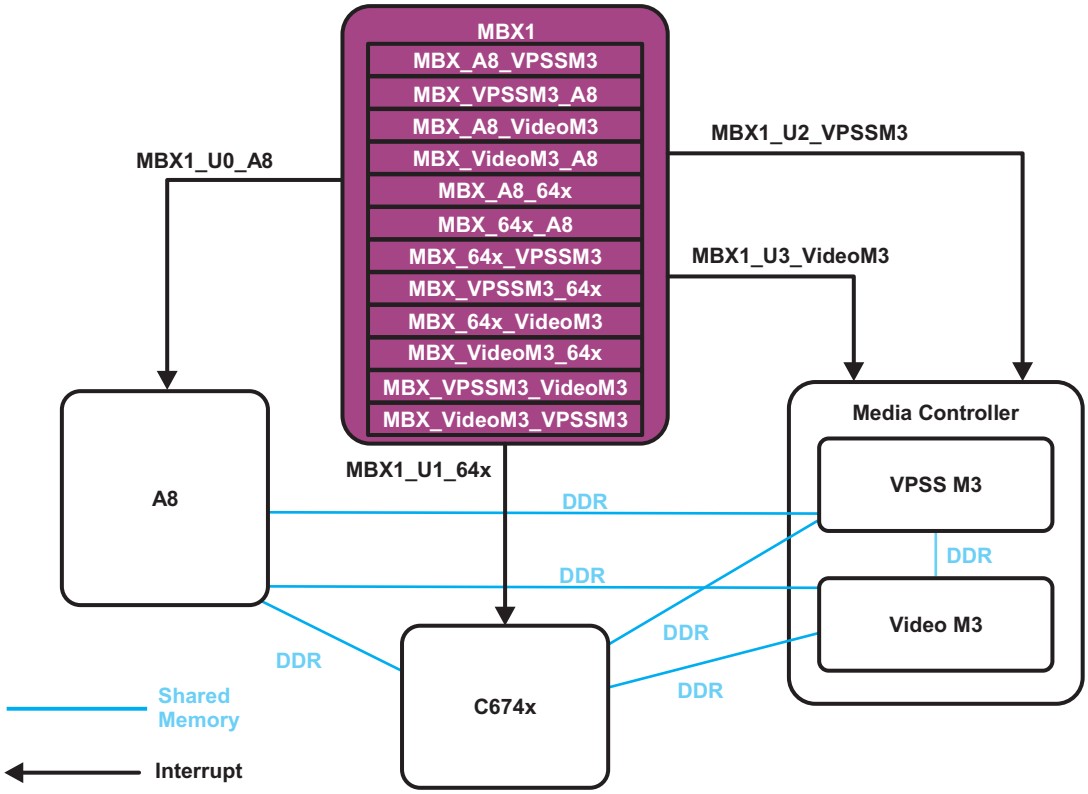
- SysLink / IPC
  - A8 to C674x™
  - A8 to Video M3
  - A8 to VPSS M3
  - Video M3 to VPSS M3
  - C674x™ to VideoM3
  - C674x™ to VPSS M3
- Custom
  - Video M3 to HDVICP2


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.8.3.1 System IPCs

Figure 1-67 depicts how Mailbox 1 (MBX1) will be used for system IPCs. Four interrupts are generated from the mailbox that allows the A8, C674x™, VPSS M3, and Video M3 to communicate. DDR is used as the shared memory interface. An IPC can be created between any of the processors listed. The A8 will communicate to the C674x™, Video M3 and VPSS M3 using SysLink.

**Figure 1-67. System IPCs**



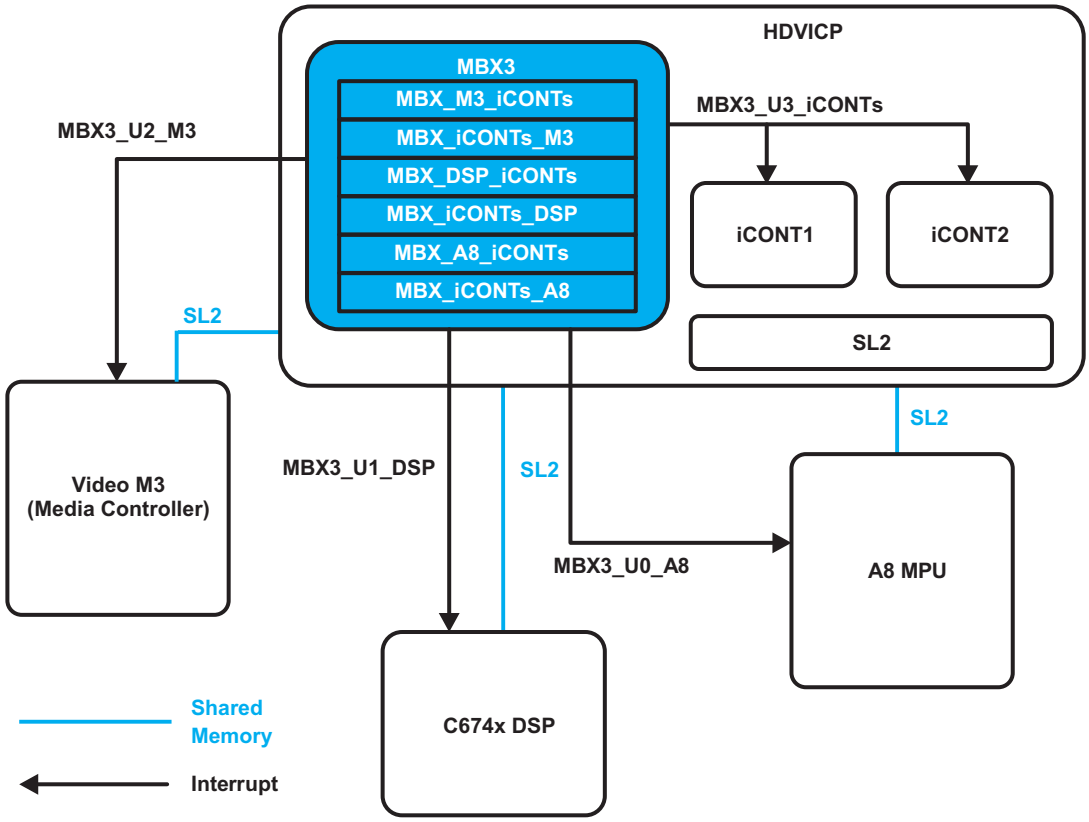


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.8.3.2 HDVICP2 IPC

Figure 1-68 depicts how Mailbox 3 (MBX3) will be used for the IPC between the HDVICP2 and the other processors that may control it. The plan of record is that only one processor will be chosen to control the HDVICP2 at a time. The Video M3 is the primary processor that will control the HDVICP2. Interrupts from the mailbox are provided as shown in Figure 1-68. IPCs will use the Shared Level 2 memory that is part of the HDVICP2 module. This is necessary since the HDVICP2 processors cannot access DDR or the L3/L4 switch fabric. The HDVICP2 software will reside beneath the xDM API.

**Figure 1-68. HDVICP2 IPC**





Copyright © 2011–2016, Texas Instruments Incorporated

## 1.8.4 IPC Component Configuration

### 1.8.4.1 Shared Memory

Shared memory will be used as necessary for the IPCs of the system and it is important that cache coherency is maintained.

### 1.8.4.2 IPC Interrupt Module - Mailboxes

Interrupts used by IPCs are generated from mailbox modules.

All mailboxes use interrupts with both an active high level and active low level. The appropriate type of interrupt is connected to each processor based on the characteristics of that processor.

Each mailbox provides two messages per mailbox submodule.

### 1.8.4.3 Hardware Spinlocks

Spinlocks are pared down semaphores designed to provide direct access mutex support without the need for interrupt generator or queues. Spinlocks can be used very effectively as a mechanism to protect data structures in shared memory space that might be accessed by multiple processing cores simultaneously. They can be utilized if the following conditions apply:

1. The time to hold the lock is predictable and small. (How small is a hardware/software system design consideration. For example a maximum hold time of less than 200 CPU cycles may be acceptable).
2. The locking task cannot be preempted or suspended or interrupted while holding the lock. (This would make the hold time large and unpredictable).
3. The lock is lightly contended, i.e., the chance of any other process (or processor) trying to acquire the lock while it is held is small.

If these conditions hold, then the locking code can retry a failed attempt to acquire the lock until success.

The spin lock unit is responsible for providing hardware assistance for synchronizing the processes running on multiple processors in the device:

- The application processor (ARM Cortex-A8)
- C674x™

Note that the HDVICP2 can’t access the L3/L4 switch fabric, so IPCs to the HDVICP2 can’t use spinlocks.

Hardware spinlocks are used by SysLink and IPC products to provide multi-core mutual-exclusion between the processors on this SoC. This is used to protect access to control structures in shared memory. In addition spinlocks can provide a mutual exclusion mechanism that could be used in a system level resource manager.

This SoC has 128 spinlocks for the system.

**Table 1-101. Hardware Spinlock Configuration**

| Hardware Spinlocks | Name | Configuration |
|---|---|---|
| 0.127 | SPINLOCK_0 .. SPINLOCK_127 | No configurable options |


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.9 Mailbox

### 1.9.1 Overview

Communication between the on-chip processors of the device uses a queued mailbox-interrupt mechanism.

The queued mailbox-interrupt mechanism allows the software to establish a communication channel between two processors through a set of registers and associated interrupt signals by sending and receiving messages (mailboxes).

There are two mailbox module instances in the device:

- System mailbox - used for Cortex-A8 microprocessor unit (Cortex-A8 MPU) and digital signal processor (DSP) communications.
- HDVICP2 mailbox - used for communication between one internal to the HDVICP2 subsystem user (imaging controller 1 - iCont1, or imaging controller 2 - iCont2) and three external to the HDVICP2 subsystem users (Cortex-A8 MPU, DSP and Video M3). This communication is insured through three pairs of mailboxes.

The mailbox module includes the following features:

- Four users for the system mailbox instance (Cortex-A8 MPU, DSP and Media Controller Subsystem) or four users for the HDVICP2 mailbox instance (iCont1/iCont2, Cortex-A8 MPU, DSP and Video-M3 MPU)
- Four mailbox message queues for the system mailbox instance/six mailbox message queues for the HDVICP2 mailbox instance
- Flexible assignment of receiver and sender for each mailbox through interrupt configuration
- Four interrupts (one per user) for the system mailbox instance/four interrupts (one per user) for the HDVICP2 mailbox instance
- 32-bit message width
- Four-message FIFO depth for each message queue
- Message reception and queue-not-full notification using interrupts
- Support of 16-/32-bit addressing scheme
- Power management support

### 1.9.2 System Mailbox Integration

This section describes the system mailbox integration in the device, including information about clocks, resets, and hardware requests. Figure 1-69 shows the system mailbox integration.

Table 1-102 through Table 1-104 summarize the system mailbox integration in the device.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-69. System Mailbox Integration**

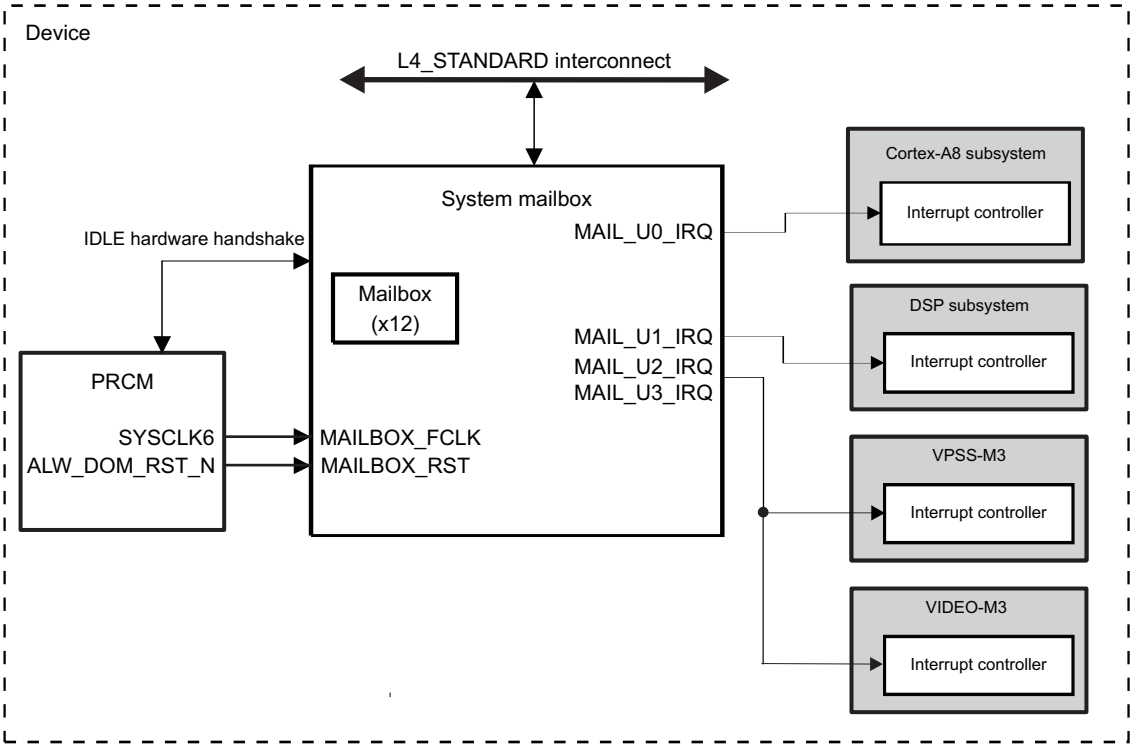



**Table 1-102. Integration Attributes**

| Module Instance | Attributes | |
|---|---|---|
| | Power Domain | Interconnect |
| SYSTEM_MAILBOX | PD_ALWAYS_ON | L4_STANDARD |

**Table 1-103. Clocks and Resets**

| **Clocks** | | | | |
|---|---|---|---|---|
| Module Instance | Destination Signal Name | Source Signal | Source | Description |
| SYSTEM_MAILBOX | MAILBOX_FCLK | SYSCLK6 | PRCM | Mailbox interface clock. This clock is used for all interface and functional operations. |
| **Resets** | | | | |
| Module Instance | Destination Signal Name | Source Signal | Source | Description |
| SYTEM_MAILBOX | MAILBOX_RST | ALW_DOM_RST_N | PRCM | Mailbox hardware reset. This reset is asynchronously applied to the Mailbox registers |

**Table 1-104. Hardware Requests**

| Module Instance | Source Signal Name | Destination Signal | Destination | Description |
|---|---|---|---|---|
| SYSTEM_MAILBOX | MAIL_U0_IRQ | A_IRQ_77 | Cortex-A8 | System Mailbox user 0 interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-104. Hardware Requests (continued)**

| Module Instance | Source Signal Name | Destination Signal | Destination | Description |
|---|---|---|---|---|
| | MAIL_U1_IRQ | D_IRQ_56 | DSP | System Mailbox user 1 interrupt |

### 1.9.3 Functional Description

This device has the following mailbox instances:

- System mailbox
- HDVICP2 mailbox

Table 1-105 shows Mailbox Implementation in this device, where u is the user number and m is the mailbox number.

**Table 1-105. Mailbox Implementation**

| Mailbox Type | User Number(u) | Mailbox Number(m) | Messages per Mailbox |
|---|---|---|---|
| System mailbox | 0 to 3 | 0 to 11 | 4 |
| HDVICP2 mailbox | 0 to 3 | 0 to 5 | 4 |

The mailbox module provides a means of communication through message queues among the users (depending on the mailbox module instance). The individual mailbox modules (12 for the system mailbox instance, six for the HDVICP2 mailbox instance), or FIFOs, can associate (or de-associate) with any of the processors using the MAILBOX_IRQENABLE_SET_u (or MAILBOX_IRQENABLE_CLR_u) register.

The system mailbox module includes the following user subsystems:

- User 0: Cortex-A8 MPU subsystem (u = 0)
- User 1: DSP subsystem (u = 1)

The HDVICP2 mailbox module includes the following user subsystems, where x=0,1,2.:

- User 0: Cortex-A8 MPU subsystem (u = 0)
- User 1: DSP subsystem (u = 1)
- User 2: HDVICP2 subsystem - only available to the HDVICP2 mailbox instance (u = 3)

Each user has a dedicated interrupt signal from the corresponding mailbox module instance and dedicated interrupt enabling and status registers. Each MAILBOX_IRQSTATUS_RAW_u/MAILBOX_IRQSTATUS_CLR_u interrupt status register corresponds to a particular user.

For the system mailbox instance, a user can query its interrupt status register through the L4_STANDARD interconnect.

For the HDVICP2 mailbox instance, a user can query its interrupt status register as follows:

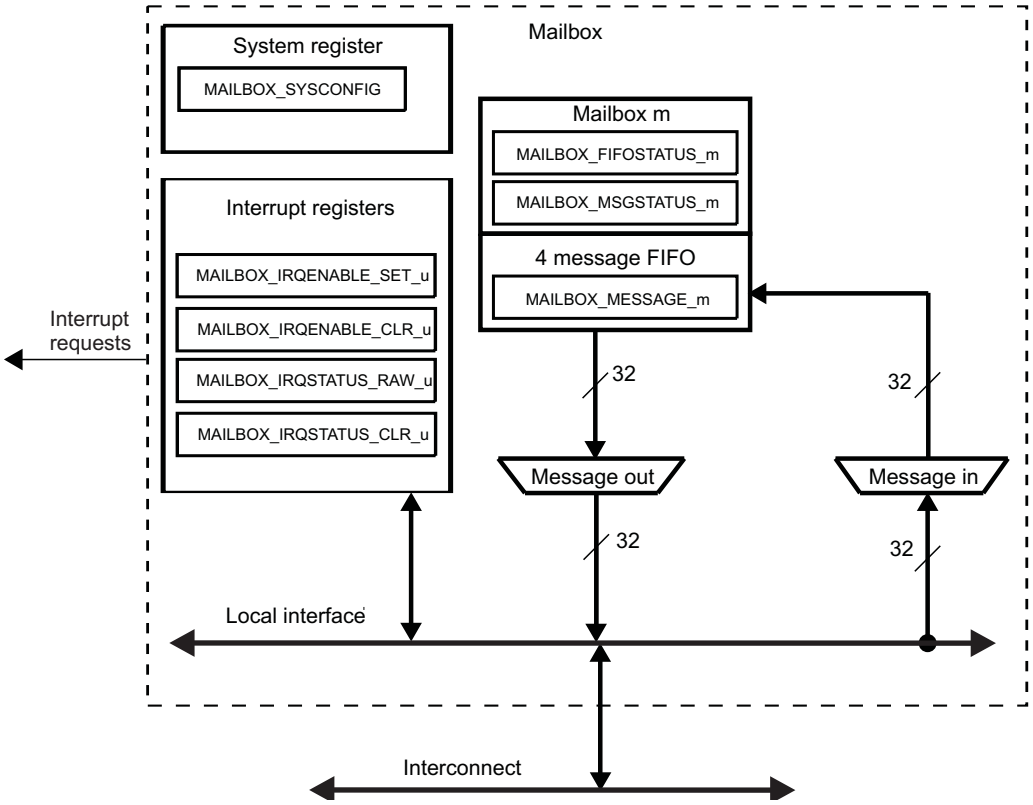
- Cortex-A8 MPU and DSP - through the L3 interconnect
- iCont1/iCont2 and DSP - private access (directly through the HDVICP2 and DSP local interconnect respectively)


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.9.3.1 Mailbox Block Diagram

Figure 1-70 shows the mailbox block diagram.

**Figure 1-70. Mailbox Block Diagram**




### 1.9.3.2 Software Reset

The mailbox module supports a software reset through the MAILBOX_SYSCONFIG[0].SOFTRESET bit. Setting this bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Reading the MAILBOX_SYSCONFIG[0] SOFTRESET bit gives the status of the software reset:

- Read 1: the software reset is on-going
- Read 0: the software reset is complete

The software must ensure that the software reset completes before doing mailbox operations.

### 1.9.3.3 Power Management

Table 1-106 describes power-management features available for the mailbox module.

**Table 1-106. Local Power Management Features**

| Feature | Registers | Description |
|---|---|---|
| Clock autogating | NA | Feature not available |
| Slave idle modes | MAILBOX_SYSCONFIG[3:2].SIDLEMODE | Force-idle, no-idle and smart-idle modes are available |
| Clock activity | NA | Feature not available |
| Master standby modes | NA | Feature not available |
| Global wake-up enable | NA | Feature not available |
| Wake-up sources enable | NA | Feature not available |


Copyright © 2011–2016, Texas Instruments Incorporated

The mailbox module can be configured using the MAILBOX_SYSCONFIG[3:2] SIDLEMODE bit field to one of the following acknowledgment modes:

- Force-idle mode (SIDLEMODE = 0x0): The mailbox module immediately enters the idle state on receiving a low-power-mode request from the PRCM module. In this mode, the software must ensure that there are no asserted output interrupts before requesting this mode to go into the idle state.
- No-idle mode (SIDLEMODE = 0x1): The mailbox module never enters the idle state.
- Smart-idle mode (SIDLEMODE = 0x2): After receiving a low-power-mode request from the PRCM module, the mailbox module enters the idle state only after all asserted output interrupts are acknowledged.

#### 1.9.3.4 Interrupt Requests

An interrupt request allows the user of the mailbox to be notified when a message is received or when the message queue is not full. There is one interrupt per user. Table 1-107 lists the event flags, and their mask, that can cause module interrupts.

**Table 1-107. Interrupt Events**

| Non-Maskable Event Flag[(1)] | Maskable Event Flag | Event Mask Bit | Event Unmask Bit | Description |
|---|---|---|---|---|
| MAILBOX_IRQSTATUS_RAW_u[0+m*2].NEWMSGSTATUSUUMBm | MAILBOX_IRQSTATUS_CLR_u[0+m*2].NEWMSGSTATUSUUMBm | MAILBOX_IRQENABLE_CLR_u[0+m*2]. | | |
| MAILBOX_IRQSTATUS_RAW_u[0+m*2].NEWMSGSTATUSUUMBm | MAILBOX_IRQSTATUS_CLR_u[0+m*2].NEWMSGSTATUSUUMBm | MAILBOX_IRQENABLE_CLR_u[0+m*2]. NEWMSGSTATUSUUMBm | MAILBOX_IRQENABLE_SET_u[0+m*2]. NEWMSGSTATUSUUMBm | Mailbox m receives a new message |
| MAILBOX_IRQSTATUS_RAW_u[1+m*2].NOTFULLSTATUSUMBm | MAILBOX_IRQSTATUS_CLR_u[1+m*2].NOTFULLSTATUSUMBm | MAILBOX_IRQENABLE_CLR_u[1+m*2]. NOTFULLSTATUSUMBm | MAILBOX_IRQENABLE_SET_u[1+m*2]. NOTFULLSTATUSUMBm | Mailbox m message queue is not full |

(1) MAILBOX.MAILBOX_IRQSTATUS_RAW_u register is mostly used for debug purposes.

**CAUTION**

Once an event generating the interrupt request has been processed by the software, it must be cleared by writing a logical 1 in the corresponding bit of the MAILBOX_IRQSTATUS_CLR_u register. Writing a logical 1 in a bit of the MAILBOX_IRQSTATUS_CLR_u register will also clear to 0 the corresponding bit in the appropriate MAILBOX_IRQSTATUS_RAW_u register.

An event can generate an interrupt request when a logical 1 is written to the corresponding unmask bit in the MAILBOX_IRQENABLE_SET_u register. Events are reported in the appropriate MAILBOX_IRQSTATUS_CLR_u and MAILBOX_IRQSTATUS_RAW_u registers.

An event stops generating interrupt requests when a logical 1 is written to the corresponding mask bit in the MAILBOX_IRQENABLE_CLR_u register. Events are only reported in the appropriate MAILBOX_IRQSTATUS_RAW_u register.

In case of the MAILBOX_IRQSTATUS_RAW_u register, the event is reported in the corresponding bit even if the interrupt request generation is disabled for this event.


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.9.3.5 Assignment

#### *1.9.3.5.1 Description*

To assign a receiver to a mailbox, set the new message interrupt enable bit corresponding to the desired mailbox in the MAILBOX_IRQENABLE_SET_u register. The receiver reads the MAILBOX_MESSAGE_m register to retrieve a message from the mailbox.

An alternate method for the receiver that does not use the interrupts is to poll the MAILBOX_FIFOSTATUS_m and/or MAILBOX_MSGSTATUS_m registers to know when to send or retrieve a message to or from the mailbox. This method does not require assigning a receiver to a mailbox. Because this method does not include the explicit assignment of the mailbox, the software must avoid having multiple receivers use the same mailbox, which can result in incoherency.

To assign a sender to a mailbox, set the queue-not-full interrupt enable bit of the desired mailbox in the MAILBOX_IRQENABLE_SET_u register, where u is the number of the sending user. However, direct allocation of a mailbox to a sender is not recommended because it can cause the sending processor to be constantly interrupted.

It is recommended that register polling be used to:

- Check the status of either the MAILBOX_FIFOSTATUS_m or MAILBOX_MSGSTATUS_m registers
- Write the message to the corresponding MAILBOX_MESSAGE_m register, if space is available

The sender might use the queue-not-full interrupt when the initial mailbox status check indicates the mailbox is full. In this case, the sender can enable the queue-not-full interrupt for its mailbox in the appropriate MAILBOX_IRQENABLE_SET_u register. This allows the sender to be notified by interrupt only when a FIFO queue has at least one available entry.

Reading the MAILBOX_IRQSTATUS_CLR_u register determines the status of the new message and the queue-not-full interrupts for a particular user. Writing 1 to the corresponding bit in the MAILBOX_IRQSTATUS_CLR_u register acknowledges, and subsequently clears, an interrupt.

**CAUTION**

Assigning multiple senders or multiple receivers to the same mailbox is not recommended.

### 1.9.3.6 Sending and Receiving Messages

#### *1.9.3.6.1 Description*

When a 32-bit message is written to the MAILBOX_MESSAGE_m register, the message is appended into the FIFO queue. This queue holds four messages. If the queue is full, the message is discarded. Queue overflow can be avoided by first reading the MAILBOX_FIFOSTATUS_m register to check that the mailbox message queue is not full before writing a new message to it. Reading the MAILBOX_MESSAGE_m register returns the message at the beginning of the FIFO queue and removes it from the queue. If the FIFO queue is empty when the MAILBOX_MESSAGE_m register is read, the value 0 is returned. The new message interrupt is asserted when at least one message is in the mailbox message FIFO queue. To determine the number of messages in the mailbox message FIFO queue, read the MAILBOX_MSGSTATUS_m register.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.9.3.7 16-Bit Register Access

##### *1.9.3.7.1 Description*

So that 16-bit processors can access the mailbox module, the module allows 16-bit register read and write access, with restrictions for the MAILBOX_MESSAGE_m registers. The 16-bit half-words are organized in little endian fashion; that is, the least-significant 16 bits are at the low address and the most-significant 16 bits are at the high address (low address + 0x02). All mailbox module registers can be read or written to directly using individual 16-bit accesses with no restriction on interleaving, except the MAILBOX_MESSAGE_m registers, which must always be accessed by either single 32-bit accesses or two consecutive 16-bit accesses.

**CAUTION**

When using 16-bit accesses to the MAILBOX_MESSAGE_m registers, the order of access must be the least-significant half-word first (low address) and the most-significant half-word last (high address). This requirement is because of the update operation by the message FIFO of the MAILBOX_MSGSTATUS_m registers. The update of the FIFO queue contents and the associated status registers and possible interrupt generation occurs only when the most-significant 16 bits of a MAILBOX_MESSAGE_m are accessed.

### *1.9.4 Programming Guide*

#### 1.9.4.1 Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the mailbox module.

##### *1.9.4.1.1 Global Initialization*

###### *1.9.4.1.1.1 Surrounding Modules Global Initialization*

This section identifies the requirements of initializing the surrounding modules when the mailbox module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the mailbox.

See Section 1.9.2 for further information.

**Table 1-108. Global Initialization of Surrounding Modules for System Mailbox**

| Surrounding Modules | Comments |
|---|---|
| PRCM | Mailbox functional/interface clock must be enabled. |
| Interrupt Controllers | Cortex-A8 MPU or DSP interrupt controller must be configured to enable the interrupt request generation to the Cortex-A8 MPU or DSP subsystem. |

**Table 1-109. Global Initialization of Surrounding Modules for HDVICP2 Mailbox**

| Surrounding Modules | Comments |
|---|---|
| PRCM | Mailbox functional/interface clock must be enabled. |
| Interrupt Controllers | Cortex-A8 MPU or DSP or iCont1/iCont2 of HDVICP2 Subsystem interrupt controller must be configured to enable the interrupt request generation to the Cortex-A8 MPUor DSP subsystem or HDVICP2 Subsystem |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-110. Global Initialization of Surrounding Modules for HDVICP2-1 Mailbox**

| Surrounding Modules | Comments |
|---|---|
| PRCM | Mailbox functional/interface clock must be enabled. |
| Interrupt Controllers | Cortex-A8 MPU or DSP or iCont1/iCont2 of HDVICP2-1 Subsystem interrupt controller must be configured to enable the interrupt request generation to the Cortex-A8 MPU or DSP subsystem or HDVICP2-1 Subsystem |

**Table 1-111. Global Initialization of Surrounding Modules for HDVICP2-2 Mailbox**

| Surrounding Modules | Comments |
|---|---|
| PRCM | Mailbox functional/interface clock must be enabled. |
| Interrupt Controllers | Cortex-A8 MPU or DSP or iCont1/iCont2 of HDVICP2-2 Subsystem interrupt controller must be configured to enable the interrupt request generation to the Cortex-A8 MPU or DSP subsystem or HDVICP2-2 Subsystem |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.9.4.1.1.2 Mailbox Global Initialization

##### 1.9.4.1.1.2.1 Main Sequence - Mailbox Global Initialization

This procedure initializes the mailbox module after a power-on or software reset.

**Table 1-112. Mailbox Global Initialization**

| | Register/Bitfield/Programming Model | Value |
|---|---|---|
| Perform a software reset | MAILBOX_SYSCONFIG[0].SOFTRESET | 1 |
| Wait until reset is complete | MAILBOX_SYSCONFIG[0].SOFTRESET | 0 |
| Set idle mode configuration | MAILBOX_SYSCONFIG[3:2].SIDLEMODE | 0x- |

#### 1.9.4.1.2 Operational Modes Configuration

##### 1.9.4.1.2.1 Main Sequence - Sending a Message (Polling Method)

**Table 1-113. Sending a Message (Polling Method)**

| Step | Register/Bitfield/Programming Model | Value |
|---|---|---|
| IF : Is FIFO full ? | MAILBOX_FIFOSTATUS_m[0].FIFOFULLMB | =1h |
| Wait until at least one message slot is available | MAILBOX_FIFOSTATUS_m[0].FIFOFULLMB | =0h |
| ELSE | | |
| Write message | MAILBOX_MESSAGE_m[31:0].MESSAGEVALUEMBM | ----h |
| ENDIF | | |

##### 1.9.4.1.2.2 Main Sequence - Sending a Message (Interrupt Method)

**Table 1-114. Sending a Message (Interrupt Method)**

| Step | Register/Bitfield/Programming Model | Value |
|---|---|---|
| IF : Is FIFO full ? | MAILBOX_FIFOSTATUS_m[0].FIFOFULLMB | =1h |
| Enable interrupt event | MAILBOX_IRQENABLE_SET_u[1+ m*2] | 1h |
| User(processor) can perform anothr task until interrupt occurs | | |
| ELSE | | |
| Write message | MAILBOX_MESSAGE_m[31:0].MESSAGEVALUEMBM | ----h |
| ENDIF | | |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.9.4.1.2.3 Main Sequence - Receiving a Message (Polling Method)

**Table 1-115. Receiving a Message (Polling Method)**

| Step | Register/Bitfield/Programming Model | Value |
|---|---|---|
| IF : Number of messages is not equal to 0 | MAILBOX_MSGSTATUS_m[2:0].NBOFMSGMB | !=0h |
| Read message | MAILBOX_MESSAGE_m[31:0].MESSAGEVALUEMBM | ----h |
| ENDIF | | |

#### 1.9.4.1.2.4 Main Sequence - Receiving a Message (Interrupt Method)

**Table 1-116. Receiving a Message (Interrupt Method)**

| Step | Register/Bitfield/Programming Model | Value |
|---|---|---|
| Enable interrupt event | MAILBOX_IRQENABLE_SET_u[0 + m*2] | 1h |
| User(processor) can perform anothr task until interrupt occurs | | |

### 1.9.4.1.3 Events Servicing

#### 1.9.4.1.3.1 Sending Mode

Table 1-117 describes the events servicing in sending mode.

**Table 1-117. Events Servicing in Sending Mode**

| Step | Register/Bitfield/Programming Model | Value |
|---|---|---|
| Read interrupt status bit | MAILBOX_IRQSTATUS_CLR_u[1 + m*2] | 1 |
| Write message | MAILBOX_MESSAGE_m[31:0].MESSAGEVALUEMBM | ----h |
| Write 1 to acknowledge interrupt | MAILBOX_IRQSTATUS_CLR_u[1 + m*2] | 1 |

#### 1.9.4.1.3.2 Receiving Mode

Table 1-118 describes the events servicing in receiving mode.

**Table 1-118. Events Servicing in Receiving Mode**

| Step | Register/Bitfield/Programming Model | Value |
|---|---|---|
| Read interrupt status bit | MAILBOX_IRQSTATUS_CLR_u[0 + m*2] | 1 |
| IF : Number of messages is not equal to 0 ? | MAILBOX_MSGSTATUS_m[2:0].NBOFMSGMB | !=0h |
| Read message | MAILBOX_MESSAGE_m[31:0].MESSAGEVALUEMBM | ----h |
| ELSE | | |
| Write 1 to acknowledge interrupt | MAILBOX_IRQSTATUS_CLR_u[0 + m*2] | 1 |
| ENDIF | | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.9.5 Mailbox Registers

Table 1-119 lists the Mailboxes available on the device. The previous register set is applicable to these mailboxes. See the device-specific data manual for the memory address of these mailboxes.

**Table 1-119. Mailboxes**

| Mailbox Type | User Number(u) | Mailbox Number(m) | Messages per Mailbox |
|---|---|---|---|
| System Mailbox | 0 to 3 | 0 to 11 | 4 |
| HDVICP2-1Mailbox | 0 to 3 | 0 to 5 | 4 |
| HDVICP2-2 Mailbox | 0 to 3 | 0 to 5 | 4 |

Table 1-120 lists the memory-mapped registers for the Mailbox Registers. See the device-specific data manual for the memory address of these registers.

**Table 1-120. Mailbox Registers Mapping Summary**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 0000h | MAILBOX_REVISION | Mailbox Revision Register | Section 1.9.5.1 |
| 0010h | MAILBOX_SYSCONFIG | Mailbox System Configuration Register | Section 1.9.5.2 |
| 0040h + (4h * m) | MAILBOX_MESSAGE_m (1) | Mailbox Message Register | Section 1.9.5.3 |
| 0080h + (4h * m) | MAILBOX_FIFOSTATUS_m (1) | Mailbox FIFO Status Register | Section 1.9.5.4 |
| 00C0h + (4h * m) | MAILBOX_MSGSTATUS_m (1) | Mailbox Message Status Register | Section 1.9.5.5 |
| 0100h + (10h * u) | MAILBOX_IRQSTATUS_RAW_u (2) | Mailbox IRQ RAW Status Register | Section 1.9.5.6 |
| 0104h + (10h * u) | MAILBOX_IRQSTATUS_CLR_u (2) | Mailbox IRQ Clear Status Register | Section 1.9.5.7 |
| 0108h + (10h * u) | MAILBOX_IRQENABLE_SET_u (2) | Mailbox IRQ Enable Set Register | Section 1.9.5.8 |
| 010Ch + (10h * u) | MAILBOX_IRQENABLE_CLR_u (2) | Mailbox IRQ Enable Clear Register | Section 1.9.5.9 |
| 0140h | RESERVED | Reserved | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.9.5.1 Revision Register (MAILBOX_REVISION)

This register contains the IP revision code. The Mailbox Revision Register (MAILBOX_REVISION) is shown in Figure 1-71 described in Table 1-121.

**Figure 1-71. Revision Register (MAILBOX_REVISION)**

| 31 | 0 |
|---|---|
| REVISION | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-121. Revision Register (MAILBOX_REVISION) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | REVISION | 0-FFFF FFFFh | IP Revision |

### 1.9.5.2 System Configuration Register (MAILBOX_SYSCONFIG)

This register controls the various parameters of the communication interface. The Mailbox System Configuration Register (MAILBOX_SYSCONFIG) is shown in Figure 1-72 and described in Table 1-122.

**Figure 1-72. System Configuration Register (MAILBOX_SYSCONFIG)**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | SIDLEMODE | | Reserved | SOFTRESET |
| R-0 | | R/W-0 | | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-122. System Configuration Register (MAILBOX_SYSCONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3-2 | SIDLEMODE | 0 | Force-idle. An idle request is acknowledged unconditionally |
| | | 1 | No-idle. An idle request is never acknowledged |
| | | 2 | Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module based on the internal activity of the module |
| | | 3 | Reserved do not use |
| 1 | Reserved | 0 | Reserved |
| 0 | SOFTRESET | 0 | Soft/Hard reset done. |
| | | 1 | Reset is ongoing. |
| | | 0 | No action. |
| | | 1 | Start the soft reset sequence. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.9.5.3 Message Register (MAILBOX_MESSAGE_m)

The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue. The Mailbox Message Register (MAILBOX_MESSAGE_m) is shown in Figure 1-73 and described in Table 1-123.

**Figure 1-73. Message Register (MAILBOX_MESSAGE_m)**

| 31 | 0 |
|---|---|
| MESSAGEVALUEMBM | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-123. Message Register (MAILBOX_MESSAGE_m) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | MESSAGEVALUEMBM | 0-FFFF FFFFh | Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue. |

### 1.9.5.4 FIFO Status Register (MAILBOX_FIFOSTATUS_m)

The FIFO status register has the status related to the mailbox internal FIFO. The Mailbox FIFO Status Register (MAILBOX_FIFOSTATUS_m) is shown in Figure 1-74 and described in Table 1-124.

**Figure 1-74. FIFO Status Register (MAILBOX_FIFOSTATUS_m)**

| 31 | 1 | 0 |
|---|---|---|
| MESSAGEVALUEMBM | | FIFOFULLMBM |
| R/W-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-124. FIFO Status Register (MAILBOX_FIFOSTATUS_m) Field Descriptions**

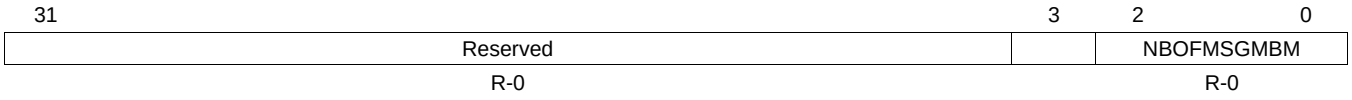
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | MESSAGEVALUEMBM | 0-7FFF FFFFh | Message in Mailbox. The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue. |
| 0 | FIFOFULLMBM | | Full flag for Mailbox |
| | | 0 | Mailbox FIFO is not full |
| | | 1 | Mailbox FIFO is full |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.9.5.5 Message Status Register (MAILBOX_MSGSTATUS_m)

The message status register has the status of the messages in the mailbox. The Mailbox Message Status Register (MAILBOX_FIFOSTATUS_m) is shown in Figure 1-75 and described in Table 1-125.

**Figure 1-75. Message Status Register (MAILBOX_MSGSTATUS_m)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-125. Message Status Register (MAILBOX_MSGSTATUS_m) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved. |
| 2-0 | NBOFMSGMBM | 0-7h | Number of unread messages in Mailbox. Limited to four messages per mailbox. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.9.5.6 IRQ RAW Status Register (MAILBOX_IRQSTATUS_RAW_u)

The interrupt status register has the raw status for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit sets this bit. This register is mainly used for debug purpose. The Mailbox IRQ RAW Status Register (MAILBOX_IRQSTATUS_RAW_u) is shown in Figure 1-76 and described in Table 1-126.

**Figure 1-76. IRQ RAW Status Register (MAILBOX_IRQSTATUS_RAW_u)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB11 | NEWMSGSTATUSUUMB11 | NOTFULLSTATUSUUMB10 | NEWMSGSTATUSUUMB10 | NOTFULLSTATUSUUMB9 | NEWMSGSTATUSUUMB9 | NOTFULLSTATUSUUMB8 | NEWMSGSTATUSUUMB8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB7 | NEWMSGSTATUSUUMB7 | NOTFULLSTATUSUUMB6 | NEWMSGSTATUSUUMB6 | NOTFULLSTATUSUUMB5 | NEWMSGSTATUSUUMB5 | NOTFULLSTATUSUUMB4 | NEWMSGSTATUSUUMB4 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB3 | NEWMSGSTATUSUUMB3 | NOTFULLSTATUSUUMB2 | NEWMSGSTATUSUUMB2 | NOTFULLSTATUSUUMB1 | NEWMSGSTATUSUUMB1 | NOTFULLSTATUSUUMB0 | NEWMSGSTATUSUUMB0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-126. IRQ RAW Status Register (MAILBOX_IRQSTATUS_RAW_u) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Reserved | 0 | Reserved |
| 23 | NOTFULLSTATUSUUMB11 | | Not Full Status bit for User u, Mailbox 11 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 22 | NEWMSGSTATUSUUMB11 | | New Message Status bit for User u, Mailbox 11 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 21 | NOTFULLSTATUSUUMB10 | | Not Full Status bit for User u, Mailbox 10 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 20 | NEWMSGSTATUSUUMB10 | | New Message Status bit for User u, Mailbox 10 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-126. IRQ RAW Status Register (MAILBOX_IRQSTATUS_RAW_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 19 | NOTFULLSTATUSUUMB9 | | Not Full Status bit for User u, Mailbox 9 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 18 | NEWMSGSTATUSUUMB9 | | New Message Status bit for User u, Mailbox 9 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 17 | NOTFULLSTATUSUUMB8 | | Not Full Status bit for User u, Mailbox 8 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 16 | NEWMSGSTATUSUUMB8 | | New Message Status bit for User u, Mailbox 8 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 15 | NOTFULLSTATUSUUMB7 | | Not Full Status bit for User u, Mailbox 7 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 14 | NEWMSGSTATUSUUMB7 | | New Message Status bit for User u, Mailbox 7 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 13 | NOTFULLSTATUSUUMB6 | | Not Full Status bit for User u, Mailbox 6 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 12 | NEWMSGSTATUSUUMB6 | | New Message Status bit for User u, Mailbox 6 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 11 | NOTFULLSTATUSUUMB5 | | Not Full Status bit for User u, Mailbox 5 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-126. IRQ RAW Status Register (MAILBOX_IRQSTATUS_RAW_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 10 | NEWMSGSTATUSUUMB5 | | New Message Status bit for User u, Mailbox 5 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 9 | NOTFULLSTATUSUUMB4 | | Not Full Status bit for User u, Mailbox 4 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 8 | NEWMSGSTATUSUUMB4 | | New Message Status bit for User u, Mailbox 4 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 7 | NOTFULLSTATUSUUMB3 | | Not Full Status bit for User u, Mailbox 3 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 6 | NEWMSGSTATUSUUMB3 | | New Message Status bit for User u, Mailbox 3 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 5 | NOTFULLSTATUSUUMB2 | | Not Full Status bit for User u, Mailbox 2 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 4 | NEWMSGSTATUSUUMB2 | | New Message Status bit for User u, Mailbox 2 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 3 | NOTFULLSTATUSUUMB1 | | Not Full Status bit for User u, Mailbox 1 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 2 | NEWMSGSTATUSUUMB1 | | New Message Status bit for User u, Mailbox 1 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-126. IRQ RAW Status Register (MAILBOX_IRQSTATUS_RAW_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | NOTFULLSTATUSUUMB0 | | Not Full Status bit for User u, Mailbox 0 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 0 | NEWMSGSTATUSUUMB0 | | New Message Status bit for User u, Mailbox 0 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.9.5.7 IRQ Clear Status Register (MAILBOX_IRQSTATUS_CLR_u)

The interrupt status register has the status combined with irq-enable for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit. The Mailbox IRQ Clear Status Register (MAILBOX_IRQSTATUS_CLR_u) is shown in Figure 1-77 and described in Table 1-127.

**Figure 1-77. IRQ Clear Status Register (MAILBOX_IRQSTATUS_CLR_u)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB11 | NEWMSGSTATUSUUMB11 | NOTFULLSTATUSUUMB10 | NEWMSGSTATUSUUMB10 | NOTFULLSTATUSUUMB9 | NEWMSGSTATUSUUMB9 | NOTFULLSTATUSUUMB8 | NEWMSGSTATUSUUMB8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB7 | NEWMSGSTATUSUUMB7 | NOTFULLSTATUSUUMB6 | NEWMSGSTATUSUUMB6 | NOTFULLSTATUSUUMB5 | NEWMSGSTATUSUUMB5 | NOTFULLSTATUSUUMB4 | NEWMSGSTATUSUUMB4 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB3 | NEWMSGSTATUSUUMB3 | NOTFULLSTATUSUUMB2 | NEWMSGSTATUSUUMB2 | NOTFULLSTATUSUUMB1 | NEWMSGSTATUSUUMB1 | NOTFULLSTATUSUUMB0 | NEWMSGSTATUSUUMB0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-127. IRQ Clear Status Register (MAILBOX_IRQSTATUS_CLR_u) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Reserved | 0 | Reserved. |
| 23 | NOTFULLSTATUSUUMB11 | | Not Full Status bit for User u, Mailbox 11 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 22 | NEWMSGSTATUSUUMB11 | | New Message Status bit for User u, Mailbox 11 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 21 | NOTFULLSTATUSUUMB10 | | Not Full Status bit for User u, Mailbox 10 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 20 | NEWMSGSTATUSUUMB10 | | New Message Status bit for User u, Mailbox 10 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-127. IRQ Clear Status Register (MAILBOX_IRQSTATUS_CLR_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 19 | NOTFULLSTATUSUUMB9 | | Not Full Status bit for User u, Mailbox 9 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 18 | NEWMSGSTATUSUUMB9 | | New Message Status bit for User u, Mailbox 9 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 17 | NOTFULLSTATUSUUMB8 | | Not Full Status bit for User u, Mailbox 8 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 16 | NEWMSGSTATUSUUMB8 | | New Message Status bit for User u, Mailbox 8 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 15 | NOTFULLSTATUSUUMB7 | | Not Full Status bit for User u, Mailbox 7 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 14 | NEWMSGSTATUSUUMB7 | | New Message Status bit for User u, Mailbox 7 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 13 | NOTFULLSTATUSUUMB6 | | Not Full Status bit for User u, Mailbox 6 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 12 | NEWMSGSTATUSUUMB6 | | New Message Status bit for User u, Mailbox 6 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 11 | NOTFULLSTATUSUUMB5 | | Not Full Status bit for User u, Mailbox 5 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-127. IRQ Clear Status Register (MAILBOX_IRQSTATUS_CLR_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 10 | NEWMSGSTATUSUUMB5 | | New Message Status bit for User u, Mailbox 5 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 9 | NOTFULLSTATUSUUMB4 | | Not Full Status bit for User u, Mailbox 4 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 8 | NEWMSGSTATUSUUMB4 | | New Message Status bit for User u, Mailbox 4 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 7 | NOTFULLSTATUSUUMB3 | | Not Full Status bit for User u, Mailbox 3 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 6 | NEWMSGSTATUSUUMB3 | | New Message Status bit for User u, Mailbox 3 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 5 | NOTFULLSTATUSUUMB2 | | Not Full Status bit for User u, Mailbox 2 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 4 | NEWMSGSTATUSUUMB2 | | New Message Status bit for User u, Mailbox 2 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 3 | NOTFULLSTATUSUUMB1 | | Not Full Status bit for User u, Mailbox 1 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 2 | NEWMSGSTATUSUUMB1 | | New Message Status bit for User u, Mailbox 1 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-127. IRQ Clear Status Register (MAILBOX_IRQSTATUS_CLR_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | NOTFULLSTATUSUUMB0 | | Not Full Status bit for User u, Mailbox 0 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 0 | NEWMSGSTATUSUUMB0 | | New Message Status bit for User u, Mailbox 0 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.9.5.8 IRQ Enable Set Register (MAILBOX_IRQENABLE_SET_u)

The interrupt enable register enables to unmask the module internal source of interrupt to the corresponding user. This register is write 1 to set. The Mailbox IRQ Enable Set Register (MAILBOX_IRQENABLE_SET_u) is shown in Figure 1-78 and described in Table 1-128.

**Figure 1-78. IRQ Enable Set Register (MAILBOX_IRQENABLE_SET_u)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB11 | NEWMSGSTATUSUUMB11 | NOTFULLSTATUSUUMB10 | NOTFULLSTATUSUUMB10 | NOTFULLSTATUSUUMB9 | NEWMSGSTATUSUUMB9 | NOTFULLSTATUSUUMB8 | NEWMSGSTATUSUUMB8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB7 | NEWMSGSTATUSUUMB7 | NOTFULLSTATUSUUMB6 | NEWMSGSTATUSUUMB6 | NOTFULLSTATUSUUMB5 | NEWMSGSTATUSUUMB5 | NOTFULLSTATUSUUMB4 | NEWMSGSTATUSUUMB4 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB3 | NEWMSGSTATUSUUMB3 | NOTFULLSTATUSUUMB2 | NEWMSGSTATUSUUMB2 | NEWMSGSTATUSUUMB1 | NEWMSGSTATUSUUMB1 | NOTFULLSTATUSUUMB0 | NEWMSGSTATUSUUMB0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-128. IRQ Enable Set Register (MAILBOX_IRQENABLE_SET_u) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Reserved | 0 | Reserved |
| 23 | NOTFULLSTATUSUUMB11 | | Not Full Status bit for User u, Mailbox 11 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 22 | NEWMSGSTATUSUUMB11 | | New Message Status bit for User u, Mailbox 11 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 21 | NOTFULLSTATUSUUMB10 | | Not Full Status bit for User u, Mailbox 10 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 20 | NEWFULLSTATUSUUMB10 | | New Message Status bit for User u, Mailbox 10 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 19 | NOTFULLSTATUSUUMB9 | | Not Full Status bit for User u, Mailbox 9 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-128. IRQ Enable Set Register (MAILBOX_IRQENABLE_SET_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 18 | NEWMSGSTATUSUUMB9 | | New Message Status bit for User u, Mailbox 9 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 17 | NOTFULLSTATUSUUMB8 | | Not Full Status bit for User u, Mailbox 8 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 16 | NEWMSGSTATUSUUMB8 | | New Message Status bit for User u, Mailbox 8 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 15 | NOTFULLSTATUSUUMB7 | | Not Full Status bit for User u, Mailbox 7 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 14 | NEWMSGSTATUSUUMB7 | | New Message Status bit for User u, Mailbox 7 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 13 | NOTFULLSTATUSUUMB6 | | Not Full Status bit for User u, Mailbox 6 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 12 | NEWMSGSTATUSUUMB6 | | New Message Status bit for User u, Mailbox 6 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 11 | NOTFULLSTATUSUUMB5 | | Not Full Status bit for User u, Mailbox 5 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 10 | NEWMSGSTATUSUUMB5 | | New Message Status bit for User u, Mailbox 5 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-128. IRQ Enable Set Register (MAILBOX_IRQENABLE_SET_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 9 | NOTFULLSTATUSUUMB4 | | Not Full Status bit for User u, Mailbox 4 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 8 | NEWMSGSTATUSUUMB4 | | New Message Status bit for User u, Mailbox 4 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 7 | NOTFULLSTATUSUUMB3 | | Not Full Status bit for User u, Mailbox 3 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 6 | NEWMSGSTATUSUUMB3 | | New Message Status bit for User u, Mailbox 3 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 5 | NOTFULLSTATUSUUMB2 | | Not Full Status bit for User u, Mailbox 2 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 4 | NEWMSGSTATUSUUMB2 | | New Message Status bit for User u, Mailbox 2 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 3 | NOTFULLSTATUSUUMB1 | | Not Full Status bit for User u, Mailbox 1 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 2 | NEWMSGSTATUSUUMB1 | | New Message Status bit for User u, Mailbox 1 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 1 | NOTFULLSTATUSUUMB0 | | Not Full Status bit for User u, Mailbox 0 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-128. IRQ Enable Set Register (MAILBOX_IRQENABLE_SET_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | NEWMSGSTATUSUUMB0 | | New Message Status bit for User u, Mailbox 0 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.9.5.9 IRQ Enable Clear Register (MAILBOX_IRQENABLE_CLR_u)

The interrupt enable register enables to mask the module internal source of interrupt to the corresponding user. This register is write 1 to clear. The IRQ Enable Clear Register (MAILBOX_IRQENABLE_CLR_u) is shown in Figure 1-79 and described in Table 1-129.

**Figure 1-79. IRQ Enable Clear Register (MAILBOX_IRQENABLE_CLR_u)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB11 | NEWMSGSTATUSUUMB11 | NOTFULLSTATUSUUMB10 | NEWMSGSTATUSUUMB10 | NOTFULLSTATUSUUMB9 | NEWMSGSTATUSUUMB9 | NOTFULLSTATUSUUMB8 | NEWMSGSTATUSUUMB8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB7 | NEWMSGSTATUSUUMB7 | NOTFULLSTATUSUUMB6 | NEWMSGSTATUSUUMB6 | NOTFULLSTATUSUUMB5 | NEWMSGSTATUSUUMB5 | NOTFULLSTATUSUUMB4 | NEWMSGSTATUSUUMB4 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NOTFULLSTATUSUUMB3 | NEWMSGSTATUSUUMB3 | NOTFULLSTATUSUUMB2 | NEWMSGSTATUSUUMB2 | NOTFULLSTATUSUUMB1 | NEWMSGSTATUSUUMB1 | NOTFULLSTATUSUUMB0 | NEWMSGSTATUSUUMB0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-129. IRQ Enable Clear Register (MAILBOX_IRQENABLE_CLR_u) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Reserved | 0 | Reserved |
| 23 | NOTFULLSTATUSUUMB11 | | Not Full Status bit for User u, Mailbox 11 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 22 | NEWMSGSTATUSUUMB11 | | New Message Status bit for User u, Mailbox 11 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 21 | NOTFULLSTATUSUUMB10 | | Not Full Status bit for User u, Mailbox 10 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 20 | NEWMSGSTATUSUUMB10 | | New Message Status bit for User u, Mailbox 10 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 19 | NOTFULLSTATUSUUMB9 | | Not Full Status bit for User u, Mailbox 9 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-129. IRQ Enable Clear Register (MAILBOX_IRQENABLE_CLR_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 18 | NEWMSGSTATUSUUMB9 | | New Message Status bit for User u, Mailbox 9 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 17 | NOTFULLSTATUSUUMB8 | | Not Full Status bit for User u, Mailbox 8 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 16 | NEWMSGSTATUSUUMB8 | | New Message Status bit for User u, Mailbox 8 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 15 | NOTFULLSTATUSUUMB7 | | Not Full Status bit for User u, Mailbox 7 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 14 | NEWMSGSTATUSUUMB7 | | New Message Status bit for User u, Mailbox 7 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 13 | NOTFULLSTATUSUUMB6 | | Not Full Status bit for User u, Mailbox 6 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 12 | NEWMSGSTATUSUUMB6 | | New Message Status bit for User u, Mailbox 6 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 11 | NOTFULLSTATUSUUMB5 | | Not Full Status bit for User u, Mailbox 5 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 10 | NEWMSGSTATUSUUMB5 | | New Message Status bit for User u, Mailbox 5 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-129. IRQ Enable Clear Register (MAILBOX_IRQENABLE_CLR_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 9 | NOTFULLSTATUSUUMB4 | | Not Full Status bit for User u, Mailbox 4 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 8 | NEWMSGSTATUSUUMB4 | | New Message Status bit for User u, Mailbox 4 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 7 | NOTFULLSTATUSUUMB3 | | Not Full Status bit for User u, Mailbox 3 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 6 | NEWMSGSTATUSUUMB3 | | New Message Status bit for User u, Mailbox 3 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 5 | NOTFULLSTATUSUUMB2 | | Not Full Status bit for User u, Mailbox 2 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 4 | NEWMSGSTATUSUUMB2 | | New Message Status bit for User u, Mailbox 2 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 3 | NOTFULLSTATUSUUMB1 | | Not Full Status bit for User u, Mailbox 1 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 2 | NEWMSGSTATUSUUMB1 | | New Message Status bit for User u, Mailbox 1 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |
| 1 | NOTFULLSTATUSUUMB0 | | Not Full Status bit for User u, Mailbox 0 |
| | | 0 | Read: No event pending (message queue full) |
| | | 1 | Read: Event pending (message queue not full) |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-129. IRQ Enable Clear Register (MAILBOX_IRQENABLE_CLR_u) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | NEWMSGSTATUSUUMB0 | | New Message Status bit for User u, Mailbox 0 |
| | | 0 | Read: No event (message) pending |
| | | 1 | Read: Event (message) pending |
| | | 0 | Write: No action |
| | | 1 | Write: Set the event (for debug) |


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.10 Spinlock

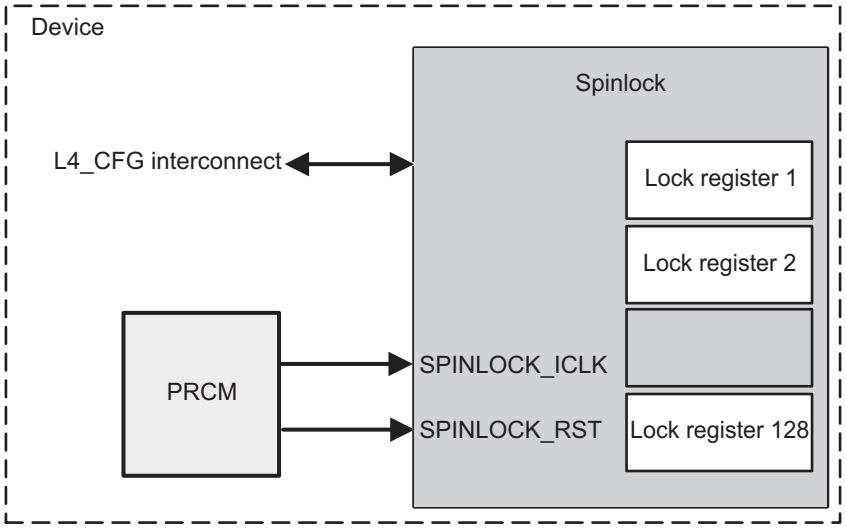
### 1.10.1 Overview

The Spinlock module provides hardware assistance for synchronizing the processes running on multiple processors in the device:

- Cortex-A8 microprocessor unit (MPU) subsystem
- Digital signal processor (DSP) subsystem
- Media Controller subsystem

The Spinlock module implements 128 spinlocks (or hardware semaphores), which provide an efficient way to perform a lock operation of a device resource using a single read-access, avoiding the need of a read-modify-write bus transfer that the programmable cores are not capable of.

**Figure 1-80. Spinlock Module**





Copyright © 2011–2016, Texas Instruments Incorporated

### 1.10.2 Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 1-81 shows the Spinlock integration.

**Figure 1-81. Spinlock Integration**

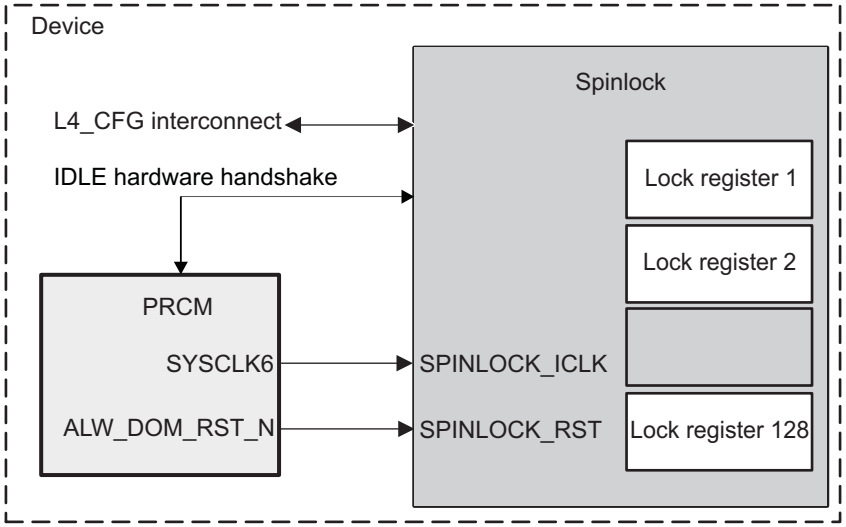



Table 1-130 and Table 1-131 summarize the integration of the module in the device.

**Table 1-130. Integration Attributes**

| Module Instance | Attributes | |
|---|---|---|
| | Power Domain | Interconnect |
| SPINLOCK | PD_ALWAYS_ON | L4_STANDARD |

**Table 1-131. Clocks and Resets**

| Clocks | | | | |
|---|---|---|---|---|
| Module Instance | Destination Signal Name | Source Signal Name | Source | Description |
| SPINLOCK | SPINLOCK_ICLK | SYSCLK6 | PRCM | Spinlock interface clock. This clock is used for all interface and functional operations. |
| **Resets** | | | | |
| Module Instance | Destination Signal Name | Source Signal Name | Source | Description |
| SPINLOCK | SPINLOCK_RST | ALW_DOM_RST_N | PRCM | Spinlock hardware reset. This reset is asynchronously applied to the Spinlock registers. |

The Spinlock module does not support any interrupt and DMA requests.


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.10.3 Functional Description

### 1.10.3.1 Software Reset

The Spinlock module can be reset by software through the SPINLOCK_SYSCONFIG[1] SOFTRESET bit. Setting this bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. This bit also indicates that the software reset is complete when its value is reset to 0. The software must ensure that the software reset completes before doing Spinlock operations.

### 1.10.3.2 Power Management

Table 1-132 describes power-management features available to the Spinlock module.

**Table 1-132. Local Power Management Features**

| Feature | Registers | Description |
|---|---|---|
| Clock auto gating | [0] AUTOGATING bit | This bit indicates that the module uses an automatic internal interface clock gating strategy, based on interface activity. |
| Slave idle modes | [4:3] SIDLEMODE bit field | This bit field indicates that the module uses smart-idle mode. |
| Clock activity | [8] CLOCKACTIVITY bit | This bit indicates that the interface clock is not required by the module during idle mode and may be switched off. |
| Global wake-up enable | [2] ENAWAKEUP bit | This bit indicates that the wake-up generation feature (at module level) is disabled. |

**NOTE:** All the local power management features are non-configurable (that is, their respective bits or bit fields are read-only).

**CAUTION**

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the CLOCKACTIVITY bit and Spinlock clock PRCM control bits.

The Spinlock module is normally idle, except when processing a request from its slave interface port. The smart-idle mode acknowledges idle requests from the PRCM only when the module is prepared to go idle. The Spinlock module is always ready to go idle if it does not have any request that it is processing.

The Spinlock module uses retention flops to retain state including the Taken state of each lock register. This means that the module can be placed in retention at any time when it is not processing a request and it is known that the system will not need to access the module.

Software must ensure to only power off the Spinlock module when no locks would be lost. In general, the steps to powering down the Spinlock module are:

- Check that all masters which might use the Spinlock module are either:
  - Already powered off
  - Notified that Spinlock is not available and the notification is acknowledged
- If desired, check that no locks are currently held in the Spinlock module. The status of each bank of 128 locks can be read from the register. If any locks are held, they are orphaned because they are not held by any master that is still active. Alternatively, you may decide to wait a timeout period to allow any active master to clean up its locks before powering down.
- The Spinlock module can now be powered off by writing the appropriate status to the PRCM.

In the case of powering off the whole system, these steps are unnecessary.


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.10.3.3 About Spinlocks

Spinlocks are present to solve the need for synchronization and mutual exclusion between heterogeneous processors and those not operating under a single, shared operating system. There is no alternative mechanism to accomplish these operations between processors in separate subsystems.

Spinlocks are not the best way to synchronize between tasks or threads on one CPU. Instead, spinlocks are for use in synchronization between different subsystems in the device that don't have any other means of hardware-based synchronization.

Spinlocks do not solve all system synchronization issues. They have limited applicability and should be used with care to implement higher level synchronization protocols.

A spinlock is appropriate for mutual exclusion for access to a shared data structure. It should be used only when:

- The time to hold the lock is predictable and small (for example, a maximum hold time of less than 200 CPU cycles may be acceptable).
- The locking task cannot be preempted, suspended, or interrupted while holding the lock (this would make the hold time large and unpredictable).
- The lock is lightly contended, that is the chance of any other process (or processor) trying to acquire the lock while it is held is small.

If these conditions are met, then the locking code can retry a failed attempt to acquire the lock until success.

If the conditions are not met, then a spinlock is not a good candidate. One alternative is to use a spinlock for critical section control (engineered to meet the conditions) to implement a higher level semaphore that can support preemption, notification, timeout or other higher level properties.

### 1.10.3.4 Functional Operation

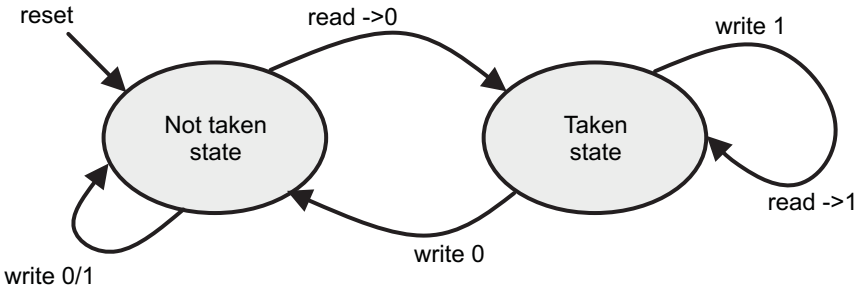
The Spinlock module supports 128 spinlocks. It accepts only a single command at a time and processes the command fully before accepting the next command. A lock is requested by reading the SPINLOCK_LOCK_REG_i[0] TAKEN bit. There are two states: Taken (SPINLOCK_LOCK_REG_i[0] TAKEN = 1) or Not Taken (SPINLOCK_LOCK_REG_i[0] TAKEN = 0).

When the status of lock i (where i = 0 to 127) is Not Taken (free), a read from the SPINLOCK_LOCK_REG_i register returns 0 and sets the lock to Taken (locked). When the status of lock i is Taken, a read returns 1 and does not change the state of the lock. A write to the SPINLOCK_LOCK_REG_i register does not change the state of lock, unless when writing 0 when the lock is in Taken state. By doing this, the requester frees the lock.

**CAUTION**

Only 32-bit reads and writes are supported.

**Figure 1-82. SPINLOCK_LOCK_REG_i Register State Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

## 1.10.4 Programming Guide

### 1.10.4.1 Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the module.

#### 1.10.4.1.1 Surrounding Modules Global Initialization

This procedure initializes the surrounding modules when the Spinlock module is used for the first time after a device reset.

#### 1.10.4.1.2 Global Initialization of Surrounding Modules

**Table 1-133. Global Initialization of Surrounding Modules**

| Surrounding Modules | Comments |
|---|---|
| PRCM | Spinlock interface clock must be enabled. For more information, Refer PRCM User Guide. |
| Interconnect | For more information about the L4-Standard interconnect configuration, Refer Bus Topology. |

### 1.10.4.2 Basic Spinlock Operations

The main spinlock operations are:

- Clear all the Taken spinlocks (only after a system bug recovery)
- Take a spinlock
- Release spinlock

#### 1.10.4.2.1 Spinlocks Clearing After a System Bug Recovery

Module initialization (after reset) is not needed, except after system bug recovery. The following table presents the Spinlock initialization after a system bug recovery. Software should store 0 into each of the SPINLOCK_LOCK_REG_i registers at system startup to insure that all locks are initialized to Not Taken.

**Table 1-134. Spinlock System Bug Recovery**

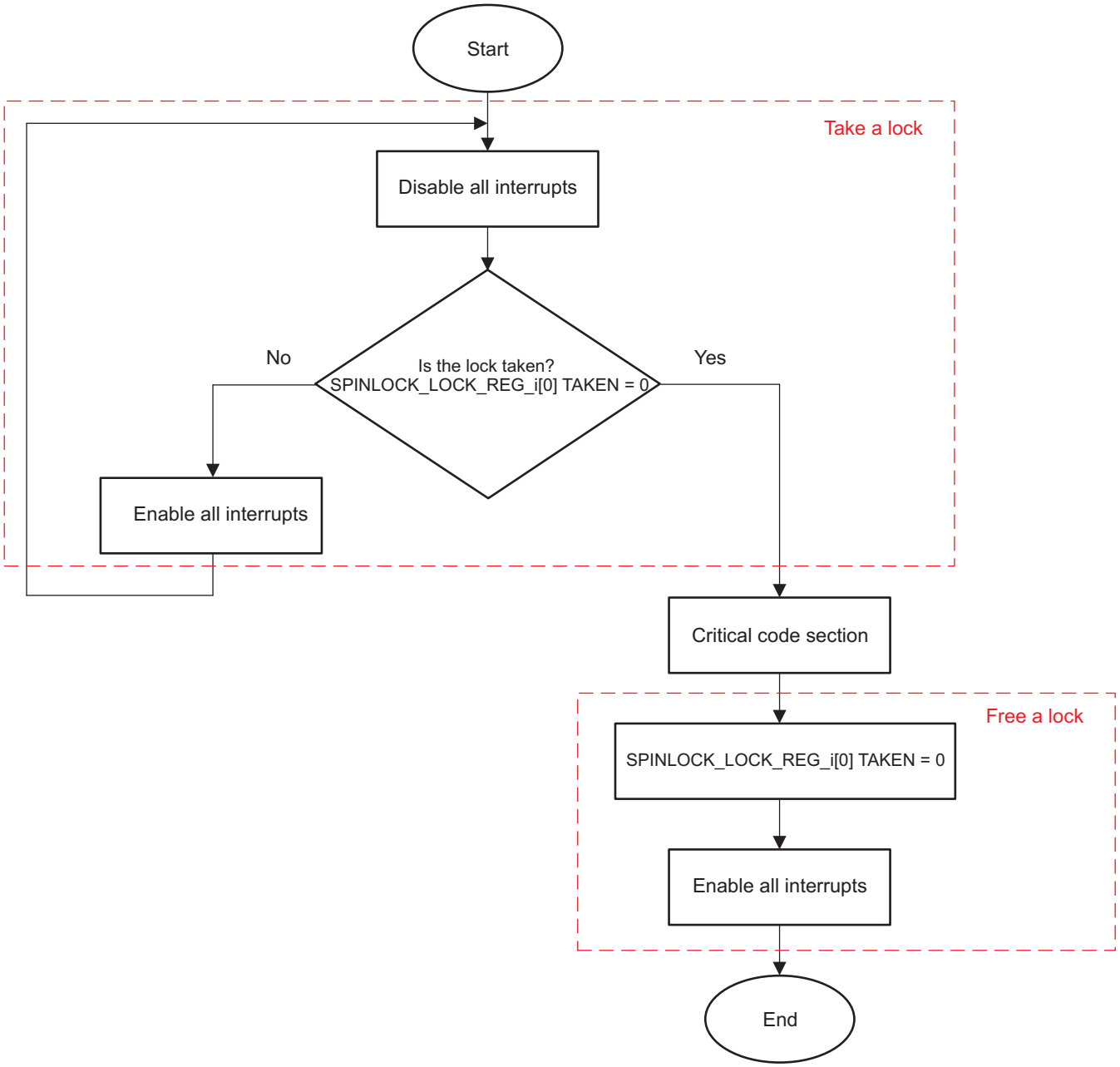
| Step | Register | Value |
|---|---|---|
| IF: SPINLOCK_SYSTATUS[0] IU0 = = 1? | SPINLOCK_SYSTATUS[0] IU0 | |
| Free the 128 locks | SPINLOCK_LOCK_REG_i[0] TAKEN (i=0 to 127) | 0 |
| END | | |

#### 1.10.4.2.2 Take and Release Spinlock

This procedure configures the take and release (free) operations for the Spinlock module. A spinlock should only be held with interrupts disabled. So, before attempting to obtain the spinlock, software should disable interrupts. Then it should read the SPINLOCK_LOCK_REG_i[0] TAKEN bit to attempt to obtain the lock. If it succeeds, it should proceed directly through the critical section then unlock and re-enable interrupts. If the acquisition attempt fails, the acquisition should be re-attempted. To prevent unknown interrupt disabled time, interrupts should be re-enabled and then disabled before re-attempting to acquire the lock. Figure 1-83 shows the described above procedure.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-83. Take and Release Spinlock**





Copyright © 2011–2016, Texas Instruments Incorporated

### 1.10.5 Spinlock Registers

Table 1-135 lists the memory-mapped registers for the Spin Lock Module. See the device-specific data manual for the memory address of these registers.

**Table 1-135. Spin Lock Module Registers Offset**

| Address Offset | Acronym | Description | Section |
|---|---|---|---|
| 00h | SPINLOCK_REV | Revision Register | Section 1.10.5.1 |
| 010h | SPINLOCK_SYSCFG | System Configuration Register | Section 1.10.5.2 |
| 014h | SPINLOCK_SYSSTAT | System Status Register | Section 1.10.5.3 |
| 800h + (4h × i) | SPINLOCK_LOCK_REG_i | Lock Register | Section 1.10.5.4 |

#### 1.10.5.1 Revision Register (SPINLOCK_REV)

The Spin Lock Revision Register (SPINLOCK_REV) is shown in Figure 1-84 and described in Table 1-136.

**Figure 1-84. Revision Register (SPINLOCK_REV)**

| 31 | 0 |
|---|---|
| REV | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-136. Revision Register (SPINLOCK_REV) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | REV | 0 | IP Revision Code. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.10.5.2 System Configuration Register (SPINLOCK_SYS_CFG)

The Spin Lock System Configuration Register (SPINLOCK_SYSCFG) is shown in Figure 1-85 described in Table 1-137.

**Figure 1-85. System Configuration Register (SPINLOCK_SYS_CFG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 9 | 8 | 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | CLOCKACTIVITY | Reserved | | SIDLEMODE | | ENWAKEUP | SOFTRESET | AUTOGATING |
| R-0 | | R-0 | R-0 | | R-0 | | R-0 | W-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-137. System Configuration Register (SPINLOCK_SYS_CFG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved |
| 8 | CLOCKACTIVITY | | Indicates whether the module requires the interface clock when in IDLE mode. |
| | | 0 | Interface clock is not required by the module during IDLE mode and may be switched off. |
| | | 1 | Interface clock is required by the module, even during idle mode. |
| 7-5 | Reserved | 0 | Reserved |
| 4-3 | SIDLEMODE | | Slave interface power management (IDLE request/acknowledgment control) |
| | | 0 | Force-idle. IDLE request is acknowledged unconditionally and immediately. |
| | | 1h | No-idle. IDLE request is never acknowledged. |
| | | 2h | Smart-idle. IDLE request acknowledgment is based on the internal module activity. |
| | | 3h | Reserved. Do not use. |
| 2 | ENWAKEUP | | Asynchronous wakeup generation |
| | | 0 | Wakeup generation is disabled. |
| | | 1 | Wakeup generation is enabled. |
| 1 | SOFTRESET | | Module software reset |
| | | 0 | No action |
| | | 1 | Start soft reset sequence |
| 0 | AUTOGATING | | Internal interface clock gating strategy. |
| | | 0 | Interface clock is not gated when the L4-STANDARD interface is idle. |
| | | 1 | Automatic internal interface clock gating strategy is applied, based on the L4-CFG interface activity. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.10.5.3 System Status Register (SPINLOCK_SYSSTAT)

The Spin Lock System Status Register (SPINLOCK_SYSSTAT) is shown in Figure 1-86 and described in Table 1-138.

**Figure 1-86. System Status Register (SPINLOCK_SYSSTAT)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| NUMLOCKS | | Reserved | |
| R-0 | | R-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| IU7 | IU6 | IU5 | IU4 | IU3 | IU2 | IU1 | IU0 | Reserved | | RESETDONE |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-138. System Status Register (SPINLOCK_SYSSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | NUMLOCKS | | Number of lock registers implemented. |
| | | 1h | This instance has 32 lock registers. |
| | | 2h | This instance has 64 lock registers. |
| | | 4h | This instance has 128 lock registers. |
| | | 8h | This instance has 256 lock registers. |
| 23-16 | Reserved | 0 | Reserved |
| 15 | IU7 | 0 | In-Use flag 7. Reads always return 0. |
| 14 | IU6 | 0 | In-Use flag 6. Reads always return 0. |
| 13 | IU5 | 0 | In-Use flag 5. Reads always return 0. |
| 12 | IU4 | 0 | In-Use flag 4. Reads always return 0. |
| 11 | IU3 | | In-Use flag 3, covering lock registers 96–127. |
| | | 0 | All lock registers 96–127 are in the Not Taken state. |
| | | 1 | At least one of the lock registers 96–127 is in the Taken state. |
| 10 | IU2 | | In-Use flag 2, covering lock registers 64–95. |
| | | 0 | All lock registers 64–95 are in the Not Taken state. |
| | | 1 | At least one of the lock registers 64–95 is in the Taken state. |
| 9 | IU1 | | In-Use flag 1, covering lock registers 0–31.In-Use flag 1, covering lock registers 32–63. |
| | | 0 | All lock registers 32–63 are in the Not Taken state. |
| | | 1 | At least one of the lock registers 32–63 is in the Taken state. |
| 8 | IU0 | | In-Use flag 0, covering lock registers 0–31. |
| | | 0 | All lock registers 0–31 are in the Not Taken state. |
| | | 1 | At least one of the lock registers 0–31 is in the Taken state. |
| 7-1 | Reserved | | Reserved |
| 0 | RESETDONE | | Reset done status. |
| | | 0 | Reset in progress. |
| | | 1 | Reset is completed. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.10.5.4 Lock Register (SPINLOCK_LOCK_REG_i)

The Spinlock Lock Register (SPINLOCK_LOCK_REG_i) is shown in Figure 1-87 and described in Table 1-139.

**NOTE:** i = 0 to 127

**Figure 1-87. Lock Register (SPINLOCK_LOCK_REG_i)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TAKEN |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-139. Lock Register (SPINLOCK_LOCK_REG_i) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | TAKEN | | Lock State |
| | | 0 | Lock was previously Not Taken (free). The requester is granted the lock. |
| | | 0 | Set the lock to Not Taken (free). |
| | | 1 | Lock was previously Taken. The requester is not granted the lock and must retry. |
| | | 1 | No update to the lock value. |


Copyright © 2011–2016, Texas Instruments Incorporated

## 1.11 Error Location Module

### 1.11.1 Error Location Module Overview

Non-managed NAND flash memories can be dense and nonvolatile in their own nature, but error-prone. When reading from NAND flash memories, some level of error-correction is required. In the case of NAND modules with no internal correction capability, sometimes referred to as *bare NANDs*, the correction process is delegated to the memory controller.

The general-purpose memory controller (GPMC) probes data read from an external NAND flash and uses this to compute checksum-like information, called syndrome polynomials, on a per-block basis. Each syndrome polynomial gives a status of the read operations for a full block, including 512 bytes of data, parity bits, and an optional spare-area data field, with a maximum block size of 1023 bytes. Computation is based on a Bose-ChaudhurI-Hocquenghem (BCH) algorithm. The error-location module (ELM) extracts error addresses from these syndrome polynomials.

Based on the syndrome polynomial value, the ELM can detect errors, compute the number of errors, and give the location of each error bit. The actual data is not required to complete the error-correction algorithm. Errors can be reported anywhere in the NAND flash block, including in the parity bits.

The maximum acceptable number of errors that can be corrected depends on a programmable configuration parameter. 4-, 8-, and 16-bit error-correction levels are supported. The ELM relies on a static and fixed definition of the generator polynomial for each error-correction level that corresponds to the generator polynomials defined in the GPMC (there are three fixed polynomial for the three correction error levels). A larger number of errors than the programmed error-correction level may be detected, but the ELM cannot correct them all. The offending block is then tagged as *uncorrectable* in the associated computation exit status register. If the computation is successful, that is, if the number of errors detected does not exceed the maximum value authorized for the chosen correction capability, the exit status register contains the information on the number of detected errors.

When the error-location process completes, an interrupt is triggered to inform the central processing unit (CPU) that its status can be checked. The number of detected errors and their locations in the NAND block can be retrieved from the module through register accesses.

#### 1.11.1.1 ELM Features

The ELM has the following features:

- 4, 8, and 16 bits per 512-byte block error-location based on BCH algorithms
- Eight simultaneous processing contexts
- Page-based and continuous modes
- Interrupt generation on error-location process completion:
  - When the full page has been processed in page mode
  - For each syndrome polynomial in continuous mode


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.2 ELM Integration

The ELM extracts error addresses from generated syndrome polynomials.

The ELM is used with the GPMC. Syndrome polynomials generated on-the-fly when reading a NAND flash page and stored in GPMC registers are passed to the ELM. The microprocessor unit (MPU) can then correct the data block by flipping the bits to which the ELM error-location outputs point.

Figure 1-88 shows the integration of the ELM subsystem in the device.

**Figure 1-88. ELM Integration**

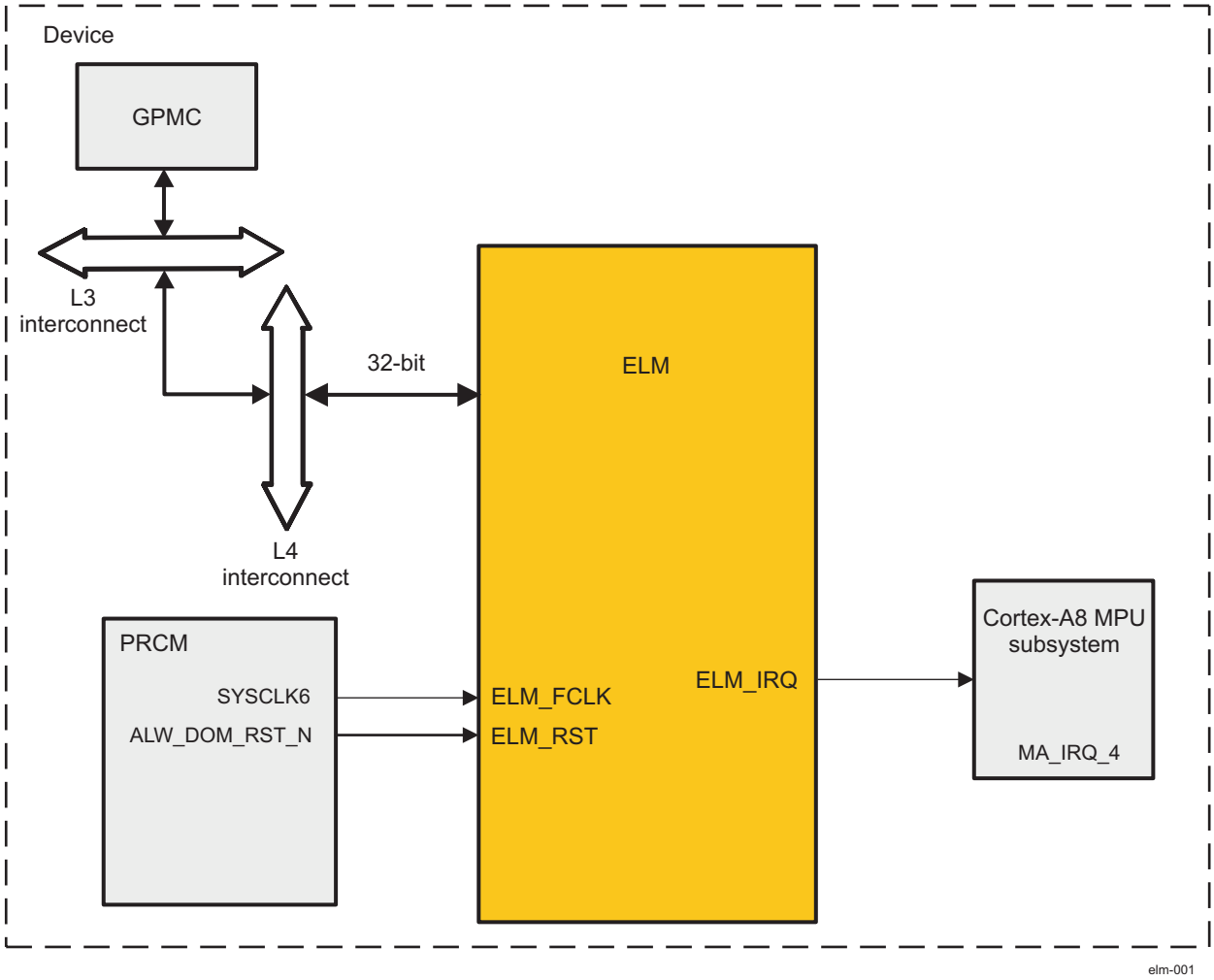



**Table 1-140. Integration Attributes**

| Module Instance | Attributes | | |
|---|---|---|---|
| | Power Domain | Wake-Up Capability | Interconnect |
| ELM | PD_ALWAYS_ON | No | NA |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-141. Clocks and Resets**

| Clocks | | | | |
|---|---|---|---|---|
| **Module Instance** | **Destination Signal Name** | **Source Signal Name** | **Source** | **Description** |
| ELM | ELM_FCLK | SYSCLK6 | PRCM | Functional clock |
| **Resets** | | | | |
| **Module Instance** | **Destination Signal Name** | **Source Signal Name** | **Source** | **Description** |
| ELM | ELM_RST | ALW_DOM_RST_N | PRCM | Power domain hardware reset |

**Table 1-142. Hardware Requests**

| Interrupt Requests | | | | |
|---|---|---|---|---|
| **Module Instance** | **Source Signal Name** | **Destination Signal Name** | **Destination** | **Description** |
| ELM | ELM_IRQ | MA_IRQ_4 | Cortex-A8 | BCH error-location module interrupt |

## 1.11.3 ELM Functional Description

The ELM is designed around the error-location engine, which handles the computation based on the input syndrome polynomials.

The ELM maps the error-location engine to a standard interconnect interface by using a set of registers to control inputs and outputs.

### 1.11.3.1 ELM Software Reset

To perform a software reset, write a 1 to the ELM_SYSCONFIG[1] SOFTRESET bit. The ELM_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the ELM_SYSCONFIG[1] SOFTRESET bit is automatically reset.

### 1.11.3.2 ELM Power Management

Table 1-143 describes the power-management features available to the ELM module.

**Table 1-143. Local Power Management Features**

| Feature | Registers | Description |
|---|---|---|
| Clock autogating | ELM_SYSCONFIG[0] AUTOGATING bit | This bit allows a local power optimization inside the module by gating the ELM_FCLK clock upon the interface activity. |
| Slave idle modes | ELM_SYSCONFIG[4:3] SIDLEMODE bit field | Force-idle, No-idle, and Smart-idle modes are available. |
| Clock activity | ELM_SYSCONFIG[8] CLOCKACTIVITY bit | The clock can be switched-off or maintained during the wake-up period. |
| Master Standby modes | N/A | |
| Global Wake-up Enable | N/A | |
| Wake-up Sources Enable | N/A | |


Copyright © 2011–2016, Texas Instruments Incorporated

**CAUTION**

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the ELM CLOCKACTIVITY and ELM clock PRCM control bits.

### 1.11.3.3 ELM Interrupt Requests

Table 1-144 lists the event flags, and their masks, that can cause module interrupts.

**Table 1-144. Events**

| Event Flag | Event Mask | Map to | Description |
|---|---|---|---|
| ELM_IRQSTATUS[8] PAGE_VALID | ELM_IRQENABLE[8] PAGE_MASK | ELM_IRQ | Page interrupt |
| ELM_IRQSTATUS[7] LOC_VALID_7 | ELM_IRQENABLE[7] LOCATION_MASK_7 | ELM_IRQ | Error-location interrupt for syndrome polynomial 7 |
| ELM_IRQSTATUS[6] LOC_VALID_6 | ELM_IRQENABLE[6] LOCATION_MASK_6 | ELM_IRQ | Error-location interrupt for syndrome polynomial 6 |
| ELM_IRQSTATUS[5] LOC_VALID_5 | ELM_IRQENABLE[5] LOCATION_MASK_5 | ELM_IRQ | Error-location interrupt for syndrome polynomial 5 |
| ELM_IRQSTATUS[4] LOC_VALID_4 | ELM_IRQENABLE[4] LOCATION_MASK_4 | ELM_IRQ | Error-location interrupt for syndrome polynomial 4 |
| ELM_IRQSTATUS[3] LOC_VALID_3 | ELM_IRQENABLE[3] LOCATION_MASK_3 | ELM_IRQ | Error-location interrupt for syndrome polynomial 3 |
| ELM_IRQSTATUS[2] LOC_VALID_2 | ELM_IRQENABLE[2] LOCATION_MASK_2 | ELM_IRQ | Error-location interrupt for syndrome polynomial 2 |
| ELM_IRQSTATUS[1] LOC_VALID_1 | ELM_IRQENABLE[1] LOCATION_MASK_1 | ELM_IRQ | Error-location interrupt for syndrome polynomial 1 |
| ELM_IRQSTATUS[0] LOC_VALID_0 | ELM_IRQENABLE[0] LOCATION_MASK_0 | ELM_IRQ | Error-location interrupt for syndrome polynomial 0 |

### 1.11.3.4 Processing Initialization

ELM_LOCATION_CONFIG global setting parameters must be set before using the error-location engine. The ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL bit defines the error-correction level used (4-,8-, or 16-bit error-correction). The ELM_LOCATION_CONFIG[26:16] ECC_SIZE bit field defines the maximum buffer length beyond which the engine processing no longer looks for errors.

The CPU can choose to use the ELM in continuous mode or page mode. If all ELM_PAGE_CTRL[i] SECTOR_i bits are reset (i is the syndrome polynomial number, i = 0 to 7), continuous mode is used. In any other case, page mode is implicitly selected.

- Continuous mode: Each syndrome polynomial is processed independently – results for a syndrome can be retrieved and acknowledged at any time, whatever the status of the other seven processing contexts.
- Page mode: Syndrome polynomials are grouped into atomic entities – only one page can be processed at any given time, even if all eight contexts are not used for this page. Unused contexts are lost and cannot be affected to any other processing. The full page must be acknowledged and cleared before moving to the next page.

For completion interrupts to be generated correctly, all ELM_IRQENABLE[i] LOCATION_MASK_i bits (i = 0 to 7) must be forced to 0 when in page mode, and set to 1 in continuous mode. Additionally, the ELM_IRQENABLE[8] PAGE_MASK bit must be set to 1 when in page mode.


Copyright © 2011–2016, Texas Instruments Incorporated

The CPU initiates error-location processing by writing a syndrome polynomial into one of the eight possible register sets. Each of these register sets includes seven registers: ELM_SYNDROME_FRAGMENT_0_i to ELM_SYNDROME_FRAGMENT_6_i. The first six registers can be written in any order, but ELM_SYNDROME_FRAGMENT_6_i must be written last because it includes the validity bit, which instructs the ELM that this syndrome polynomial must be processed (the ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit).

As soon as one validity bit is asserted (ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x1, with i = 0 to 7), error-location processing can start for the corresponding syndrome polynomial. The associated ELM_LOCATION_STATUS_i and ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i registers are not reset (i = 0 to 7). The software must not consider them until the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit is set.

#### 1.11.3.5 Processing Sequence

While the error-location engine is busy processing one syndrome polynomial, further syndrome polynomials can be written. They are processed when the current processing completes.

The engine completes early when:

- No error is detected; that is, when the ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE bit is set to 1 and the ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS bit field is set to 0x0.
- Too many errors are detected; that is, when the ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE bit is set to 0 while the ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS bit field is set with the value output by the error-location engine. The reported number of errors is not ensured if ECC_CORRECTABLE is 0.

If the engine completes early, the associated error-location registers ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i are not updated (i = 0 to 7).

In all other cases, the engine goes through the entire error-location process. Each time an error-location is found, it is logged in the associated ECC_ERROR_LOCATION bit field. The first error detected is logged in the ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION bit field; the second in the ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION bit field, and so on.

**Table 1-145. ELM_LOCATION_STATUS_i Value Decoding Table**

| ECC_CORRECTABLE Value | ECC_NB_ERRORS Value | Status | Number of Errors Detected | Action Required |
|---|---|---|---|---|
| 1 | 0 | OK | 0 | None |
| 1 | ≠ 0 | OK | ECC_NB_ERRORS | Correct the data buffer read based on the ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i results. |
| 0 | Any | Failed | Unknown | Software-dependant |

#### 1.11.3.6 Processing Completion

When the processing for a given syndrome polynomial completes, its ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit is reset. It must not be set again until the exit status registers, ELM_LOCATION_STATUS_i (i = 0 to 7), for this processing are checked. Failure to comply with this rule leads to potential loss of the first polynomial process data output.

The error-location engine signals the process completion to the ELM. When this event is detected, the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit is set (i = 0 to 7). The processing-exit status is available from the associated ELM_LOCATION_STATUS_i register, and error locations are stored in order in the ECC_ERROR_LOCATION fields. The software must only read valid error-location registers based on the number of errors detected and located.


Copyright © 2011–2016, Texas Instruments Incorporated

Immediately after the error-location engine completes, a new syndrome polynomial can be processed, if any is available, as reported by the ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID validity bit, depending on the configured error-correction level. If several syndrome polynomials are available, a round-robin arbitration is used to select one for processing.

In continuous mode (that is, all bits in ELM_PAGE_CTRL are reset), an interrupt is triggered whenever a ELM_IRQSTATUS[i] LOC_VALID_i bit is asserted. The CPU must read the ELM_IRQSTATUS register to determine which polynomial is processed and retrieve the exit status and error locations (ELM_LOCATION_STATUS_i and ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i). When done, the CPU must clear the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit by writing it to 1. Other status bits must be written to 0 so that other interrupts are not unintentionally cleared. When using this mode, the ELM_IRQSTATUS[8] PAGE_VALID interrupt is never triggered.

In page mode, the module does not trigger interrupts for the processing completion of each polynomial because the ELM_IRQENABLE[i] LOCATION_MASK_i bits are cleared. A page is defined using the ELM_PAGE_CTRL register. Each SECTOR_i bit set means the corresponding polynomial i is part of the page processing. A page is fully processed when all tagged polynomials have been processed, as logged in the ELM_IRQSTATUS[i] LOC_VALID_i bit fields. The module triggers an ELM_IRQSTATUS[8] PAGE_VALID interrupt whenever it detects that the full page has been processed. To make sure the next page can be correctly processed, all status bits in the ELM_IRQSTATUS register must be cleared by using a single atomic-write access.

**NOTE:** Do not modify page setting parameters in the ELM_PAGE_CTRL register unless the engine is idle, no polynomial input is valid, and all interrupts have been cleared.

Because no polynomial-level interrupt is triggered in page mode, polynomials cleared in the ELM_PAGE_CTRL[i] SECTOR_i bit fields (i = 0 to 7) are processed as usual, but are essentially ignored. The CPU must manually poll the ELM_IRQSTATUS bits to check for their status.

## *1.11.4 ELM Basic Programming Model*

### 1.11.4.1 ELM Low Level Programming Model

#### *1.11.4.1.1 Processing Initialization*

**Table 1-146. ELM Processing Initialization**

| Step | Register/ Bit Field / Programming Model | Value |
|---|---|---|
| Resets the module | ELM_SYSCONFIG[1] SOFTRESET | 0x1 |
| Wait until reset is done. | ELM_SYSSTATUS[0] RESETDONE | 0x1 |
| Configure the slave interface power management. | ELM_SYSCONFIG[4:3] SIDLEMODE | Set value |
| Defines the error-correction level used | ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL | Set value |
| Defines the maximum buffer length | ELM_LOCATION_CONFIG[26:16] ECC_SIZE | Set value |
| Sets the ELM in continuous mode or page mode | ELM_PAGE_CTRL | Set value |
| **If** continuous mode is used | All ELM_PAGE_CTRL[i] SECTOR_i (i = 0 to 7) | 0x0 |
| Enables interrupt for syndrome polynomial i | ELM_IRQENABLE[i] LOCATION_MASK_i | 0x1 |
| **else** (page mode is used) | One syndrome polynomial i is set ELM_PAGE_CTRL[i] SECTOR_i (i = 0 to 7) | 0x1 |
| Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt. | All ELM_IRQENABLE[i] LOCATION_MASK_i = 0x0 and ELM_IRQENABLE[8] PAGE_MASK = 0x1 | Set value |
| **endif** | | Set value |
| Set the input syndrome polynomial i. | ELM_SYNDROME_FRAGMENT_0_i | Set value |
| | ELM_SYNDROME_FRAGMENT_1_i | Set value |
| | ELM_SYNDROME_FRAGMENT_5_i | Set value |
| | ELM_SYNDROME_FRAGMENT_6_i | Set value |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-146. ELM Processing Initialization (continued)**

| Step | Register/ Bit Field / Programming Model | Value |
|---|---|---|
| Initiates the computation process | ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID | 0x1 |

#### 1.11.4.1.2 Read Results

The engine goes through the entire error-location process and results can be read. Table 1-147 and Table 1-148 describe the processing completion for continuous and page modes, respectively.

**Table 1-147. ELM Processing Completion for Continuous Mode**

| Step | Register/ Bit Field / Programming Model | Value |
|---|---|---|
| Wait until process is complete for syndrome polynomial i:<br>Wait until the ELM_IRQ interrupt is generated, or poll the status register. | | |
| Read for which i the error-location process is complete. | ELM_IRQSTATUS[i] LOC_VALID_i | 0x1 |
| **if** the process fails (too many errors) | ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE | 0x0 |
| It is software dependant. | | |
| **else** (process successful, the engine completes) | ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE | 0x1 |
| Read the number of errors. | ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS | |
| Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers.<br>It is the software responsibility to correct errors in the data buffer. | ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION | |
| | ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION | |
| | ... | |
| | ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION | |
| **endif** | | |
| Clear the corresponding i interrupt. | ELM_IRQSTATUS[i] LOC_VALID_i | 0x1 |

A new syndrome polynomial can be processed after the end of processing (ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x0) and after the exit status register check (ELM_LOCATION_STATUS_i).

**Table 1-148. ELM Processing Completion for Page Mode**

| Step | Register/ Bit Field / Programming Model | Value |
|---|---|---|
| Wait until process is complete for syndrome polynomial i:<br>Wait until the ELM_IRQ interrupt is generated, or poll the status register. | | |
| Wait for page completed interrupt:<br>All error locations are valid. | ELM_IRQSTATUS[8] PAGE_VALID | 0x1 |
| **Repeat** the following actions the necessary number of times. That is, once for each valid defined block in the page. | | |
| Read the process exit status. | ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE | |
| **if** the process fails (too many errors) | ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE | 0x0 |
| It is software dependant. | | |
| **else** (process successful, the engine completes) | ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE | 0x1 |
| Read the number of errors. | ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-148. ELM Processing Completion for Page Mode (continued)**

| Step | Register/ Bit Field / Programming Model | Value |
|---|---|---|
| Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers. | ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION | |
| | ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION | |
| | ... | |
| | ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION | |
| **endif** | | |
| **End Repeat** | | |
| Clear the ELM_IRQSTATUS register. | ELM_IRQSTATUS | 0x1FF |

Next page can be correctly processed after a page is fully processed, when all tagged polynomials have been processed (ELM_IRQSTATUS[i] LOC_VALID_i = 0x1 for all syndrome polynomials i used in the page).

### 1.11.4.2 Use Case: ELM Used in Continuous Mode

In this example, the ELM module is programmed for an 8-bit error-correction capability in continuous mode. After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, a non-zero polynomial syndrome is reported from the GPMC (Polynomial syndrome 0 is used in the ELM):

- P = 0x0A16ABE115E44F767BFB0D0980

**Table 1-149. Use Case: Continuous Mode**

| Step | Register/ Bit Field / Programming Model | Value |
|---|---|---|
| Resets the module | ELM_SYSCONFIG[1] SOFTRESET | 0x1 |
| Wait until reset is done. | ELM_SYSSTATUS[0] RESETDONE | 0x1 |
| Configure the slave interface power management: Smart idle is used. | ELM_SYSCONFIG[4:3] SIDLEMODE | 0x2 |
| Defines the error-correction level used: 8 bits | ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL | 0x1 |
| Defines the maximum buffer length: 528 bytes (2x528 = 1056) | ELM_LOCATION_CONFIG[26:16] ECC_SIZE | 0x420 |
| Sets the ELM in continuous mode | ELM_PAGE_CTRL | 0 |
| Enables interrupt for syndrome polynomial 0 | ELM_IRQENABLE[0] LOCATION_MASK_0 | 0x1 |
| Set the input syndrome polynomial 0. | ELM_SYNDROME_FRAGMENT_0_i (i=0) | 0xFB0D0980 |
| | ELM_SYNDROME_FRAGMENT_1_i (i=0) | 0xE44F767B |
| | ELM_SYNDROME_FRAGMENT_2_i (i=0) | 0x16ABE115 |
| | ELM_SYNDROME_FRAGMENT_3_i (i=0) | 0x0000000A |
| Initiates the computation process | ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=0) | 0x1 |
| Wait until process is complete for syndrome polynomial 0:<br>IRQ_ELM is generated or poll the status register. | | |
| Read that error-location process is complete for syndrome polynomial 0. | ELM_IRQSTATUS[0] LOC_VALID_0 | 0x1 |
| Read the process exit status:<br>All errors were successfully located. | ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=0) | 0x1 |
| Read the number of errors:<br>Four errors detected. | ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=0) | 0x4 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-149. Use Case: Continuous Mode (continued)**

| Step | Register/ Bit Field / Programming Model | Value |
|---|---|---|
| Read the error-location bit addresses for syndrome polynomial 0 of the 4 first registers:<br>Errors are located in the data buffer at decimal addresses 431, 1062, 1909, 3452. | ELM_ERROR_LOCATION_0_i (i=0) | 0x1AF |
| | ELM_ERROR_LOCATION_1_i (i=0) | 0x426 |
| | ELM_ERROR_LOCATION_2_i (i=0) | 0x775 |
| | ELM_ERROR_LOCATION_3_i (i=0) | 0xD7C |
| Clear the corresponding interrupt for polynomial 0. | ELM_IRQSTATUS[0] LOC_VALID_0 | 0x1 |

The NAND flash data in the sector are seen as a polynomial of degree 4223 (number of bits in a 528 byte buffer minus 1), with each data bit being a coefficient in the polynomial. When reading from a NAND flash using the GPMC module, computation of the polynomial syndrome assumes that the first NAND word read at address 0x0 contains the highest-order coefficient in the message. Furthermore, in the 16-bit NAND word, bits are ordered from bit 7 to bit 0, then from bit 15 to bit 8. Based on this convention, an address table of the data buffer can be built. NAND memory addresses in Table 1-150 are given in decimal format.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-150. 16-bit NAND Sector Buffer Address Map**

| NAND Memory Address | Message bit addresses in the memory word | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 4215 | 4214 | 4213 | 4212 | 4211 | 4210 | 4209 | 4208 | 4223 | 4222 | 4221 | 4220 | 4219 | 4218 | 4217 | 4216 |
| 1 | 4175 | 4174 | 4173 | 4172 | 4171 | 4170 | 4169 | 4168 | 4183 | 4182 | 4181 | 4180 | 4179 | 4178 | 4177 | 4176 |
| ... | | | | | | | | | | | | | | | | |
| 47 | 3463 | 3462 | 3461 | 3460 | 3459 | 3458 | 3457 | 3456 | 3471 | 3470 | 3469 | 3468 | 3467 | 3466 | 3465 | 3464 |
| 48 | 3447 | 3446 | 3445 | 3444 | 3443 | 3442 | 3441 | 3440 | 3455 | 3454 | 3453 | 3452 | 3451 | 3450 | 3449 | 3448 |
| 49 | 3431 | 3430 | 3429 | 3428 | 3427 | 3426 | 3425 | 3424 | 3439 | 3438 | 3437 | 3436 | 3435 | 3434 | 3433 | 3432 |
| 50 | 3415 | 3414 | 3413 | 3412 | 3411 | 3410 | 3409 | 3408 | 3423 | 3422 | 3421 | 3420 | 3419 | 3418 | 3417 | 3416 |
| ... | | | | | | | | | | | | | | | | |
| 255 | 135 | 134 | 133 | 132 | 131 | 130 | 129 | 128 | 143 | 142 | 141 | 140 | 139 | 138 | 137 | 136 |
| 256 | 119 | 118 | 117 | 116 | 115 | 114 | 113 | 112 | 127 | 126 | 125 | 124 | 123 | 122 | 121 | 120 |
| 257 | 103 | 102 | 101 | 100 | 99 | 98 | 97 | 96 | 111 | 110 | 109 | 108 | 107 | 106 | 105 | 104 |
| 258 | 87 | 86 | 85 | 84 | 83 | 82 | 81 | 80 | 95 | 94 | 93 | 92 | 91 | 90 | 89 | 88 |
| 259 | 71 | 70 | 69 | 68 | 67 | 66 | 65 | 64 | 79 | 78 | 77 | 76 | 75 | 74 | 73 | 72 |
| 260 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 |
| 261 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 |
| 262 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 263 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |

The table can now be used to determine which bits in the buffer were incorrect and must be flipped. In this example, the first bit to be flipped is bit 4 from the 49th byte read from memory. It is up to the processor to correctly map this word to the copied buffer and to flip this bit. The same process must be repeated for all detected errors.

### 1.11.4.3 Use Case: ELM Used in Page Mode

In this example, the ELM module is programmed for an 16-bit error-correction capability in page mode. After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, four non-zero polynomial syndromes are reported from the GPMC (Polynomial syndrome 0, 1, 2, and 3 are used in the ELM):

- P0 = 0xE8B0 12ADDB5A318E05BE B0693DB28330B5CC A329AA05E0B718EF
- P1 = 0xBAD0 49A0D932C22E6669 0948DF08BE093336 79C6BA10E5F935EB
- P2 = 0x69D9 B86ABCD5EC3697FA A6498FEE54556EA0 1579EF7D60BA3189
- P3 = 0x0

**Table 1-151. Use Case: Page Mode**

| Step | Register/ Bit Field / Programming Model | Value |
|---|---|---|
| Resets the module | ELM_SYSCONFIG[1] SOFTRESET | 0x1 |
| Wait until reset is done. | ELM_SYSSTATUS[0] RESETDONE | 0x1 |
| Configure the slave interface power management: Smart idle is used. | ELM_SYSCONFIG[4:3] SIDLEMODE | 0x2 |
| Defines the error-correction level used: 16 bits | ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL | 0x2 |
| Defines the maximum buffer length: 528 bytes | ELM_LOCATION_CONFIG[26:16] ECC_SIZE | 0x420 |
| Sets the ELM in page mode (4 blocks in a page) | ELM_PAGE_CTRL[0] SECTOR_0 | 0x1 |
| | ELM_PAGE_CTRL[1] SECTOR_1 | 0x1 |
| | ELM_PAGE_CTRL[2] SECTOR_2 | 0x1 |
| | ELM_PAGE_CTRL[3] SECTOR_3 | 0x1 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-151. Use Case: Page Mode (continued)**

| Step | Register/ Bit Field / Programming Model | Value |
|---|---|---|
| Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt. | ELM_IRQENABLE | 0x100 |
| Set the input syndrome polynomial 0. | ELM_SYNDROME_FRAGMENT_0_i (i=0) | 0xE0B718EF |
| | ELM_SYNDROME_FRAGMENT_1_i (i=0) | 0xA329AA05 |
| | ELM_SYNDROME_FRAGMENT_2_i (i=0) | 0x8330B5CC |
| | ELM_SYNDROME_FRAGMENT_3_i (i=0) | 0xB0693DB2 |
| | ELM_SYNDROME_FRAGMENT_4_i (i=0) | 0x318E05BE |
| | ELM_SYNDROME_FRAGMENT_5_i (i=0) | 0x12ADDB5A |
| | ELM_SYNDROME_FRAGMENT_6_i (i=0) | 0xE8B0 |
| Set the input syndrome polynomial 1. | ELM_SYNDROME_FRAGMENT_0_i (i=1) | 0xE5F935EB |
| | ELM_SYNDROME_FRAGMENT_1_i (i=1) | 0x79C6BA10 |
| | ELM_SYNDROME_FRAGMENT_2_i (i=1) | 0xBE093336 |
| | ELM_SYNDROME_FRAGMENT_3_i (i=1) | 0x0948DF08 |
| | ELM_SYNDROME_FRAGMENT_4_i (i=1) | 0xC22E6669 |
| | ELM_SYNDROME_FRAGMENT_5_i (i=1) | 0x49A0D932 |
| | ELM_SYNDROME_FRAGMENT_6_i (i=1) | 0xBAD0 |
| Set the input syndrome polynomial 2. | ELM_SYNDROME_FRAGMENT_0_i (i=2) | 0x60BA3189 |
| | ELM_SYNDROME_FRAGMENT_1_i (i=2) | 0x1579EF7D |
| | ELM_SYNDROME_FRAGMENT_2_i (i=2) | 0x54556EA0 |
| | ELM_SYNDROME_FRAGMENT_3_i (i=2) | 0xA6498FEE |
| | ELM_SYNDROME_FRAGMENT_4_i (i=2) | 0xEC3697FA |
| | ELM_SYNDROME_FRAGMENT_5_i (i=2) | 0xB86ABCD5 |
| | ELM_SYNDROME_FRAGMENT_6_i (i=2) | 0x69D9 |
| Set the input syndrome polynomial 3. | ELM_SYNDROME_FRAGMENT_0_i (i=3) | 0x0 |
| | ELM_SYNDROME_FRAGMENT_1_i (i=3) | 0x0 |
| | ELM_SYNDROME_FRAGMENT_2_i (i=3) | 0x0 |
| | ELM_SYNDROME_FRAGMENT_3_i (i=3) | 0x0 |
| | ELM_SYNDROME_FRAGMENT_4_i (i=3) | 0x0 |
| | ELM_SYNDROME_FRAGMENT_5_i (i=3) | 0x0 |
| | ELM_SYNDROME_FRAGMENT_6_i (i=3) | 0x0 |
| Initiates the computation process for syndrome polynomial 0 | ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=0) | 0x1 |
| Initiates the computation process for syndrome polynomial 1 | ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=1) | 0x1 |
| Initiates the computation process for syndrome polynomial 2 | ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=2) | 0x1 |
| Initiates the computation process for syndrome polynomial 3 | ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=3) | 0x1 |
| Wait until process is complete for syndrome polynomial 0, 1, 2, and 3:<br>Wait until the ELM_IRQ interrupt is generated or poll the status register. | | |
| Wait for page completed interrupt:<br>All error locations are valid. | ELM_IRQSTATUS[8] PAGE_VALID | 0x1 |
| Read the process exit status for syndrome polynomial 0:<br>All errors were successfully located. | ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=0) | 0x1 |
| Read the process exit status for syndrome polynomial 1:<br>All errors were successfully located. | ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=1) | 0x1 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-151. Use Case: Page Mode (continued)**

| Step | Register/ Bit Field / Programming Model | Value |
|---|---|---|
| Read the process exit status for syndrome polynomial 2:<br>All errors were successfully located. | ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=2) | 0x1 |
| Read the process exit status for syndrome polynomial 3:<br>All errors were successfully located. | ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=3) | 0x1 |
| Read the number of errors for syndrome polynomial 0:<br>4 errors detected. | ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=0) | 0x4 |
| Read the number of errors for syndrome polynomial 1:<br>2 errors detected. | ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=1) | 0x2 |
| Read the number of errors for syndrome polynomial 2:<br>1 error detected. | ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=2) | 0x1 |
| Read the number of errors for syndrome polynomial 3:<br>0 errors detected. | ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=3) | 0x0 |
| Read the error-location bit addresses for syndrome polynomial 0 of the 4 first registers: | ELM_ERROR_LOCATION_0_i (i=0) | 0x1FE |
| | ELM_ERROR_LOCATION_1_i (i=0) | 0x617 |
| | ELM_ERROR_LOCATION_2_i (i=0) | 0x650 |
| | ELM_ERROR_LOCATION_3_i (i=0) | 0xA83 |
| Read the error-location bit addresses for syndrome polynomial 1 of the 2 first registers: | ELM_ERROR_LOCATION_0_i (i=1) | 0x4 |
| | ELM_ERROR_LOCATION_1_i (i=1) | 0x1036 |
| Read the errors location bit addresses for syndrome polynomial 2 of the first registers: | ELM_ERROR_LOCATION_0_i (i=1) | 0x3E8 |
| Clear the ELM_IRQSTATUS register. | ELM_IRQSTATUS | 0x1FF |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5 ELM Registers

**Table 1-152. ELM Registers Mapping Summary**

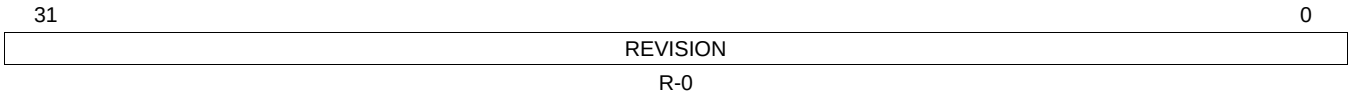
| Address Offset | Acronym | Description | Section |
|---|---|---|---|
| 000 | ELM_REVISION | ELM Revision Register | Section 1.11.5.1 |
| 010h | ELM_SYSCONFIG | ELM System Configuration Register | Section 1.11.5.2 |
| 014h | ELM_SYSSTATUS | ELM System Status Register | Section 1.11.5.3 |
| 018h | ELM_IRQSTATUS | ELM Interrupt Status Register | Section 1.11.5.4 |
| 01Ch | ELM_IRQENABLE | ELM Interrupt Enable Register | Section 1.11.5.5 |
| 020h | ELM_LOCATION_CONFIG | ELM Location Configuration Register | Section 1.11.5.6 |
| 080h | ELM_PAGE_CTRL | ELM Page Definition Register | Section 1.11.5.7 |
| 400h + (40h × i) | ELM_SYNDROME_FRAGMENT_0_i | ELM_SYNDROME_FRAGMENT_0_i Register | Section 1.11.5.8 |
| 404h+ (40h × i) | ELM_SYNDROME_FRAGMENT_1_i | ELM_SYNDROME_FRAGMENT_1_i Register | Section 1.11.5.9 |
| 408h + (40h × i) | ELM_SYNDROME_FRAGMENT_2_i | ELM_SYNDROME_FRAGMENT_2_i Register | Section 1.11.5.10 |
| 40Ch + (40h × i) | ELM_SYNDROME_FRAGMENT_3_i | ELM_SYNDROME_FRAGMENT_3_i Register | Section 1.11.5.11 |
| 410h + (40h × i) | ELM_SYNDROME_FRAGMENT_4_i | ELM_SYNDROME_FRAGMENT_4_i Register | Section 1.11.5.12 |
| 414h + (40h × i) | ELM_SYNDROME_FRAGMENT_5_i | ELM_SYNDROME_FRAGMENT_5_i Register | Section 1.11.5.13 |
| 418h + (40h × i) | ELM_SYNDROME_FRAGMENT_6_1 | ELM_SYNDROME_FRAGMENT_6_1 Register | Section 1.11.5.14 |
| 800h + (100h × i) | ELM_LOCATION_STATUS_i | ELM_LOCATION_STATUS_i Register | Section 1.11.5.15 |
| 880h-8BCh + (100h × i) | ELM_ERROR_LOCATION_0-15_i | ELM_ERROR_LOCATION_0-15_i Registers | Section 1.11.5.16 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5.1 ELM Revision Register (ELM_REVISION)

This register contains the IP revision code. The ELM Revision Register (ELM_REVISION) is shown in Figure 1-89 and described in Table 1-153.

**Figure 1-89. ELM Revision Register (ELM_REVISION)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

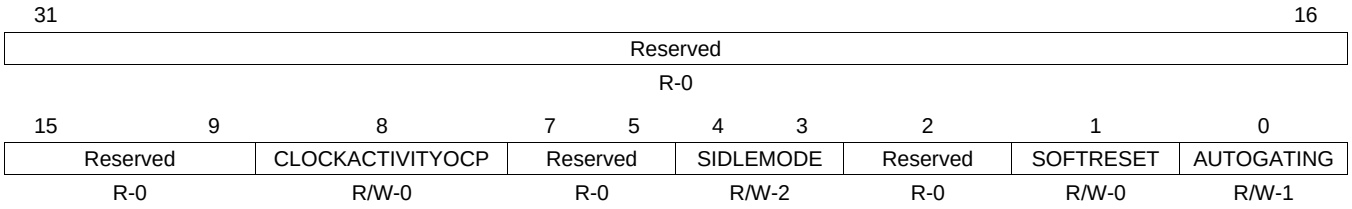
**Table 1-153. ELM Revision Register (ELM_REVISION) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | REVISION | 0-FFFF FFFFh | IP Revision |

### 1.11.5.2 ELM System Configuration Register (ELM_SYSCONFIG)

This register allows controlling various parameters of the OCP interface. The ELM System Configuration Register (ELM_SYSCONFIG) is shown in Figure 1-90 and described in Table 1-154.

**Figure 1-90. ELM System Configuration Register (ELM_SYSCONFIG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-154. ELM System Configuration Register (ELM_SYSCONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved |
| 8 | CLOCKACTIVITY OCP | | OCP Clock activity when module is in IDLE mode (during wake up mode period). |
| | | 0 | OCP Clock can be switch-off |
| | | 1 | OCP Clock is maintained during wake up period |
| 7-5 | Reserved | 0 | Reserved |
| 4-3 | SIDLEMODE | | Slave interface power management (IDLE req/ack control). |
| | | 0 | FORCE Idle. IDLE request is acknowledged unconditionally and immediately. (Default *Dumb* mode for safety) |
| | | 1h | NO idle. IDLE request is never acknowledged. |
| | | 2h | SMART Idle. The acknowledgment to an IDLE request is given based on the internal activity. |
| | | 3h | Reserved - do not use |
| 2 | Reserved | 0 | Reserved |
| 1 | SOFTRESET | | Module software reset. This bit is automatically reset by hardware (During reads, it always returns 0.). It has same effect as the OCP hardware reset. |
| | | 0 | Normal mode. |
| | | 1 | Start soft reset sequence. |
| 0 | AUTOGATING | | Internal OCP clock gating strategy. (No module visible impact other than saving power.) |
| | | 0 | OCP clock is free-running. |
| | | 1 | Automatic internal OCP clock gating strategy is applied based on the OCP interface activity. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5.3 ELM System Status Register (ELM_SYSSTATUS)

The ELM System Status Register (ELM_SYSSTATUS) is shown in Figure 1-91 and described in Table 1-155.

**Figure 1-91. ELM System Status Register (ELM_SYSSTATUS)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | RESETDONE |
| R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-155. ELM System Status Register (ELM_SYSSTATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | | | Internal reset monitoring (OCP domain). Undefined since:<br>From hardware perspective, the reset state is 0.<br>From software user perspective, when the accessible module is 1. |
| | | 0 | Reset is on-going |
| | | 1 | Reset is done (completed) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5.4 ELM Interrupt Status Register (ELM_IRQSTATUS)

This register doubles as a status register for the error-location processes. The ELM Interrupt Status Register (ELM_IRQSTATUS) is shown in Figure 1-92 and described in Table 1-156.

**Figure 1-92. ELM Interrupt Status Register (ELM_IRQSTATUS)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | PAGE_VALID |
| R-0 | | | | | | | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| LOC_VALID_7 | LOC_VALID_6 | LOC_VALID_5 | LOC_VALID_4 | LOC_VALID_3 | LOC_VALID_2 | LOC_VALID_1 | LOC_VALID_0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-156. ELM Interrupt Status Register (ELM_IRQSTATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved |
| 8 | PAGE_VALID | | Error-location status for a full page, based on the mask definition. |
| | | 0 | Read: Error locations invalid for all polynomials enabled in the ECC_INTERRUPT_MASK register. |
| | | 1 | Read: All error locations valid. |
| | | 0 | Write: No effect. |
| | | 1 | Write: Clear interrupt. |
| 7 | LOC_VALID_7 | | Error-location status for syndrome polynomial 7. |
| | | 0 | Read: No syndrome processed or process in progress. |
| | | 1 | Read: Error-location process completed. |
| | | 0 | Write: No effect. |
| | | 1 | Write: Clear interrupt. |
| 6 | LOC_VALID_6 | | Error-location status for syndrome polynomial 6. |
| | | 0 | Read: No syndrome processed or process in progress. |
| | | 1 | Read: Error-location process completed. |
| | | 0 | Write: No effect. |
| | | 1 | Write: Clear interrupt. |
| 5 | LOC_VALID_5 | | Error-location status for syndrome polynomial 5. |
| | | 0 | Read: No syndrome processed or process in progress. |
| | | 1 | Read: Error-location process completed. |
| | | 0 | Write: No effect. |
| | | 1 | Write: Clear interrupt. |
| 4 | LOC_VALID_4 | | Error-location status for syndrome polynomial 4. |
| | | 0 | Read: No syndrome processed or process in progress. |
| | | 1 | Read: Error-location process completed. |
| | | 0 | Write: No effect. |
| | | 1 | Write: Clear interrupt. |
| 3 | LOC_VALID_3 | | Error-location status for syndrome polynomial 3. |
| | | 0 | Read: No syndrome processed or process in progress. |
| | | 1 | Read: Error-location process completed. |
| | | 0 | Write: No effect. |
| | | 1 | Write: Clear interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-156. ELM Interrupt Status Register (ELM_IRQSTATUS) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 2 | LOC_VALID_2 | | Error-location status for syndrome polynomial 2. |
| | | 0 | Read: No syndrome processed or process in progress. |
| | | 1 | Read: Error-location process completed. |
| | | 0 | Write: No effect. |
| | | 1 | Write: Clear interrupt. |
| 1 | LOC_VALID_1 | | Error-location status for syndrome polynomial 1. |
| | | 0 | Read: No syndrome processed or process in progress. |
| | | 1 | Read: Error-location process completed. |
| | | 0 | Write: No effect. |
| | | 1 | Write: Clear interrupt. |
| 0 | LOC_VALID_0 | | Error-location status for syndrome polynomial 0. |
| | | 0 | Read: No syndrome processed or process in progress. |
| | | 1 | Read: Error-location process completed. |
| | | 0 | Write: No effect. |
| | | 1 | Write: Clear interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5.5 ELM Interrupt Enable Register (ELM_IRQENABLE)

The ELM Interrupt Enable Register (ELM_IRQENABLE) is shown in Figure 1-93 and described in Table 1-157.

**Figure 1-93. ELM Interrupt Enable Register (ELM_IRQENABLE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | PAGE_MASK |
| R-0 | | | | | | | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| LOCATION_MASK_7 | LOCATION_MASK_6 | LOCATION_MASK_5 | LOCATION_MASK_4 | LOCATION_MASK_3 | LOCATION_MASK_2 | LOCATION_MASK_1 | LOCATION_MASK_0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-157. ELM Interrupt Enable Register (ELM_IRQENABLE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved |
| 8 | PAGE_MASK | | Page interrupt mask bit |
| | | 0 | Disable interrupt. |
| | | 1 | Enable interrupt. |
| 7 | LOCATION_MASK_7 | | Error-location interrupt mask bit for syndrome polynomial 7. |
| | | 0 | Disable interrupt. |
| | | 1 | Enable interrupt. |
| 6 | LOCATION_MASK_6 | | Error-location interrupt mask bit for syndrome polynomial 6. |
| | | 0 | Disable interrupt. |
| | | 1 | Enable interrupt. |
| 5 | LOCATION_MASK_5 | | Error-location interrupt mask bit for syndrome polynomial 5. |
| | | 0 | Disable interrupt. |
| | | 1 | Enable interrupt. |
| 4 | LOCATION_MASK_4 | | Error-location interrupt mask bit for syndrome polynomial 4. |
| | | 0 | Disable interrupt. |
| | | 1 | Enable interrupt. |
| 3 | LOCATION_MASK_3 | | Error-location interrupt mask bit for syndrome polynomial 3. |
| | | 0 | Disable interrupt. |
| | | 1 | Enable interrupt. |
| 2 | LOCATION_MASK_2 | | Error-location interrupt mask bit for syndrome polynomial 2. |
| | | 0 | Disable interrupt. |
| | | 1 | Enable interrupt. |
| 1 | LOCATION_MASK_1 | | Error-location interrupt mask bit for syndrome polynomial 1. |
| | | 0 | Disable interrupt. |
| | | 1 | Enable interrupt. |
| 0 | LOCATION_MASK_0 | | Error-location interrupt mask bit for syndrome polynomial 0. |
| | | 0 | Disable interrupt. |
| | | 1 | Enable interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5.6 ELM Location Configuration Register (ELM_LOCATION_CONFIG)

The ELM Location Configuration Register (ELM_LOCATION_CONFIG) is shown in Figure 1-94 and described in Table 1-158.

**Figure 1-94. ELM Location Configuration Register (ELM_LOCATION_CONFIG)**

| 31 27 | 16 16 | 15 2 | 1 0 |
|---|---|---|---|
| Reserved | ECC_SIZE | Reserved | ECC_BCH_LEVEL |
| R-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-158. ELM Location Configuration Register (ELM_LOCATION_CONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-27 | Reserved | 0 | Reserved |
| 26-16 | ECC_SIZE | 0-7FFh | Maximum size of the buffers for which the error-location engine is used, in number of nibbles (4-bits entities) |
| 15-2 | Reserved | 0 | Reserved |
| 1-0 | ECC_BCH_LEVEL | | Error correction level. |
| | | 0 | 4 bits. |
| | | 1h | 8 bits. |
| | | 2h | 16 bits. |
| | | 3h | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5.7 ELM Page Definition Register (ELM_PAGE_CTRL)

The ELM Page Definition Register (ELM_PAGE_CTRL) is shown in Figure 1-95 and described in Table 1-159.

**Figure 1-95. ELM Page Definition Register (ELM_PAGE_CTRL)**

| 31 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SECTOR_7 | SECTOR_6 | SECTOR_5 | SECTOR_4 | SECTOR_3 | SECTOR_2 | SECTOR_1 | SECTOR_0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-159. ELM Page Definition Register (ELM_PAGE_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7 | SECTOR_7 | 0-1 | Set to 1 if syndrome polynomial 7 is part of the page in page mode. Must be 0 in continuous mode. |
| 6 | SECTOR_6 | 0-1 | Set to 1 if syndrome polynomial 6 is part of the page in page mode. Must be 0 in continuous mode. |
| 5 | SECTOR_5 | 0-1 | Set to 1 if syndrome polynomial 5 is part of the page in page mode. Must be 0 in continuous mode. |
| 4 | SECTOR_4 | 0-1 | Set to 1 if syndrome polynomial 4 is part of the page in page mode. Must be 0 in continuous mode. |
| 3 | SECTOR_3 | 0-1 | Set to 1 if syndrome polynomial 3 is part of the page in page mode. Must be 0 in continuous mode. |
| 2 | SECTOR_2 | 0-1 | Set to 1 if syndrome polynomial 2 is part of the page in page mode. Must be 0 in continuous mode. |
| 1 | SECTOR_1 | 0-1 | Set to 1 if syndrome polynomial 1 is part of the page in page mode. Must be 0 in continuous mode. |
| 0 | SECTOR_0 | 0-1 | Set to 1 if syndrome polynomial 0 is part of the page in page mode. Must be 0 in continuous mode. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5.8 ELM_SYNDROME_FRAGMENT_0_i Register

The ELM_SYNDROME_FRAGMENT_0_i Register is shown in Figure 1-96 and described in Table 1-160.

**Figure 1-96. ELM_SYNDROME_FRAGMENT_0_i Register**

| 31 | 0 |
|---|---|
| SYNDROME_0 | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-160. ELM_SYNDROME_FRAGMENT_0_i Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SYNDROME_0 | 0-FFFF FFFFh | Syndrome bits 0 to 31. |

### 1.11.5.9 ELM_SYNDROME_FRAGMENT_1_i Register

The ELM_SYNDROME_FRAGMENT_1_i Register is shown in Figure 1-97 and described in Table 1-161.

**Figure 1-97. ELM_SYNDROME_FRAGMENT_1_i Register**

| 31 | 0 |
|---|---|
| SYNDROME_1 | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-161. ELM_SYNDROME_FRAGMENT_1_i Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SYNDROME_1 | 0-FFFF FFFFh | Syndrome bits 32 to 63. |

### 1.11.5.10 ELM_SYNDROME_FRAGMENT_2_i Register

The ELM_SYNDROME_FRAGMENT_2_i Register is shown in Figure 1-98 and described in Table 1-162.

**Figure 1-98. ELM_SYNDROME_FRAGMENT_2_i Register**

| 31 | 0 |
|---|---|
| SYNDROME_2 | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-162. ELM_SYNDROME_FRAGMENT_2_i Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SYNDROME_2 | 0-FFFF FFFFh | Syndrome bits 64 to 95. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5.11 ELM_SYNDROME_FRAGMENT_3_i Register

The ELM_SYNDROME_FRAGMENT_3_i Register is shown in Figure 1-99 and described in Table 1-163.

**Figure 1-99. ELM_SYNDROME_FRAGMENT_3_i Register**

| 31 | 0 |
|---|---|
| SYNDROME_3 | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-163. ELM_SYNDROME_FRAGMENT_3_i Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SYNDROME_3 | 0-FFFF FFFFh | Syndrome bits 96 to 127. |

### 1.11.5.12 ELM_SYNDROME_FRAGMENT_4_i Register

The ELM_SYNDROME_FRAGMENT_4_i Register is shown in Figure 1-100 and described in Table 1-164.

**Figure 1-100. ELM_SYNDROME_FRAGMENT_4_i Register**

| 31 | 0 |
|---|---|
| SYNDROME_4 | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-164. ELM_SYNDROME_FRAGMENT_4_i Register Field Descriptions**

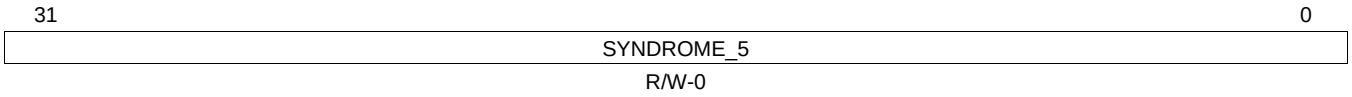
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SYNDROME_4 | 0-FFFF FFFFh | Syndrome bits 128 to 159. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5.13 ELM_SYNDROME_FRAGMENT_5_i Register

The ELM_SYNDROME_FRAGMENT_5_i Register is shown in Figure 1-101 and described in Table 1-165.

**Figure 1-101. ELM_SYNDROME_FRAGMENT_5_i Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

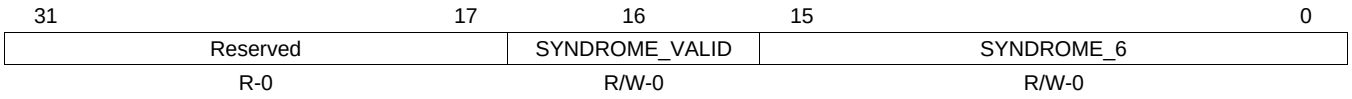
**Table 1-165. ELM_SYNDROME_FRAGMENT_5_i Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SYNDROME_5 | 0-FFFF FFFFh | Syndrome bits 160 to 191. |

### 1.11.5.14 ELM_SYNDROME_FRAGMENT_6_i Register

The ELM_SYNDROME_FRAGMENT_6_i Register is shown in Figure 1-102 and described in Table 1-166.

**Figure 1-102. ELM_SYNDROME_FRAGMENT_6_i Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 1-166. ELM_SYNDROME_FRAGMENT_6_i Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved |
| 16 | SYNDROME_VALID | | Syndrome valid bit. |
| | | 0 | This syndrome polynomial should not be processed. |
| | | 1 | This syndrome polynomial must be processed. |
| 15-0 | SYNDROME_6 | 0-FFFFh | Syndrome bits 192 to 207. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.11.5.15 ELM_LOCATION_STATUS_i Register

The ELM_LOCATION_STATUS_i Register is shown in Figure 1-103 and described in Table 1-167.

**Figure 1-103. ELM_LOCATION_STATUS_i Register**

| 31 | 9 | 8 | 7 | 5 | 4 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | ECC_CORRECTABL | Reserved | | ECC_NB_ERRORS | |
| R-0 | | R-0 | R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-167. ELM_LOCATION_STATUS_i Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved |
| 8 | ECC_CORRECTABL | | Error-location process exit status. |
| | | 0 | ECC error-location process failed. Number of errors and error locations are invalid. |
| | | 1 | All errors were successfully located. Number of errors and error locations are valid. |
| 7-5 | Reserved | 0 | Reserved |
| 4-0 | ECC_NB_ERRORS | 0-1Fh | Number of errors detected and located. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.11.5.16 ELM_ERROR_LOCATION_0-15_i Registers

The ELM_ERROR_LOCATION_0-15_i Registers is shown in Figure 1-104 and described in Table 1-168.

**Figure 1-104. ELM_ERROR_LOCATION_0-15_i Registers**

| 31 | 13 | 12 | 0 |
|---|---|---|---|
| Reserved | | ECC_ERROR_LOCATION | |
| R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 1-168. ELM_ERROR_LOCATION_0-15_i Registers Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-13 | Reserved | 0 | Reserved |
| 12-0 | ECC_ERROR_LOCATION | 0-1FFFh | Error-location bit address. |

## 1.12 Bus Interconnect

### *1.12.1 Interconnect Overview*

**NOTE:** The L3 interconnect is instantiation of the NoC interconnect from Arteris, Inc. Arteris® is a registered trademark of Arteris, Inc.

This document contains materials that are ©Arteris, Inc., and that constitute proprietary information of Arteris, Inc.

All materials and trademarks are used under license from Arteris, Inc. For additional information, see the Arteris Reference manuals, or contact Arteris, Inc.

NoC is an abbreviation for Network On Chip.

**NOTE:** The L4 interconnects are instantiations of the Sonics3220™ interconnect from Sonics, Inc.

This document contains materials that are ©2003-2009 Sonics, Inc., and that constitute proprietary information of Sonics, Inc.

SonicsMX™ and Sonics3220, are trademarks or registered trademarks of Sonics, Inc. All such materials and trademarks are used under license from Sonics, Inc. For additional information, see the SonicsMX or Sonics3220 Reference manuals, or contact Sonics, Inc.

SMX is an abbreviation for SonicsMX.

#### 1.12.1.1 Terminology

The following terminology is critical to understanding the interconnect:

- Initiator: Module able to initiate read and write requests to the chip interconnect (typically: processors, DMA, etc.).
- Target: Unlike an initiator, a target module cannot generate read/write requests to the chip interconnect, but it can respond to these requests. However, it may generate interrupts or a DMA request to the system (typically: peripherals, memory controllers).

**NOTE:** A module can have several separate ports; therefore, a module can be an initiator and a target.

- Agent: Each connection of one module to one interconnect is done using an agent, which is an adaptation (sometimes configurable) between the module and the interconnect. A target module is connected by a target agent (TA), and an initiator module is connected by an initiator agent (IA).


Copyright © 2011–2016, Texas Instruments Incorporated

- Interconnect: The decoding, routing, and arbitration logic that enables the connection between multiple initiator modules and multiple target modules connected on it.
- Register target (RT): Special TA used to access the interconnect internal configuration registers
- Data-flow signal: Any signal that is part of a clearly identified transfer or data flow (typically: command, address, byte enables, etc.). Signal behavior is defined by the protocol semantics.
- Sideband signal: Any signal whose behavior is not associated to a precise transaction or data flow.
- Out-of-band error: Any signal whose behavior is associated to a device error-reporting scheme, as opposed to in-band errors.

**NOTE:** Interrupt requests and DMA requests are not routed by the interconnect in the device.

- Firewall: A programmable feature integrated in a target agent or L4 interconnect to prevent unauthorized access to or from a module. A firewall can be configured using three criteria:
  - Initiator requesting access
  - Address space access
  - Type of access
- ConnID: Any transaction in the system interconnect is tagged by an in-band qualifier ConnID, which uniquely identifies the initiator at a given interconnect point. A ConnID is transmitted in band with the request and is used for firewall and error-logging mechanism.
- Firewall comparison mechanism: A comparison made in the firewall between access in-band qualifiers and access permissions that are programmed in the firewall configuration registers. If the comparison is successful, access is allowed; otherwise, access is denied.
- MCmd qualifier: Command bus that indicates the type of transfer requested. Table 1-169 lists the commands encoded.

**Table 1-169. MCmd Qualifier Description**

| MCmd[2:0] | Transaction Type |
|---|---|
| 0 0 0 | Idle |
| 0 0 1 | Write |
| 0 1 0 | Read |
| 0 1 1 | ReadEx |
| 1 0 0 | Read link |
| 1 0 1 | Write nonposted |
| 1 1 0 | Write conditional |
| 1 1 1 | Write broadcast |

- MReqInfo qualifier: Four MReqInfo qualifiers describe the access during the use of the firewall comparison mechanism, as described in Table 1-170.

**Table 1-170. MReqInfo Qualifier Description**

| Qualifiers | Description |
|---|---|
| MReqType | 0: Data access |
| | 1: Opcode fetch |
| MReqSupervisor | 0: User mode |
| | 1: Supervisor mode |
| MReqDebug | 0: Functional access |
| | 1: Debug access |


Copyright © 2011–2016, Texas Instruments Incorporated

- Register that configures the combination of the MReqInfo, allowing access permission to the TM based on the MReqInfo in-band qualifier values.
- SError: Target that indicates an error condition to the initiator.
- SResp qualifier: Response from the target to the initiator concerning the transaction, as described in Table 1-171.

**Table 1-171. SResp Qualifier Description**

| SResp[1:0] | Description |
|---|---|
| 0 0 | No response |
| 0 1 | Data valid/accept |
| 1 0 | Not used |
| 1 1 | Error |

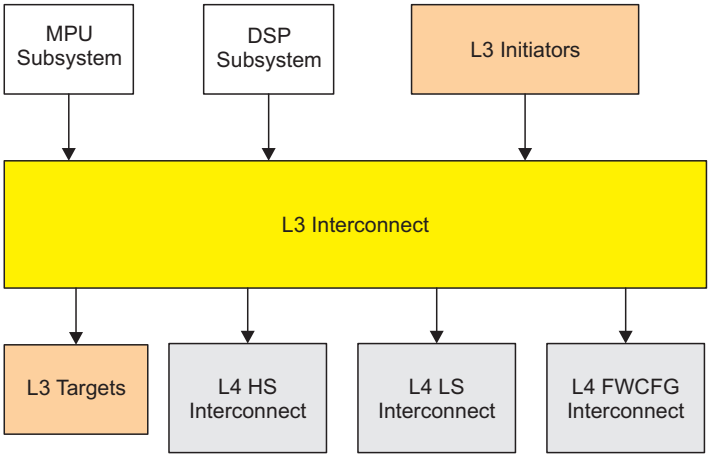
### 1.12.1.2 Architecture Overview

The device memory hierarchy includes four levels:

- L1 is internal to the CPUs. It concerns data exchange with the internal Level1 cache memory subsystem, and it is the closest memory to the microprocessor unit (MPU) core.
- L2 is included in the DSP subsystem and the MPU subsystem.
- The chip-level interconnect consists of one L3 interconnect and three L4 interconnects. It enables communication among the modules and subsystems in the device.

Figure 1-105 shows an overview of the L3 and L4 interconnect architecture.

**Figure 1-105. Interconnect Overview**





Copyright © 2011–2016, Texas Instruments Incorporated

- L3 handles many types of data transfers, especially exchanges with system-on-chip/external memories. L3 transfers data with a maximum width of 64 bits from the initiator to the target. The L3 interconnect is a little-endian platform
- The L4 is composed of the following:
  - L4 HS: Includes the majority of the configuration interface for level 3 (L3) system modules and peripheral interconnect
  - L4 LS : Includes the main peripherals that require enhanced direct memory access (EDMA) access
  - L4 FWCFG: Includes peripherals attached to the WKUP power domain.

Modules are connected to the interconnect through an IA for the initiator module and a TA for target modules. Each module/subsystem connection is statically configured to tune the access, depending on the characteristics of the module.

To unauthorize a module or L4 interconnect access, some TAs include configurable firewalls (FWs). A firewall restricts or filters the accesses allowed to an initiator according to different access criteria. The firewalls can usually be configured by software.

The L3 and L4 interconnect default settings are fully functional; they enable all possible functional data paths and a minimal default protection setting. However, the interconnect parameters can be modified to fit user expectations.


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.2 L3 Interconnect

This section describes the L3 interconnect and its components. With the exception of register points, each component includes functions for the request and response networks.

#### 1.12.2.1 L3 Interconnect Overview

The L3 interconnect links cores in a flexible topology that couples low power with high performance. Innovative physical structures and advanced protocols ensure bandwidth and latency to individual IP cores, providing dedicated connections between IP cores and logical connections over a shared interconnect.

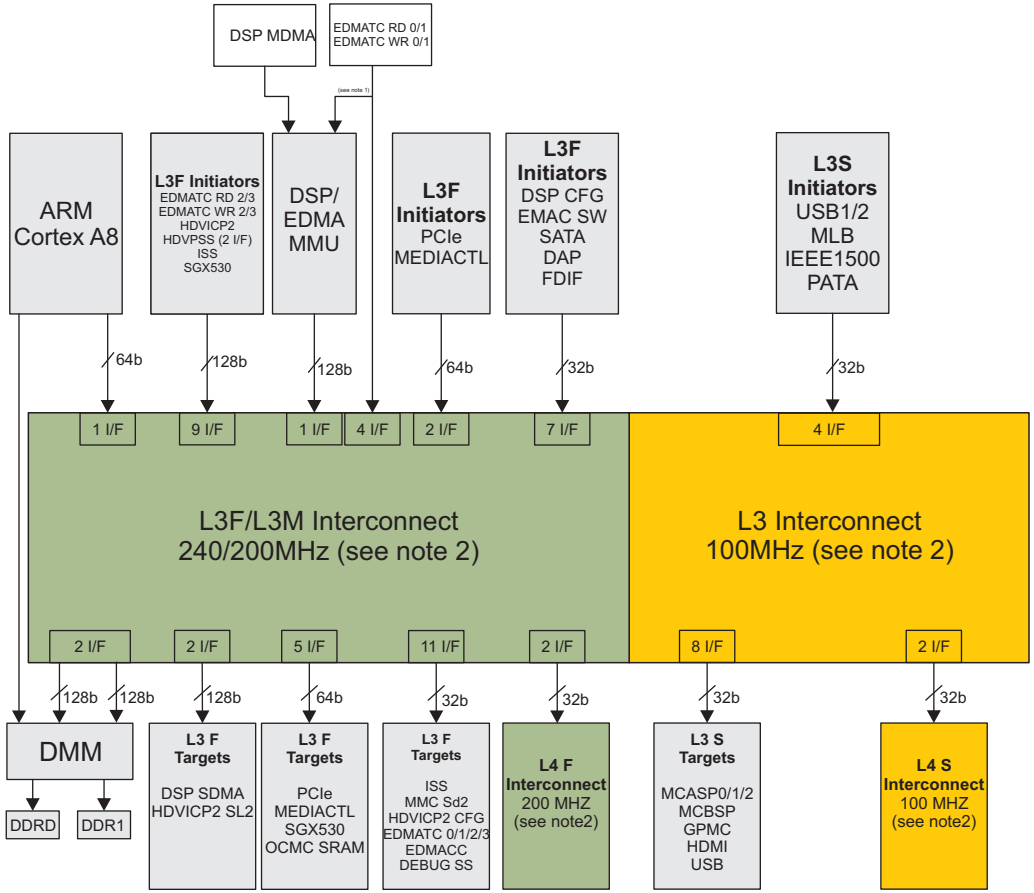
The main features of the L3 interconnects are:

- Network interface units (NIUs): Master NIUs for the initiator agents (IAs) and slave NIUs for the target agents (TAs)
- True little-endian platform
- Transaction errors tracking and logging registers
- All signaling support for chip-level power-management infrastructure
- One interrupt line signaling transaction error
- One interrupt line for reporting satistical events on the L3 interconnect

Figure 1-106 is an overview of the L3 interconnect and the peripherals attached to it.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-106. L3 Interconnect Overview**






Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.2.2 L3 Interconnect Integration

Table 1-172 through Table 1-173 summarize the integration of the module in the device.

**Table 1-172. Integration Attributes**

| Module Instance | Attributes |
|---|---|
| | Power Domain |
| L3 | Always On power domain |

**Table 1-173. Clocks and Resets**

| Clocks | | | |
|---|---|---|---|
| **Module Instance** | **Destination Signal Name** | **Source** | **Description** |
| L3 | L3_220M_ICLK | PRCM | Root clock for L3F clock domain |
| L3 | L3_110M_ICLK | PRCM | Root clock for L3S clock domain |
| L3 | L3_200M_ICLK | PRCM | Root clock for L3M debug clock domain |
| **Resets** | | | |
| **Module Instance** | **Destination Signal Name** | **Source** | **Description** |
| L3 | L3_220M_RST | PRCM | Reset to L3F clock domain |
| L3 | L3_110M_RST | PRCM | Reset to L3Sclock domain |
| L3 | L3_200M_RST | PRCM | Reset to L3M debug clock domain |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.2.3 L3 Interconnect Functional Description

#### *1.12.2.3.1 Module Use in L3 Interconnect*

The L3 interconnect network components have ConnID values for each master NIU or slave NIU. The ID uniquely identifies the master NIU or the slave NIU for an interconnect transfer. The interconnect uses ConnIDs for a number of purposes, including the following:

- Slave NIUs for error logging
- Power disconnect slave NIU for error logging
- FLAGMUX to mask interrupts: Three, one for each clock domain
- STATCOLL for configuring and monitoring: The STATCOLL components compute the traffic statistics within a user-defined window and periodically report to the user through the debug interface.
- Bandwidth regulator for configuration.
- Some rate adapters for control of throughput ratio

#### *1.12.2.3.2 Module Distribution*

Master NIUs and slave NIUs provide the interface to connect the different modules to their associated interconnect.

Table 1-174 and Table 1-175 list all the modules and subsystems with their associated agents. The agents are listed for each L3 interconnect domain.

##### *1.12.2.3.2.1 L3 Interconnect Agents*

Any initiator or target core is connected to the L3 interconnect through an NIU. NIUs act as entry points to the L3 interconnect, and also include various programming registers. Table 1-174 lists the supported master NIU ports.

**Table 1-174. Master NIUs**

| Master NIU | Description |
|---|---|
| L3_MN_MPU | Cortex A8 microprocessor subsystem unit 128-bit port |
| L3_MN_HDVICP | HDVICP subsystem port 1 128-bit initiator port |
| L3_MN_ISS | Imaging subsystem 128-bit initiator port |
| L3_MN_MMU | Memory management unit 128-bit initiator port |
| L3_MN_DSP1 | DSP subsystem 128-bit initiator port1 (GEM MDMA) |
| L3_MN_EXPSLOT0 | Expansion slot 0 128-bit initiator port |
| L3_MN_TPTC_RD1/2/3/4 | TPTC 128-bit read initiator ports |
| L3_MN_TPTC_WR1/2/3/4 | TPTC 128-bit write initiator ports |
| L3_MN_HDVPSS1 | Display subsystem port 1 - 128-bit initiator |
| L3_MN_HDVPSS2 | Display subsystem port 2 - 128-bit initiator |
| L3_MN_DSP2 | DSP subsystem 128-bit initiator port 2 (GEM CFG) |
| L3_MN_SGX | SGX 530 2D graphics accelerator 128-bit initiator port |
| L3_MN_MEDIACONTROLLER | Media Controller 64-bit initiator port |
| L3_MN_PCIE | PCIe 64-bit initiator port |
| L3_MN_3PGSW | 3PGSW 32-bit initiator port |
| L3_MN_FDIF | Face Detection Interface 32-bit initiator port |
| L3_MN_SATA | SATA controller 32-bit initiator port |
| L3_MN_DAP | Debug access point 32-bit initiator port |
| L3_MN_USB1/2 | USB CPP DMA/QUEUE Manager 32-bit initiator port |
| L3_MN_PATA | PATA 32-bit initiator port |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-174. Master NIUs (continued)**

| Master NIU | Description |
|---|---|
| L3_MN_IEEE1500 | IEEE1500 32-bit initiator port |
| L3_MN_MLB | MLB 32-bit initiator port |

Table 1-175 lists the supported slave NIU ports.

**Table 1-175. Slave NIUs**

| Slave NIU | Description |
|---|---|
| L3_SN_DMM0 | Direct memory manager 0 (DMM0) 128-bit target port |
| L3_SN_DMM1 | DMM1 128-bit target port |
| L3_SN_SL2 | HDVICP SL2IF 128-bit target port |
| L3_SN_EXPSLOT0 | Expansion slot 0 128-bit target port |
| L3_SN_DSP | DSP (GEM SDMA) 128-bit target port |
| L3_SN_TPTC_CFG | 4 32-bit TPTC configuration target ports |
| L3_SN_TPCC_CFG | 32-bit TPCC configuration target port |
| L3_SN_SGX | 2D Graphics accelerator (SGX530) 64-bit target port |
| L3_SN_OCMRAM0 | OCM RAM 64-bit target port |
| L3_SN_MEDIACONTROLLER | Media Controller 64-bit target port |
| L3_SN_PCIE | PCIe 64-bit target port |
| L3_SN_VCP | VCP2 64-bit target port |
| L3_SN_HDVICP | HDVICP Host 1 Control 32-bit target port |
| L3_SN_ISS | ISS 32-bit target port |
| L3_SN_MMCHS2 | MMCHS 32-bit target port |
| L3_SN_L3_INSTR | L3 Instrumentation 32-bit target port |
| L3_SN_L4F | 2 L4_FAST peripheral 32-bit target ports |
| L3_SN_DAP | Debug subsystem target port |
| L3_SN_L4S | 2 L3_SLOW peripheral 32-bit target port |
| L3_SN_GPMC | General Purpose Memory Controller 32-bit target port |
| L3_SN_MCASP0/1/2 | McASP 32-bit target ports |
| L3_SN_HDMI | HDMI 32-bit target port |
| L3_SN_USB | USB 32-bit target port |
| L3_SN_L4 Firewall | L3 Firewalls target port |
| L3_SN_MCBSP | McBSP 32-bit target port |

#### *1.12.2.3.2.2 L3 Connectivity Matrix*

The L3 interconnect is divided into two main clock domains plus one small domain dedicated to DRAM performance monitorring:

- L3F clock domain (220 MHz)
- L3S clock domain (110 MHz)
- L3M cock domain (200MHz clock domain for performance monitoring)

The two main clock domains (L3F and L3S) are sychronous and balanced to each other. The third clock domain L3M, dedicated to performance monitoring is asynchronous to all other domains.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.12.2.3.2.2.1 Clock Domain Mapping of the L3 Interconnect Modules

Each clock domain (L3F, L3S, and L3M) has its own host, flag mux, slave NIUs, and bandwidth regulators.

**Table 1-176. L3 Clock Domains and Elements**

| Domain | Elements |
|---|---|
| L3F | HOST_CLK1 |
| | MMU OCP |
| | MMU BW Regulator |
| | TPTC1 OCP |
| | TPTC4 RD BW Regulator |
| | TPTC4 WR BW Regulator |
| | TPTC2 RD BW Regulator |
| | TPTC2 WR BW Regulator |
| | TPTC3 OCP |
| | TPTC4 OCP |
| | TPCC OCP |
| | SL2_1 OCP |
| | GEMDMA OCP |
| | EXPANSION OCP |
| | STATCOLL2 |
| | FLAGMUX_CLK1 |
| | FLAGMUX_TOP |
| | STATCOLL 0 |
| | STATCOLL 1 |
| L3M | HOST_CLK2 |
| | L4HS_0 OCP |
| | L4HS_1 OCP |
| | MMCHS2 OCP |
| | EMUSS OCP |
| | STATCOLL3 |
| | FLGMUX_CLK2 |
| | RAM0 OCP |
| | Media Controller OCP |
| | ISS OCP |
| | ISS BW Regulator |
| | DMM1 OCP |
| | DMM2 OCP |
| | HDVICP OCP |
| | HDVICP BW Regulator |
| | SGX OCP |
| | SGX BW Regulator |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-176. L3 Clock Domains and Elements (continued)**

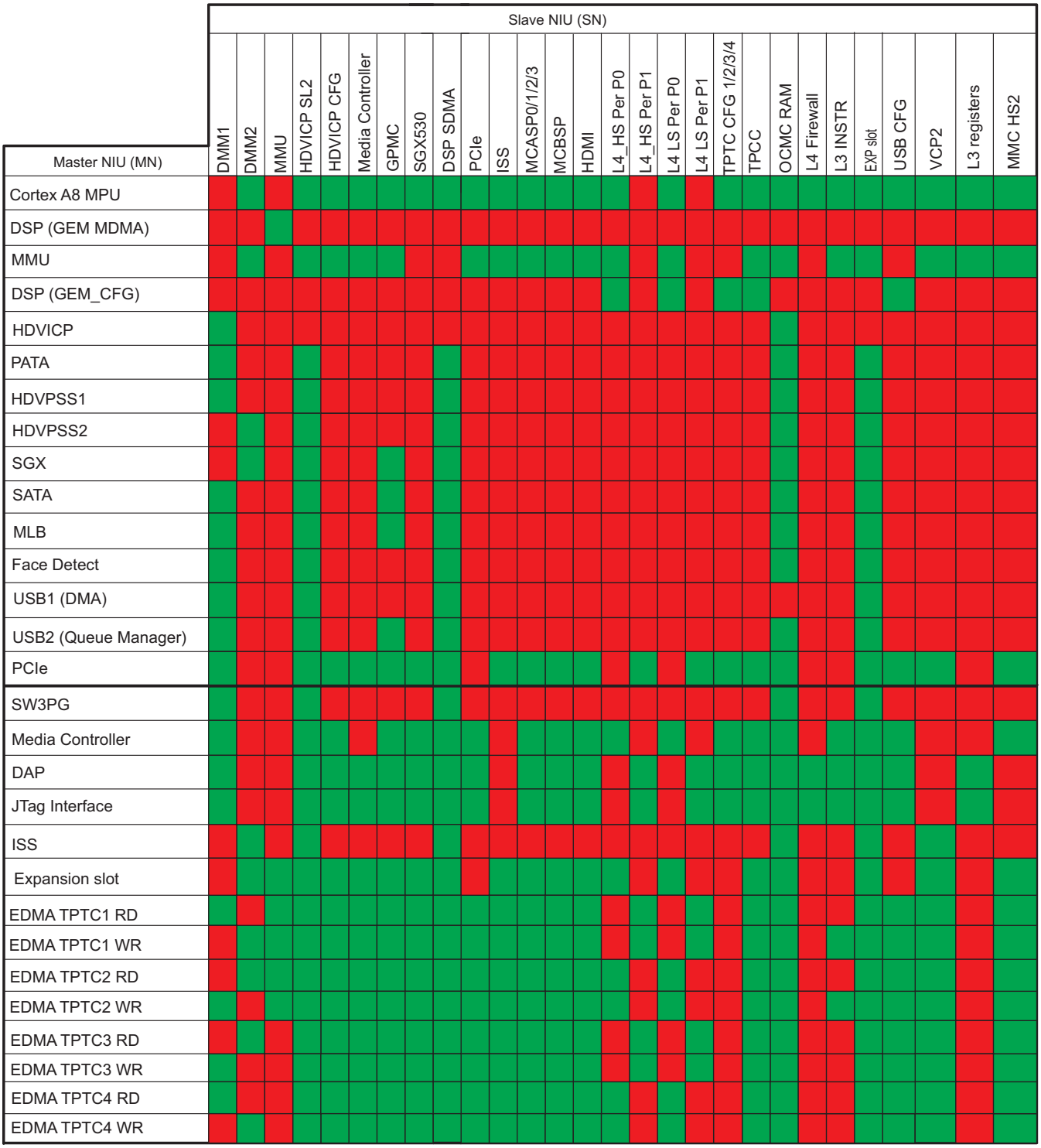
| Domain | Elements |
|---|---|
| L3S | HOST_CLK3 |
| | MCBSP OCP |
| | USB OCP |
| | MCASP1/2/3 OCP |
| | FLAGMUX_CLK3 |
| | STATCOLL4 |
| | L4FW_CFG OCP |
| | L4LS_0/1 OCP |
| | GPMC OCP |


Copyright © 2011–2016, Texas Instruments Incorporated

Figure 1-107 lists the functional paths between the L3 master NIUs and the L3 and L4 slave NIU agents. The functional paths in the figure are indicated by the following:

- A cell is green when a functional path exists.
- A cell is red when a functional path does not exist.

**Figure 1-107. L3 Connectivity Matrix**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.12.2.3.2.3 Master NIU Identification

An master NIU ID (ConnID) is assigned to every module in the device. The ID uniquely identifies the master NIU for an interconnect transfer (see Table 1-177). The interconnect uses ConnID values for a number of purposes, including:

- Master source identification for the protection mechanism
- Response route generation
- Firewall error logging
- L3 interconnect error logging

**Table 1-177. ConnID Values**

| 4 Bit ConnID Value (security) (hex) | 6 Bit ConnID Value (debug) (hex) | Master NIU |
|---|---|---|
| 0x0 | 0 | Cortex-A8 MPU |
| - | 2 | HDVICP_SW_INSTR_IC0 |
| - | 3 | HDVICP_SW_INSTR_IC1 |
| 1 | 4 | DAP |
| 1 | 5 | JTAG Interface (IEEE1500) |
| 2 | 8 | GEM (DSP_MDMA) |
| 2 | 9 | GEM_CFG (DSP_CFG) |
| 2 | A | MMU |
| 3 | E | Media Controller |
| 6 | 18 | TPTC_RD1 |
| 6 | 19 | TPTC_WR1 |
| 6 | 1A | TPTC_RD2 |
| 6 | 1B | TPTC_WR2 |
| 7 | 1C | TPTC_RD3 |
| 7 | 1D | TPTC_WR3 |
| 7 | 1E | TPTC_RD4 |
| 7 | 1F | TPTC_WR4 |
| 8 | 20 | SGX |
| - | 21 | RESERVED (Hardware fixed) |
| - | 22 | RESERVED (Hardware fixed) |
| 9 | 24 | HDVPSS1 |
| 9 | 25 | HDVPSS2 |
| A | 28 | HDVICP |
| - | 29 | RESERVED (Hardware fixed) |
| B | 2D | ISS |
| C | 30 | FDIF |
| C | 31 | SW3PG |
| D | 34 | USB1 |
| D | 35 | USB2 |
| D | 36 | PATA |
| D | 37 | STATCOLL4 |
| E | 38 | MLB |
| E | 39 | SATA |
| E | 3A | PCIe |
| E | 3B | EXPANSION SLOT |
| F | 3C | STATCOLL0 |
| F | 3D | STATCOLL1 |
| F | 3E | STATCOLL2 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-177. ConnID Values (continued)**

| 4 Bit ConnID Value (security) (hex) | 6 Bit ConnID Value (debug) (hex) | Master NIU |
|---|---|---|
| F | 3F | STATCOLL3 |

#### 1.12.2.3.3 Bandwidth Regulators

The bandwidth regulators prevent master NIUs from consuming too much bandwidth of a link, or a slave NIU that is shared between several data flows: packets are then transported at a slower rate. The value of a bandwidth can be programmed in the bandwidth regulator. When the bandwidth is below the programmed value, the pressure bit is set to 1, giving priority to this master. When the bandwidth is above the programmed value, the pressure bit is set to 0 and the concerned master has the same weight as others.

The bandwidth registers regulate the packet flow by applying flow control on the RX port, thus ensuring that the traffic does not exceed the allocated bandwidth. The next packet is sent only when an internal timer expires. The registers in this group are:

- L3_BW_REGULATOR_WATERMARK: Gives the amount of data allowed to exceed the average bandwidth during a short time period
- L3_BW_REGULATOR_PRESS: Describes the pressure applied to outgoing packets
- L3_BW_REGULATOR_CLEARHISTORY: Resets the traffic counter when set to 1. It is used after an update in the L3_BW_REGULATOR_BANDWIDTH and L3_BW_REGULATOR_WATERMARK registers (see Section 1.12.2.5.1.4, *L3 BW Regulator Register Summary and Description*).

Bandwidth regulators are mainly used to give priority to the following masters: HDVICP, TPTC_RD2/4, TPTC_WR2/4, MMU, ISS and SGX.

#### 1.12.2.3.4 Flag Muxing

The flag mux generator collects information such as errors and interrupts from slave NIUs and the interconnect firewall. The result signals are then sent to the MPU interrupt controller without interfering with the interconnect traffic. Flag mux unit contains software registers that are able to eventually mask some interrupts in order to prevent the unit from seeing them and to help with debug.

There are four registers in this group:

- One register to mask application error sources
- One status register to check which application error sources are active
- One register to mask debug error sources
- One status register to check which debug error source are active

By default none of the flags are masked.

There are four flag muxes in Centaurus L3 Interconnect, one per each clock domain plus additional flag mux collecting flags from the three previous ones. In concequnce L3 will generate two flags: APPLICATION error and DEBUG error.

The unit-specific registers (MASK bit 0 or bit 1 of the flag inputs, and L3_FLAGMUX_REGERR bit 0 or bit 1) are used to read the input errors. Each register is dedicated to reporting the bit corresponding to the register number. Flag mux inputs for the different sources can be see in Table 1-230.

#### 1.12.2.3.5 Statistic Collectors Group

Statistic collectors are internal masters that share the same master address as the master NIUs. These components compute the traffic statistics within a defined window and periodically report through the DEBUG interface. The key features of the statistic collector are:

- Nonintrusive monitoring
- Programmable filters and counters
- Collects results at a programmable time interval


Copyright © 2011–2016, Texas Instruments Incorporated

Event detectors are programmed through the L3_STCOL_REQEVT and L3_STCOL_RSPEVT configuration registers for request and response ports, respectively. The following events can be identified:

- Word transfer
- WAIT cycles
- Flow control
- Payload transfers
- Latency measurements

Performance monitoring is enabled through the L3_STCOL_EN register. The L3_STCOL_SOFTEN register enables software to monitor the performance. Event muxes are programmed through the L3_STCOL_EVTMUX_SEL0 configuration register, which determines which port will be monitored by a filter configured by the filter registers (see Table 1-268).

Filters are programmed through the Table 1-268 configuration register, along with additional selection criteria programmed through the mask/match registers (see Table 1-250). A filter can be configured to accept or reject:

- Read operations
- Write operations
- Errors
- Addresses

Filter operation is programmed through the L3_STCOL_OP registers (see Table 1-250).

There are five statistic collectors embedded in Centaurus L3, one dedicated to DRAM load monitoring and four statistic collectors dedicated to master latency monitoring.

#### 1.12.2.3.6 L3 Interconnect Error Handling

Error logging is enabled in the L3 interconnect. The three major types of errors are:

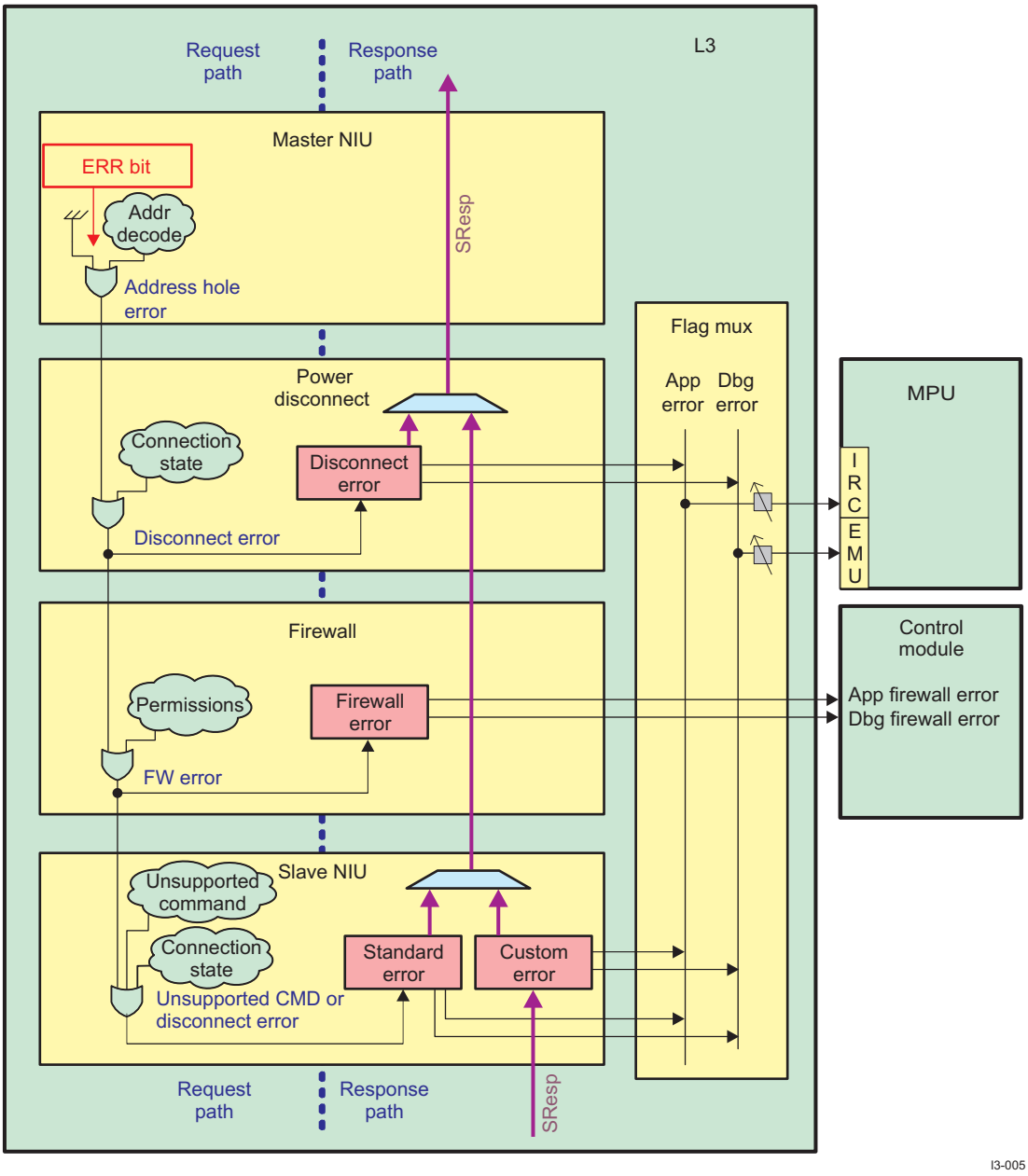
- Slave NIU errors
- Firewall errors (see , *L3Firewall Error Logging*)
- Flag mux errors

##### 1.12.2.3.6.1 Global Error-Routing Scheme

Figure 1-108 shows the L3 global error-routing scheme.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-108. L3 Global Error-Routing Scheme**




### *1.12.2.3.6.2 Slave NIU Error Logging*

Error logging is implemented only at slave NIUs. Because the interconnect does not support master NIU error logging, an erroneous packet must be created and sent to one of the slave NIUs. The slave NIU that receives an erroneous packet is predictable but can change per master (see Table 1-178).


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-178. L3 Connectivity and Holes Error Routing**

| Master | Connectivity and Holes Errors Logged Into Slave NIUs |
|---|---|
| DAP | GPMC |
| Dual Cortex M3 (Media Controller) | |
| Expansion slot | |
| JTag interface (IEEE1500) | |
| MMU | |
| MPU (Modena) | |
| PCIe | |
| SATA | |
| MLB | |
| SGX | |
| TPTC_RD1/2/3/4 | |
| TPTC_WR1/2/3/4 | |
| USB2 | |
| SW3PG | OCMC RAM0 |
| HDVPSS1/2 | |
| HDVICP | |
| FACEDETECT | |
| PATA | |
| DSP (GEM_CFG) | USB Targ |
| DSP (GEM_MDMA) | MMU Targ |
| USB1 | DSP (GEM_SDMA) |

The slave NIU can be configured to report standard errors (errors generated within the interconnect):

- Firewall error: Protection violation; this error indicates that a request was rejected by a firewall and is reported to the control module.
- Address hole: This error reports an unknown address for a request. The address map is local to each master NIU; therefore, an address hole error is reported each time a master NIU requests an access to a slave NIU to which it is not logically connected, even if this address exists in the global L3address map. This error is detected only once per burst.
- Unsupported commands: This error reports that the master NIU sent a command that cannot be processed, because the slave NIU cannot accept it and no conversion to another command is possible. This error is detected only once per burst.
- Report custom errors: Basically, when the slave answer is SResp = ERR
- Report severity level, for standard error and custom errors:
  - None: Error logging for this type of error is disabled.
  - Error: Error is logged for this type of error.
  - Fault: Error is logged and interrupt is generated for this type of error.
- Generate interrupt on 2 bits depending on the MReqDebug qualifier:
  - Application error - Fault[0]
  - Debug error - Fault[1]

  By default, all slave NIUs are configured with standard and custom error levels set to FAULT, active high. The errors are reported on the two flag muxes (see Figure 1-108), depending on the access type, application or debug. For more information, see Section 1.12.2.3.6.3, *Flag Mux Error Logging*.

  The slave NIU power-disconnect component also has error logging enabled. By nature, this component can generate only standard errors. By default, it is configured with the error level set to FAULT.

  Wake up on demand: If an error packet reaches a slave NIU that is set with MDiscBehave = 1 (wake up on demand), then the active signal is asserted and L3 processes the error generation when the slave is awake. This is inefficient but it simplifies NIU implementation and should not be a problem,


Copyright © 2011–2016, Texas Instruments Incorporated

because errors are supposed to occur only during software debug.

#### 1.12.2.3.6.3 Flag Mux Error Logging

All fault signals are sent to a flag mux component. There are four important FLAGMUX registers:

- L3_FLAGMUX_MASK0: Masks application error sources
- L3_FLAGMUX_MASK1: Masks debug error sources
- L3_FLAGMUX_REGERR0: Checks which application error sources are active
- L3_FLAGMUX_REGERR1: Checks which debug error sources are active

The two L3_FLAGMUXMASK registers mask bit 0 or bit 1 of the flag inputs, and the L3_FLAGMUX_REGERR registers read input errors. Each register is dedicated to reporting the bit corresponding to the register number.

Table 1-179 describes the mapping of the flags to the corresponding sources.

**Table 1-179. Interconnect Flag Mapping**

| | Flag Mux Input | Source |
|---|---|---|
| L3F Flag Mux | 0 | MMU |
| | 1 | DMM port1 |
| | 2 | DMM port2 |
| | 3 | TPTC3 |
| | 4 | TPTC4 |
| | 5 | DSP (GEM_SDMA) |
| | 6 | TPTC2 |
| | 7 | TPTC1 |
| | 8 | TPCC |
| | 9 | SL2 port1 |
| | 10 | Expansion slot |
| L3M Flag Mux | 0 | Cortex M3 Media Controller |
| | 1 | HDVICP config |
| | 2 | ISS |
| | 3 | L4_HS0 |
| | 4 | L4_HS1 |
| | 5 | VCP2 |
| | 6 | OCMC_RAM |
| | 7 | PCIe |
| | 8 | Reserved |
| | 9 | SGX |
| | 11 | MMCHS2 |
| | 12 | EMUSS |
| L3S Flag Mux | 0 | GPMC |
| | 1 | HDMI |
| | 2 | L4_FWCFG |
| | 3 | L4_LS0 |
| | 4 | MCBSP |
| | 5 | L4_LS1 |
| | 6 | MCASP1 |
| | 7 | MCASP2 |
| | 8 | MCASP3 |
| | 9 | USB |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-179. Interconnect Flag Mapping (continued)**

| | Flag Mux Input | Source |
|---|---|---|
| Top Flag MUX | 0 | L3F Flag Mux |
| | 1 | L3M Flag Mux |
| | 2 | L3S Flag Mux |

#### *1.12.2.3.6.4 Severity Level of Standard and Custom Errors*

The slave NIU registers are important for error logging.

- The L3_TARG_STDERRLOG_SVRTSTDLVL register shows the severity level for standard errors. According to the severity level, error logging is disabled, enabled with level ERROR, or enabled with level ERROR and flag FAULT.
- The L3_TARG_STDERRLOG_SVRTCUSTOMLVL register shows the severity level for custom errors (see Table 1-191).
- The L3_TARG_STDERRLOG_MAIN register is the main register for error-logging management. It shows the validity of the logged information, standard or custom.
- The L3_TARG_STDERRLOG_HDR register stores packets in case of a standard error.
- The L3_TARG_STDERRLOG_MSTADDR register returns the MSTADDR field of the logged packet.
- The L3_TARG_STDERRLOG_SLVADDR register returns the SLVADDR field of the stored packet.
- The L3_TARG_STDERRLOG_INFO register saves the information field of the logged packet.


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.2.4 L3 Interconnect Programming Guide

#### *1.12.2.4.1 L3 Interconnect Low-Level Programming Models*

This section describes the low-level hardware programming sequences for configuring and using the L3 interconnect module.

##### *1.12.2.4.1.1 Global Initialization*

###### *1.12.2.4.1.1.1 Global Initialization of Surrounding Modules*

This section identifies the requirements for initializing the surrounding modules when the L3 interconnect module is to be used for the first time after a device reset. The initialization of surrounding modules is based on the integration and environment of the L3interconnect. For more information, see Section 1.12.2.2, *L3 Interconnect Integration* .

**Table 1-180. Global Initialization of Surrounding Modules**

| Surrounding Modules | Comments |
|---|---|
| PRCM | For information about the configuration of the PRCM module, see the *PRCM* chapter. |
| Control module | For information about the configuration of the control module, see the *Control Module* chapter. |
| MPU INTC | The MPU INTC must be configured to enable interrupts from the L3 interconnect module. See the *ARM Interrupt Controller (AINTC)* chapter. |
| EDMA | For information about the configuration of the EDMA, see the *EDMA* chapter. |
| L3 interconnect | For information about the configuration of the interconnect, see the *L3 Interconnect* section. |

#### *1.12.2.4.2 Operational Modes Configuration*

##### *1.12.2.4.2.1 L3 Interconnect Error Analysis Mode*

###### *1.12.2.4.2.1.1 Main Sequence: L3 Interconnect Error Analysis Mode*

The information required to analyze an error source is logged in several registers. The number of registers to access depends on the error source.

Figure 1-109 shows the software sequence required in most cases.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-109. Typical Error Analysis Sequence**

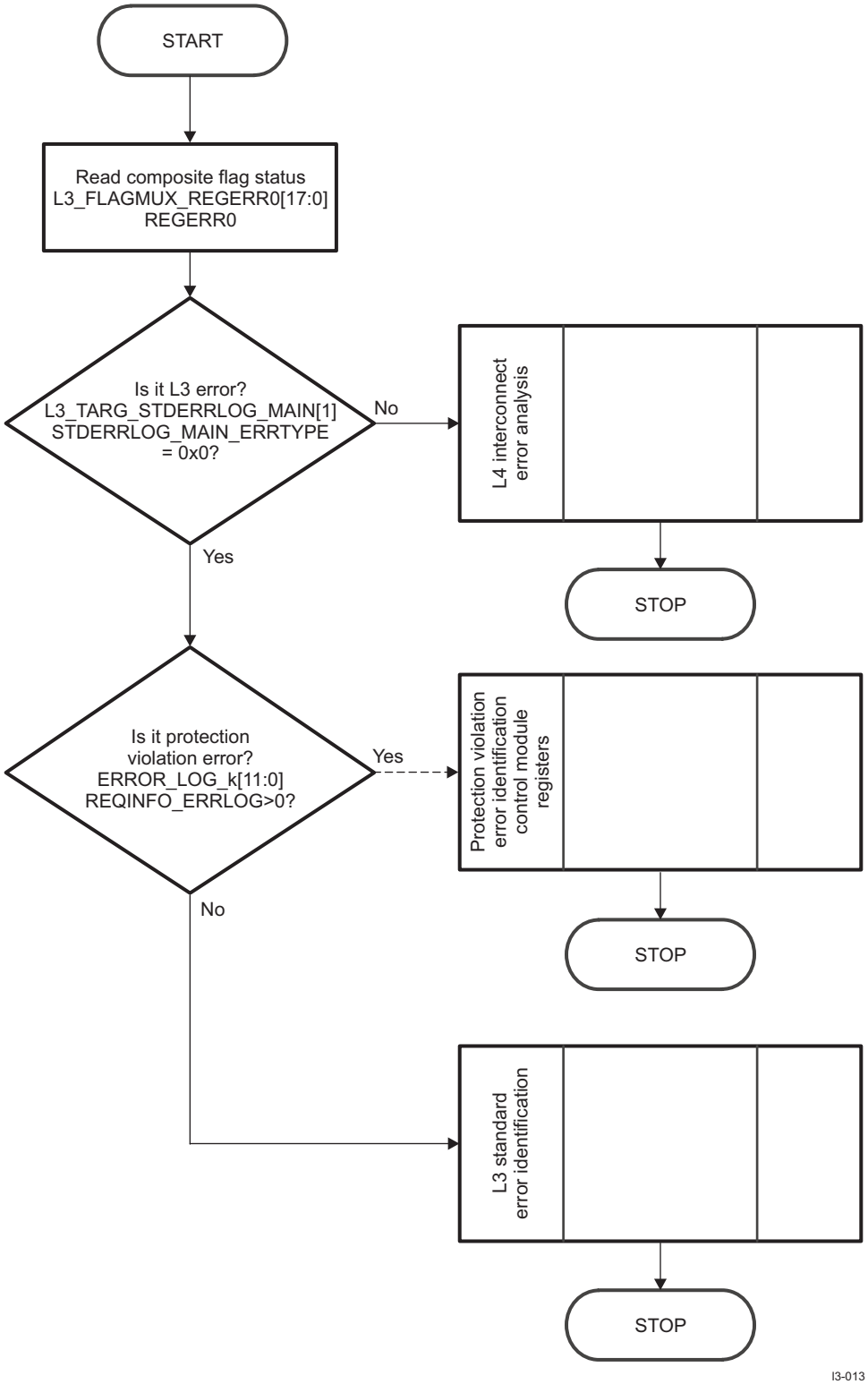



Table 1-181 lists the subprocess call summary for error analysis mode in the main sequence.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-181. Subprocess Call Summary for Main Sequence – Error Analysis Mode**

| Subprocess | Cross-Reference |
|---|---|
| L4 interconnect error analysis | See Section 1.12.2.4.2.1, *L4 Interconnect Error Analysis Mode.* |
| L3 interconnect protection violation error identification | See Section 1.12.2.4.2.1.1.2, *Subsequence: L3 Interconnect Protection Violation Error Identification.* |
| L3 interconnect unsupported command/address hole error identification | See Section 1.12.2.4.2.1.1.3, *Subsequence: L3Interconnect Unsupported Command/Address Hole Error Identification.* |
| L3 interconnect reset FLAGMUX and module | See Section 1.12.2.4.2.1.1.4, *Subsequence: L3 Interconnect Reset FLAGMUX and Module.* |

#### *1.12.2.4.2.1.1.1 Subsequence: L3 Custom Error Identification*

The procedure listed in Table 1-182 describes custom error identification.

**Table 1-182. Custom Error Identification**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| IF: Is custom error detected? | L3_TARG_STDERRLOG_MAIN[1] STDERRLOG_MAIN_ERRTYPE | =0x1 |
| Read info field of the response packet. | L3_TARG_STDERRLOG_CUSTOMINFO_INFO[7:0] STDERRLOG_CUSTOMINFO_INFO | xxx |
| Read the address of the initiator that caused error. | L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR[5:0] STDERRLOG_CUSTOMINFO_MSTADDR | xxx |
| Read the type of operation (read/write). | L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE[1:0] STDERRLOG_CUSTOMINFO_OPCODE | xxx |
| ENDIF | | |

#### *1.12.2.4.2.1.1.2 Subsequence: L3 Interconnect Protection Violation Error Identification*

The procedure listed in Table 1-183 describes protection violation error identification and where it is logged in the control module registers. Two types of errors are logged: application errors and debug errors.

**Table 1-183. Protection Violation Error Identification**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Read the burst violation. | ERROR_LOG_k[23] 2D_BURST_VIOLATION | xxx |
| Read the initiator ID. | ERROR_LOG_k[3:0] ConnID | xxx |
| Read the command that caused the error. | ERROR_LOG_k[7] MCMD | xxx |
| Read the address of the request that caused the error. | ERROR_LOG_k[31:12] SLVOFS_LOGICAL | xxx |
| **IF**: Is it an application error? | L3_FLAGMUX_REGERR0 [10:0] REGERROR0 | 0x0 |
| Clear the status bit. | L3_TARG_STDERRLOG_MAIN [31] STDERRLOG_MAIN_CLRLOG | 0x0 |
| **ELSE IF** | L3_FLAGMUX_REGERR1[10:0] REGERROR1 | = 0x1 |
| Clear the status bit. | L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_MAIN_CLRLOG | 0x0 |
| **ENDIF** | | |
| Clear the burst violation. | ERROR_LOG_k[23] 2D_BURST_VIOLATION | 0x0 |
| Clear the error status. | ERROR_LOG_k[21:17] REGION_START_ERRLOG | 0x00 |

#### *1.12.2.4.2.1.1.3 Subsequence: L3 Interconnect Standard Error Identification*

The procedure listed in Table 1-184 describes the identification of standard errors inside the L3 interconnect. The standard errors are: unsupported command, address hole, and disconnect.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-184. L3 Standard Error Identification**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| **IF**: Is an error detected? | L3_TARG_STDERRLOG_MAIN[18] STDERRLOG_MAIN_ERRCNT | = 0x1 |
| Read the corresponding flag. | L3_FLAGMUX_REGERR0[10:0] REGERROR0 | xxx |
| Read the corresponding flag. | L3_FLAGMUX_REGERR1[10:0] REGERROR1 | xxx |
| Localize the slave NIU that generated the error. | See Table 1-179, *Interconnect Flag Mapping*. | |
| **ELSE** | | |
| Clear the error log. | L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_MAIN_CLRLOG | 0x0 |
| Clear the severity error status. | L3_TARG_STDERRLOG_SVRTSTDLVL[1:0] STDERRLOG_SVRTSTDLVL_0 | 0x2 |
| **ENDIF** | | |

#### *1.12.2.4.2.1.1.4 Subsequence: L3 Interconnect FLAGMUX Configuration*

The procedure listed in Table 1-185 gives information about the configuration of FLAGMUX masks.

**Table 1-185. FLAGMUX Configuration**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Set the FLAGMUX masks to mask an event. | L3_FLAGMUX_MASK0[10:0] MASK0<br>L3_FLAGMUX_MASK1[10:0] MASK1 | xxx |
| Read the REGERR bits to see if an error is recorded. | L3_FLAGMUX_REGERR0[10:0] REGERR0<br>L3_FLAGMUX_REGERR1[10:0] REGERR1 | xxx |
| Clear the slave NIU error log and the FLAGMUX error. | L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_SVRTSTDLVL_0 | 0x1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.2.5 L3 Interconnect Register Manual

#### *1.12.2.5.1 L3 Register Group Summary*

The registers in the L3 interconnect are divided into eight groups:

- HOST registers (See Table 1-187)
- TARG registers (See Table 1-206)
- FLAGMUX registers (See Table 1-231)
- BW registers (See Table 1-241)
- STATCOLL registers (See Table 1-250)

##### *1.12.2.5.1.1 L3 Host Register Summary and Description*

**Table 1-186. HOST Instance Summary**

| Module Name | Base Address | Size |
|---|---|---|
| L3F | 0x4400 0000 | 8MB |
| L3M | 0x4440 0000 | 8MB |
| L3S | 0x44800 0000 | 8MB |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.2.5.1.1.1 L3 HOST Register Summary

**Table 1-187. HOST Registers Summary**

| Register Name | Type | Register Width (Bits) | Address Offset for L3F | L3F L3 Physical Address | Address Offset for L3M | L3M L3 Physical Address | Address Offset for L3S | L3S L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_HOST_STDHOSTHDR_COREREG | R | 32 | 0x0000 0000 | 0x4400 0000 | 0x0040 0000 | 0x4440 0000 | 0x0080 0000 | 0x4480 0000 |
| L3_HOST_STDHOSTHDR_VERSIONREG | R | 32 | 0x0000 0004 | 0x4400 0004 | 0x0040 0004 | 0x4440 0004 | 0x0080 0004 | 0x4480 0004 |
| L3_HOST_STDHOSTHDR_MAINCTLREG | R | 32 | 0x0000 0008 | 0x4400 0008 | 0x0040 0008 | 0x4440 0008 | 0x0080 0008 | 0x4480 0008 |
| L3_HOST_STDERRLOG_SVRTSTDLVL | RW | 32 | 0x0000 0040 | 0x4400 0040 | 0x0040 0040 | 0x4440 0040 | 0x0080 0040 | 0x4480 0040 |
| L3_HOST_STDERRLOG_SVRTCUSTOMLVL | RW | 32 | 0x0000 0044 | 0x4400 0044 | 0x0040 0044 | 0x4440 0044 | 0x0080 0044 | 0x4480 0044 |
| L3_HOST_STDERRLOG_MAIN | RW | 32 | 0x0000 0048 | 0x4400 0048 | 0x0040 0048 | 0x4440 0048 | 0x0080 0048 | 0x4480 0048 |
| L3_HOST_STDERRLOG_HDR | R | 32 | 0x0000 004C | 0x4400 004C | 0x0040 004C | 0x4440 004C | 0x0080 004C | 0x4480 004C |
| L3_HOST_STDERRLOG_MSTADDR | R | 32 | 0x0000 0050 | 0x4400 0050 | 0x0040 0050 | 0x4440 0050 | 0x0080 0050 | 0x4480 0050 |
| L3_HOST_STDERRLOG_SLVADDR | R | 32 | 0x0000 0054 | 0x4400 0054 | 0x0040 0054 | 0x4440 0054 | 0x0080 0054 | 0x4480 0054 |
| L3_HOST_STDERRLOG_INFO | R | 32 | 0x0000 0058 | 0x4400 0058 | 0x0040 0058 | 0x4440 0058 | 0x0080 0058 | 0x4480 0058 |
| L3_HOST_STDERRLOG_SLVOFSLSB | R | 32 | 0x0000 005C | 0x4400 005C | 0x0040 005C | 0x4440 005C | 0x0080 005C | 0x4480 005C |
| L3_HOST_STDERRLOG_SLVOFSMSB | R | 32 | 0x0000 0060 | 0x4400 0060 | 0x0040 0060 | 0x4440 0060 | 0x0080 0060 | 0x4480 0060 |
| L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR | R | 32 | 0x0000 0064 | 0x4400 0064 | 0x0040 0064 | 0x4440 0064 | 0x0080 0064 | 0x4480 0064 |
| L3_HOST_STDERRLOG_CUSTOMINFO_INFO | R | 32 | 0x0000 0068 | 0x4400 0068 | 0x0040 0068 | 0x4440 0068 | 0x0080 0068 | 0x4480 0068 |
| L3_HOST_STDERRLOG_CUSTOMINFO_WR | R | 32 | 0x0000 006C | 0x4400 006C | 0x0040 006C | 0x4440 006C | 0x0080 006C | 0x4480 006C |
| L3_HOST_STDERRLOG_CUSTOMINFO_ADDR | R | 32 | 0x0000 0070 | 0x4400 0070 | 0x0040 0070 | 0x4440 0070 | 0x0080 0070 | 0x4480 0070 |
| L3_HOST_STDERRLOG_CUSTOMINFO_DECERR | R | 32 | 0x0000 0074 | 0x4400 0074 | 0x0040 0074 | 0x4440 0074 | 0x0080 0074 | 0x4480 0074 |

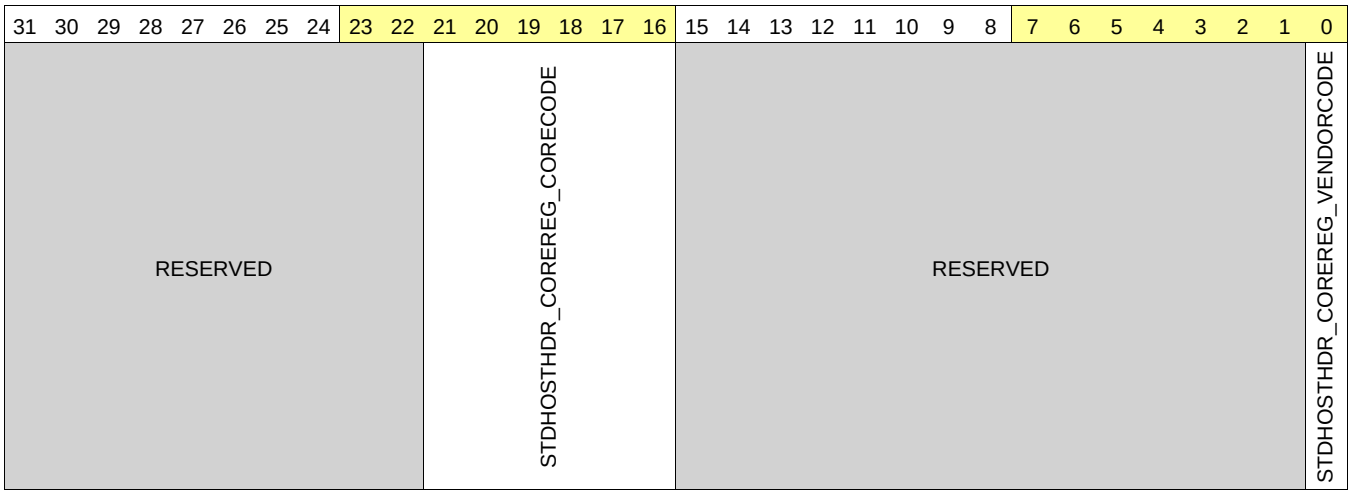


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.12.2.5.1.1.2 L3 HOST Register Description

**Table 1-188. L3_HOST_STDHOSTHDR_COREREG**

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:22 | RESERVED | | R | 0x000 |
| 21:16 | STDHOSTHDR_COREREG_CORECODE | The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x1A. | R | 0x1A |
| 15:1 | RESERVED | | R | 0x0000 |
| 0 | STDHOSTHDR_COREREG_VENDORCODE | The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1.<br>Read 0x0: Third-party vendor.<br>Read 0x1: | R | 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-189. L3_HOST_STDHOSTHDR_VERSIONREG

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STDHOSTHDR_VERSIONREG_REVISIONID | | | | | | | | STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:24 | STDHOSTHDR_VERSIONREG_REVISIONID | The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0. | R | 0x00 |
| 23:0 | STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM | Reserved. Type: Reserved. Reset value: Reserved. | R | 0x000000 |

### Table 1-190. L3_HOST_STDHOSTHDR_MAINCTLREG

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |

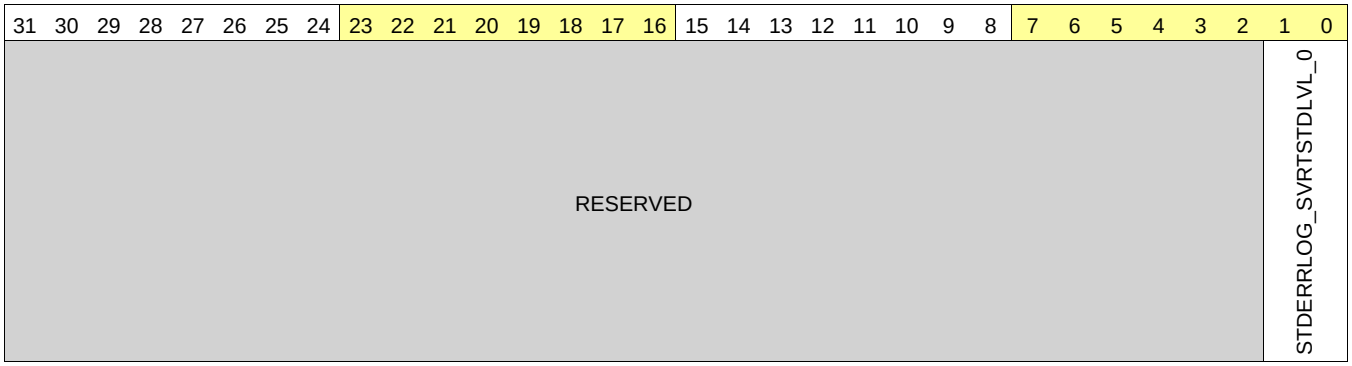
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | STDHOSTHDR_MAINCTLREG_FLT | RESERVED | |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:3 | RESERVED | | R | 0x0000 0000 |
| 2 | STDHOSTHDR_MAINCTLREG_FLT | Asserted when a Fault condition is detected: if the unit includes Error Logging, Flt is asserted when the FltCnt register field indicates a Fault, and deasserted when FltCnt is reset. If no Error Logging is implemented, this bit becomes unit-dependent. In all cases, Flt bit and Flt pin (service network) have the same logical level. Type: Status. Reset value: X. | R | 0 |
| 1:0 | RESERVED | | R | 0x0 |

**Table 1-191. L3_HOST_STDERRLOG_SVRTSTDLVL**

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | RW | | |

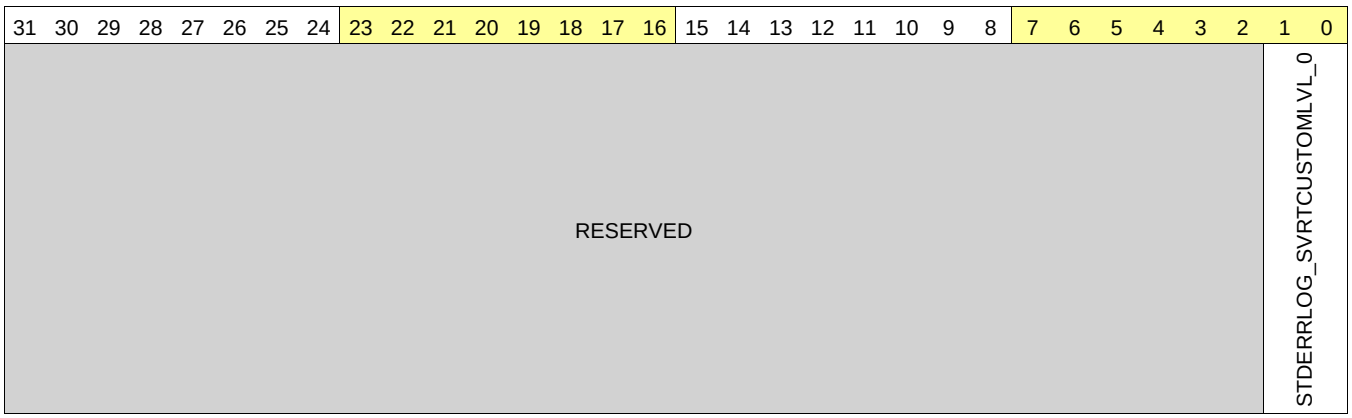



| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | RESERVED | | R | 0x0000 0000 |
| 1:0 | STDERRLOG_SVRTSTDLVL_0 | Severity level parameters Type: Control. Reset value: 0x0.<br>0x0: Error logging is disabled.<br>0x1: Errors are logged with severity level Error.<br>0x2: Errors are logged with severity level Fault. | RW | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-192. L3_HOST_STDERRLOG_SVRTCUSTOMLVL

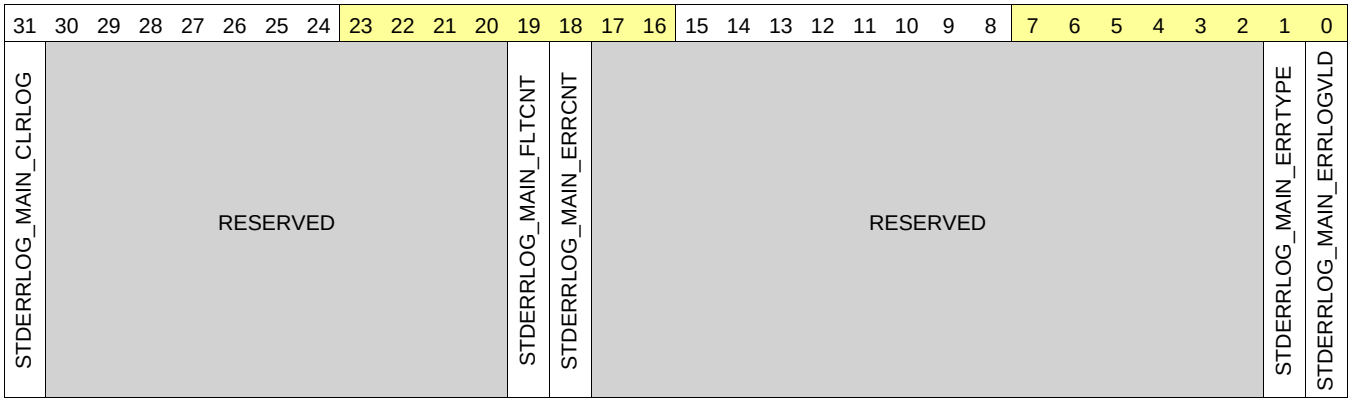
| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | RW | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | RESERVED | | R | 0x0000 0000 |
| 1:0 | STDERRLOG_SVRTCUSTOMLVL_0 | Severity level parameters Type: Control. Reset value: 0x0.<br>0x0: Error logging is disabled.<br>0x1: Errors are logged with severity level Error.<br>0x2: Errors are logged with severity level Fault. | RW | 0x0 |

### Table 1-193. L3_HOST_STDERRLOG_MAIN

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | RW | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31 | STDERRLOG_MAIN_CLRLOG | Clears Error Logging Valid bit when written to 1. Type: Give_AutoCleared. Reset value: 0x0. | RW | 0 |
| 30:20 | RESERVED | | R | 0x000 |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 19 | STDERRLOG_MAIN_FLTCNT | Asserted when at least one error with severity level FAULT is detected. Reset when written to 1. Type: Take. Reset value: 0x0. | RW | 0 |
| 18 | STDERRLOG_MAIN_ERRCNT | Asserted when at least one error with severity level ERROR is detected. Reset when written to 1. Type: Take. Reset value: 0x0. | RW | 0 |
| 17:2 | RESERVED | | R | 0x0000 |
| 1 | STDERRLOG_MAIN_ERRTYPE | Indicates logging type. Type: Status. Reset value: X.<br>Read 0x0: Logged Error format is standard (header and necker recorded).<br>Read 0x1: Logged Error format is module dependent. CustomInfo register(s) may be implemented to store additional information. | R | 0 |
| 0 | STDERRLOG_MAIN_ERRLOGVLD | Error Logging Valid. Asserted when logging information is valid(indicates that an error has been logged). Type: Status. Reset value: X. | R | 0 |

### Table 1-194. L3_HOST_STDERRLOG_HDR

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |

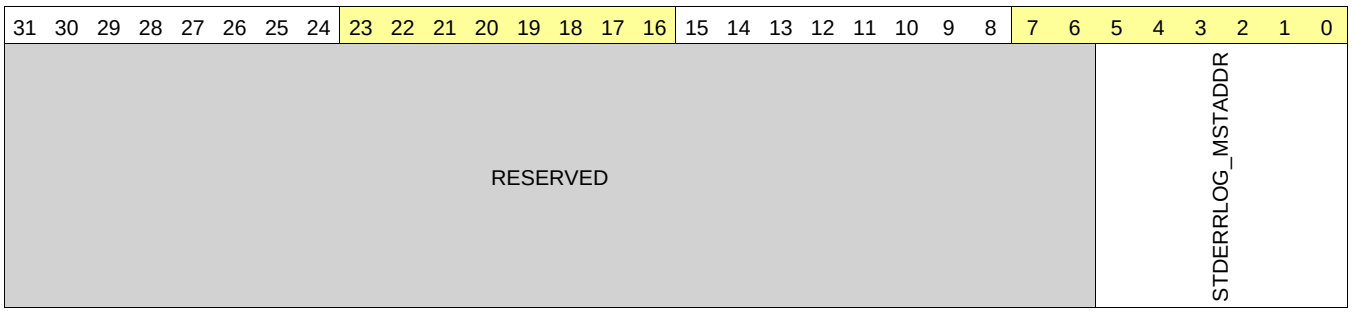
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | STDERRLOG_HDR_LEN1 | | | | | | | | RESERVED | | | | STDERRLOG_HDR_STOPOFSWRPSZ | | | | STDERRLOG_HDR_ERR | RESERVED | | | | STDERRLOG_HDR_PRESSURE | | | RESERVED | | | | STDERRLOG_HDR_OPCODE | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:26 | RESERVED | | R | 0x00 |
| 25:18 | STDERRLOG_HDR_LEN1 | This field contains the number of payload cell(s) minus one of the logged packet. Type: Status. Reset value: X. | R | 0x00 |
| 17:16 | RESERVED | | R | 0x0 |
| 15:12 | STDERRLOG_HDR_STOPOFSWRPSZ | StopOfs or WrapSize field of the logged packet (meaning depends on Wrp bit of logged opcode). Type: Status. Reset value: X. | R | 0x0 |
| 11 | STDERRLOG_HDR_ERR | Err bit of the logged packet. Type: Status. Reset value: X. | R | 0 |
| 10:8 | RESERVED | | R | 0x0 |
| 7:6 | STDERRLOG_HDR_PRESSURE | Pressure field of the logged packet. Type: Status. Reset value: X. | R | 0 |
| 5:4 | RESERVED | | R | 0x0 |
| 3:0 | STDERRLOG_HDR_OPCODE | Opcode of the logged packet. Type: Status. Reset value: X. | R | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-195. L3_HOST_STDERRLOG_MSTADDR

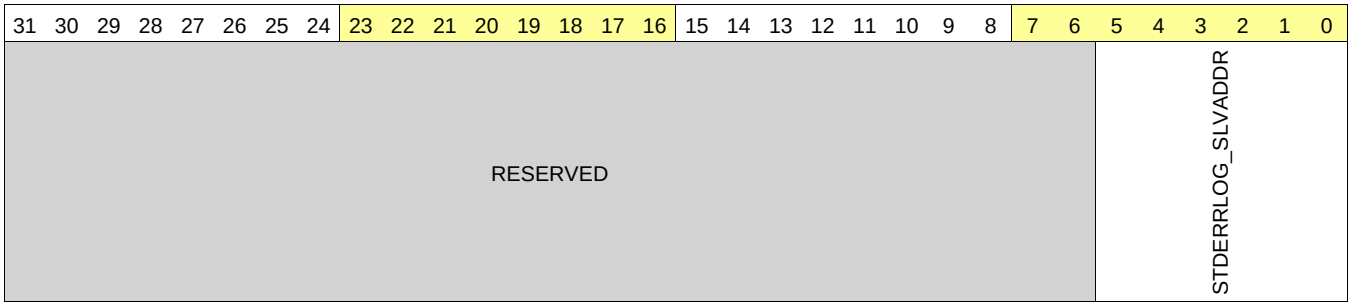
| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | RESERVED | | R | 0x000000 |
| 5:0 | STDERRLOG_MSTADDR | Master Address field of the logged packet. Type: Status. Reset value: X. | R | 0x00 |

### Table 1-196. L3_HOST_STDERRLOG_SLVADDR

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | | **Instance** | |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | RESERVED | | R | 0x0000000 |
| 5:0 | STDERRLOG_SLVADDR | Slave Address field of the logged packet. Type: Status. Reset value: X. | R | 0x00 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-197. L3_HOST_STDERRLOG_INFO

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STDERRLOG_INFO | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:8 | RESERVED | | R | 0x000000 |
| 7:0 | STDERRLOG_INFO | Info field of the logged packet. Type: Status. Reset value: X. | R | 0x00 |

### Table 1-198. L3_HOST_STDERRLOG_SLVOFSLSB

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STDERRLOG_SLVOFSLSB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | STDERRLOG_SLVOFSLSB | LSB of the slave offset field, concatenated with start offset field of the logged packet. Type: Status. Reset value: X. | R | 0x0000 0000 |

### Table 1-199. L3_HOST_STDERRLOG_SLVOFSMSB

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |

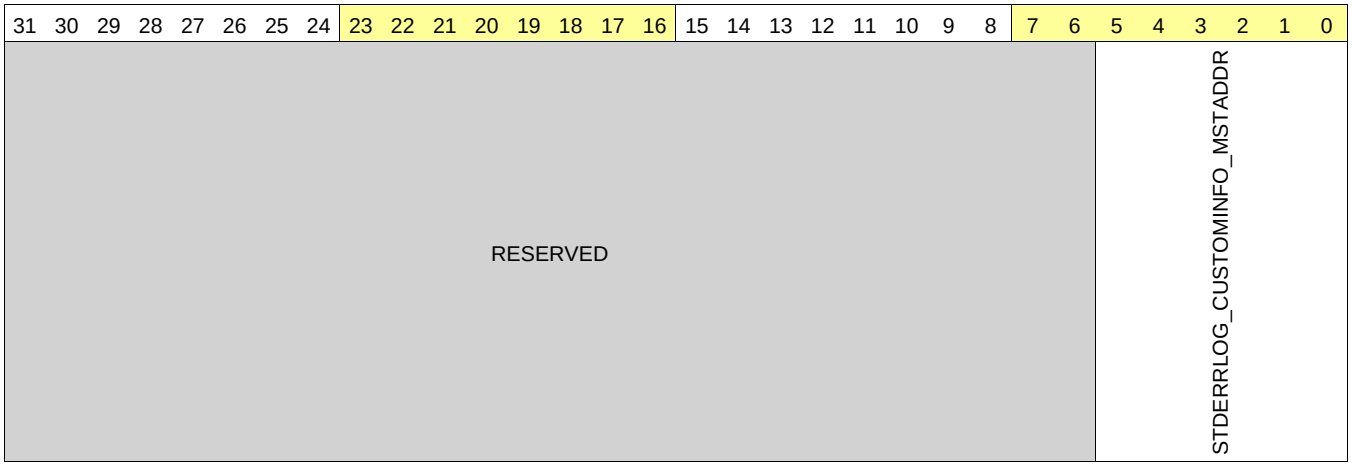
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | STDERRLOG_SLVOFSMSB |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | | R | 0x0000 0000 |
| 0 | STDERRLOG_SLVOFSMSB | MSB of the slave offset field of the logged packet (according to NTTP packet format, this register field may exceed the actual slave offset size. Unused bits are stuck at 0, if any). Type: Status. Reset value: X. | R | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-200. L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR

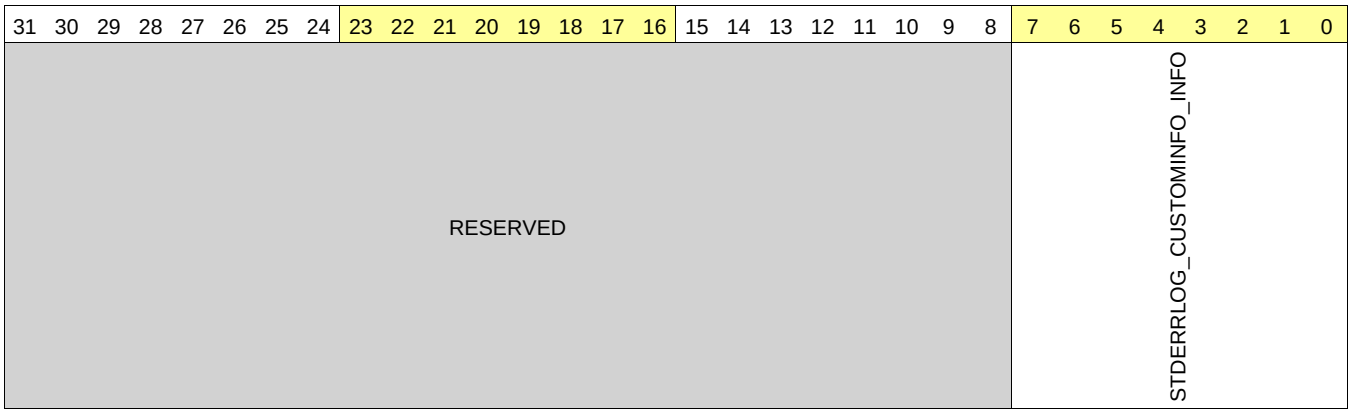
| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | RESERVED | | R | 0x000000 |
| 5:0 | STDERRLOG_CUSTOMINFO_MSTADDR | Type: Status. Reset value: X. | R | 0x00 |

### Table 1-201. L3_HOST_STDERRLOG_CUSTOMINFO_INFO

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |

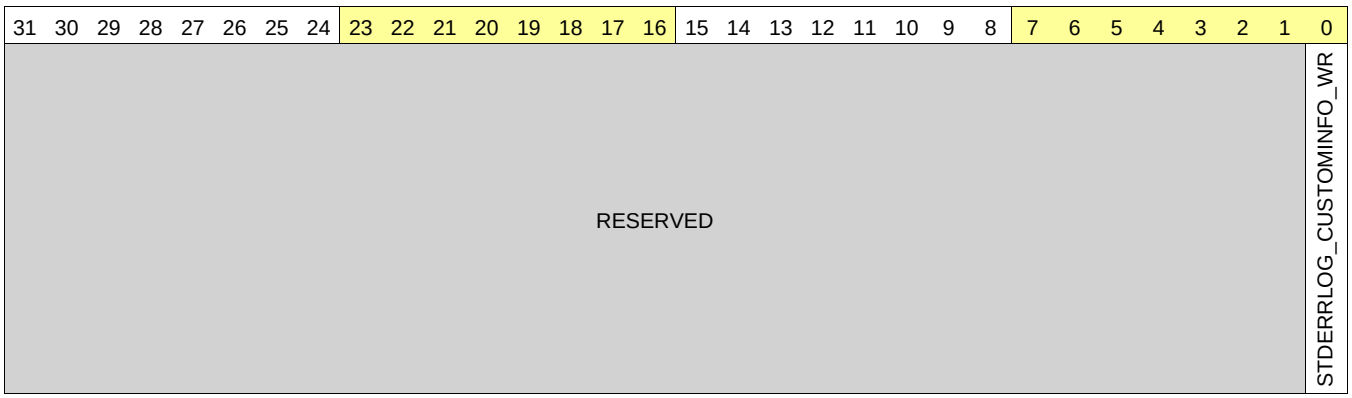



| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:8 | RESERVED | | R | 0x000000 |
| 7:0 | STDERRLOG_CUSTOMINFO_INFO | Type: Status. Reset value: X. | R | 0x00 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-202. L3_HOST_STDERRLOG_CUSTOMINFO_WR

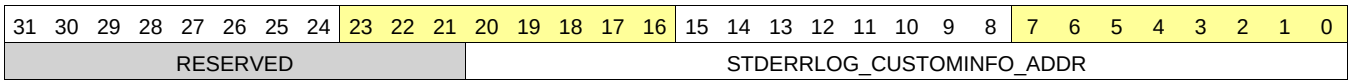
| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | | R | 0x0000 0000 |
| 0 | STDERRLOG_CUSTOMINFO_WR | Type: Status. Reset value: X. | R | 0 |

### Table 1-203. L3_HOST_STDERRLOG_CUSTOMINFO_ADDR

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |

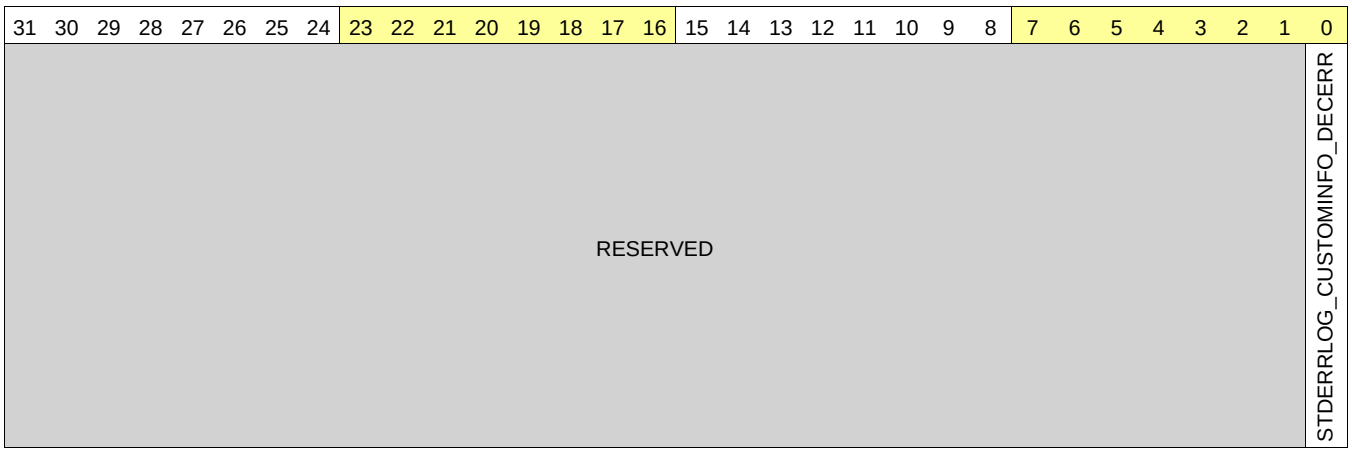



| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:21 | RESERVED | | R | 0x000 |
| 20:0 | STDERRLOG_CUSTOMINFO_ADDR | Type: Status. Reset value: X. | R | 0x000000 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-204. L3_HOST_STDERRLOG_CUSTOMINFO_DECERR**

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-187. | | |
| **Physical Address** | See Table 1-187. | **Instance** | See Table 1-187. |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | | R | 0x0000 0000 |
| 0 | STDERRLOG_CUSTOMINFO_DECERR | Type: Status. Reset value: X. | R | 0 |

### *1.12.2.5.1.2 L3 TARG Register Summary and Description*

**Table 1-205. TARG Instance Summary**

| Module Name | Base Address | Size |
|---|---|---|
| L3F_SL2_TARG | 0x4400 0300 | 4KiB |
| L3F_GEMSDMA_TARG | 0x4400 0500 | 4KiB |
| L3F_EXP_TARG | 0x4400 0600 | 4KiB |
| L3F_TPTC1_TARG | 0x4400 0700 | 4KiB |
| L3F_TPTC2_TARG | 0x4400 0800 | 4KiB |
| L3F_TPTC3_TARG | 0x4400 0900 | 4KiB |
| L3F_TPTC4_TARG | 0x4400 0A00 | 4KiB |
| L3F_TPCC_TARG | 0x4400 0B00 | 4KiB |
| L3F_MMU_TARG | 0x4400 0C00 | 4KiB |
| L3F_MEDIACONTROLLER_TARG | 0x4440 0100 | 4KiB |
| L3F_SGX_TARG | 0x4440 0200 | 4KiB |
| L3F_PCIE_TARG | 0x4440 0300 | 4KiB |
| L3F_OCMCRAM0_TARG | 0x4440 0400 | 4KiB |
| L3F_VCP2_TARG | 0x4440 0500 | 4KiB |
| L3F_HDVICP_TARG | 0x4440 0600 | 4KiB |
| L3F_ISS_TARG | 0x4440 0700 | 4KiB |
| L3F_L4HS0_TARG | 0x4440 0900 | 4KiB |
| L3F_L4HS1_TARG | 0x4440 0B00 | 4KiB |
| L3F_MMC2_TARG | 0x4440 0C00 | 4KiB |
| L3F_EMUSS_TARG | 0x4440 0D00 | 4KiB |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-205. TARG Instance Summary (continued)**

| Module Name | Base Address | Size |
|---|---|---|
| L3F_DMM1_TARG | 0x4440 1200 | 4KiB |
| L3F_DMM2_TARG | 0x4440 1300 | 4KiB |
| L3S_MCBSP_TARG | 0x4480 0200 | 4KiB |
| L3S_HDMI_TARG | 0x4480 0300 | 4KiB |
| L3S_L4FW_TARG | 0x4480 0400 | 4KiB |
| L3S_L4LS0_TARG | 0x4480 0500 | 4KiB |
| L3S_L4LS1_TARG | 0x4480 0600 | 4KiB |
| L3S_GPMC_TARG | 0x4480 0700 | 4KiB |
| L3S_MCASP1_TARG | 0x4480 0800 | 4KiB |
| L3S_MCASP2_TARG | 0x4480 0900 | 4KiB |
| L3S_MCASP3_TARG | 0x4480 0A00 | 4KiB |
| L3S_USB_TARG | 0x4480 0B00 | 4KiB |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.2.5.1.2.1 L3 TARG Registers Summary

**Table 1-206. TARG Registers Summary**

| Register Name | Type | Register Width (Bits) | Address Offset | L3F_SL2_TARG L3 Physical Address | L3F_GEMSDMA_TARG L3 Physical Address | L3F_EXP_TARG L3 Physical Address | L3F_TPTC1_TARG L3 Physical Address | L3F_TPTC2_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_TARG_STDHOSTHDR_COREREG | R | 32 | 0x0000 0000 | 0x4400 0300 | 0x4400 0500 | 0x4400 0600 | 0x4400 0700 | 0x4400 0800 |
| L3_TARG_STDHOSTHDR_VERSIONREG | R | 32 | 0x0000 0004 | 0x4400 0304 | 0x4400 0504 | 0x4400 0604 | 0x4400 0704 | 0x4400 0804 |
| L3_TARG_STDHOSTHDR_MAINCTLREG | RW | 32 | 0x0000 0008 | 0x4400 0308 | 0x4400 0508 | 0x4400 0608 | 0x4400 0708 | 0x4400 0808 |
| L3_TARG_STDHOSTHDR_NTTPADDR_0 | R | 32 | 0x0000 0010 | 0x4400 0310 | 0x4400 0510 | 0x4400 0610 | 0x4400 0710 | 0x4400 0810 |
| L3_TARG_STDERRLOG_SVRTSTDLVL | RW | 32 | 0x0000 0040 | 0x4400 0340 | 0x4400 0540 | 0x4400 0640 | 0x4400 0740 | 0x4400 0840 |
| L3_TARG_STDERRLOG_SVRTCUSTOMLVL | RW | 32 | 0x0000 0044 | 0x4400 0344 | 0x4400 0544 | 0x4400 0644 | 0x4400 0744 | 0x4400 0844 |
| L3_TARG_STDERRLOG_MAIN | RW | 32 | 0x0000 0048 | 0x4400 0348 | 0x4400 0548 | 0x4400 0648 | 0x4400 0748 | 0x4400 0848 |
| L3_TARG_STDERRLOG_HDR | R | 32 | 0x0000 004C | 0x4400 034C | 0x4400 054C | 0x4400 064C | 0x4400 074C | 0x4400 084C |
| L3_TARG_STDERRLOG_MSTADDR | R | 32 | 0x0000 0050 | 0x4400 0350 | 0x4400 0550 | 0x4400 0650 | 0x4400 0750 | 0x4400 0850 |
| L3_TARG_STDERRLOG_SLVADDR | R | 32 | 0x0000 0054 | 0x4400 0354 | 0x4400 0554 | 0x4400 0654 | 0x4400 0754 | 0x4400 0854 |
| L3_TARG_STDERRLOG_INFO | R | 32 | 0x0000 0058 | 0x4400 0358 | 0x4400 0558 | 0x4400 0658 | 0x4400 0758 | 0x4400 0858 |
| L3_TARG_STDERRLOG_SLVOFSLSB | R | 32 | 0x0000 005C | 0x4400 035C | 0x4400 055C | 0x4400 065C | 0x4400 075C | 0x4400 085C |
| L3_TARG_STDERRLOG_SLVOFSMSB | R | 32 | 0x0000 0060 | 0x4400 0360 | 0x4400 0560 | 0x4400 0660 | 0x4400 0760 | 0x4400 0860 |
| L3_TARG_STDERRLOG_CUSTOMINFO_INFO | R | 32 | 0x0000 0064 | 0x4400 0364 | 0x4400 0564 | 0x4400 0664 | 0x4400 0764 | 0x4400 0864 |
| L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR | R | 32 | 0x0000 0068 | 0x4400 0368 | 0x4400 0568 | 0x4400 0668 | 0x4400 0768 | 0x4400 0868 |
| L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE | R | 32 | 0x0000 006C | 0x4400 036C | 0x4400 056C | 0x4400 066C | 0x4400 076C | 0x4400 086C |
| L3_TARG_ADDRSPACESIZELOG | RW | 32 | 0x0000 0080 | 0x4400 0380 | 0x4400 0580 | 0x4400 0680 | 0x4400 0780 | 0x4400 0880 |

**Table 1-207. TARG Registers Summary**

| Register Name | Type | Register Width (Bits) | Address Offset | L3F_TPTC3_TARG L3 Physical Address | L3F_TPTC4_TARG L3 Physical Address | L3F_TPCC_TARG L3 Physical Address | L3F_MMU_TARG L3 Physical Address | L3F_MEDIACONTROLLER_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_TARG_STDHOSTHDR_COREREG | R | 32 | 0x0000 0000 | 0x4400 0900 | 0x4400 0A00 | 0x4400 0B00 | 0x4400 0C00 | 0x4440 0100 |
| L3_TARG_STDHOSTHDR_VERSIONREG | R | 32 | 0x0000 0004 | 0x4400 0904 | 0x4400 0A04 | 0x4400 0B04 | 0x4400 0C04 | 0x4440 0104 |
| L3_TARG_STDHOSTHDR_MAINCTLREG | RW | 32 | 0x0000 0008 | 0x4400 0908 | 0x4400 0A08 | 0x4400 0B08 | 0x4400 0C08 | 0x4440 0108 |
| L3_TARG_STDHOSTHDR_NTTPADDR_0 | R | 32 | 0x0000 0010 | 0x4400 0910 | 0x4400 0A10 | 0x4400 0B10 | 0x4400 0C10 | 0x4440 0110 |
| L3_TARG_STDERRLOG_SVRTSTDLVL | RW | 32 | 0x0000 0040 | 0x4400 0940 | 0x4400 0A40 | 0x4400 0B40 | 0x4400 0C40 | 0x4440 0140 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-207. TARG Registers Summary (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset | L3F_TPTC3_TARG L3 Physical Address | L3F_TPTC4_TARG L3 Physical Address | L3F_TPCC_TARG L3 Physical Address | L3F_MMU_TARG L3 Physical Address | L3F_MEDIACONTROLLER_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_TARG_STDERRLOG_SVRTCUSTOMLVL | RW | 32 | 0x0000 0044 | 0x4400 0944 | 0x4400 0A44 | 0x4400 0B44 | 0x4400 0C44 | 0x4440 0144 |
| L3_TARG_STDERRLOG_MAIN | RW | 32 | 0x0000 0048 | 0x4400 0948 | 0x4400 0A48 | 0x4400 0B48 | 0x4400 0C48 | 0x4440 0148 |
| L3_TARG_STDERRLOG_HDR | R | 32 | 0x0000 004C | 0x4400 094C | 0x4400 0A4C | 0x4400 0B4C | 0x4400 0C4C | 0x4440 014C |
| L3_TARG_STDERRLOG_MSTADDR | R | 32 | 0x0000 0050 | 0x4400 0950 | 0x4400 0A50 | 0x4400 0B50 | 0x4400 0C50 | 0x4440 0150 |
| L3_TARG_STDERRLOG_SLVADDR | R | 32 | 0x0000 0054 | 0x4400 0954 | 0x4400 0A54 | 0x4400 0B54 | 0x4400 0C54 | 0x4440 0154 |
| L3_TARG_STDERRLOG_INFO | R | 32 | 0x0000 0058 | 0x4400 0958 | 0x4400 0A58 | 0x4400 0B58 | 0x4400 0C58 | 0x4440 0158 |
| L3_TARG_STDERRLOG_SLVOFSLSB | R | 32 | 0x0000 005C | 0x4400 095C | 0x4400 0A5C | 0x4400 0B5C | 0x4400 0C5C | 0x4440 015C |
| L3_TARG_STDERRLOG_SLVOFSMSB | R | 32 | 0x0000 0060 | 0x4400 0960 | 0x4400 0A60 | 0x4400 0B60 | 0x4400 0C60 | 0x4440 0160 |
| L3_TARG_STDERRLOG_CUSTOMINFO_INFO | R | 32 | 0x0000 0064 | 0x4400 0964 | 0x4400 0A64 | 0x4400 0B64 | 0x4400 0C64 | 0x4440 0164 |
| L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR | R | 32 | 0x0000 0068 | 0x4400 0968 | 0x4400 0A68 | 0x4400 0B68 | 0x4400 0C68 | 0x4440 0168 |
| L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE | R | 32 | 0x0000 006C | 0x4400 096C | 0x4400 0A6C | 0x4400 0B6C | 0x4400 0C6C | 0x4440 016C |
| L3_TARG_ADDRSPACESIZELOG | RW | 32 | 0x0000 0080 | 0x4400 0980 | 0x4400 0A80 | 0x4400 0B80 | 0x4400 0C80 | 0x4440 0180 |

**Table 1-208. TARG Registers Summary**

| Register Name | Type | Register Width (Bits) | Address Offset | L3F_SGX_TARG L3 Physical Address | L3F_PCIE_TARG L3 Physical Address | L3F_OCMCRAM0_TARG L3 Physical Address | L3F_VCP2_TARG L3 Physical Address | L3F_HDVICP_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_TARG_STDHOSTHDR_COREREG | R | 32 | 0x0080 0300 | 0x4440 0200 | 0x4440 0300 | 0x4440 0400 | 0x4440 0500 | 0x4440 0600 |
| L3_TARG_STDHOSTHDR_VERSIONREG | R | 32 | 0x0080 0304 | 0x4440 0204 | 0x4440 0304 | 0x4440 0404 | 0x4440 0504 | 0x4440 0604 |
| L3_TARG_STDHOSTHDR_MAINCTLREG | RW | 32 | 0x0080 0308 | 0x4440 0208 | 0x4440 0308 | 0x4440 0408 | 0x4440 0508 | 0x4440 0608 |
| L3_TARG_STDHOSTHDR_NTTPADDR_0 | R | 32 | 0x0080 0310 | 0x4440 0210 | 0x4440 0310 | 0x4440 0410 | 0x4440 0510 | 0x4440 0610 |
| L3_TARG_STDERRLOG_SVRTSTDLVL | RW | 32 | 0x0080 0340 | 0x4440 0240 | 0x4440 0340 | 0x4440 0440 | 0x4440 0540 | 0x4440 0640 |
| L3_TARG_STDERRLOG_SVRTCUSTOMLVL | RW | 32 | 0x0080 0344 | 0x4440 0244 | 0x4440 0344 | 0x4440 0444 | 0x4440 0544 | 0x4440 0644 |
| L3_TARG_STDERRLOG_MAIN | RW | 32 | 0x0080 0348 | 0x4440 0248 | 0x4440 0348 | 0x4440 0448 | 0x4440 0548 | 0x4440 0648 |
| L3_TARG_STDERRLOG_HDR | R | 32 | 0x0080 034C | 0x4440 024C | 0x4440 034C | 0x4440 044C | 0x4440 054C | 0x4440 064C |
| L3_TARG_STDERRLOG_MSTADDR | R | 32 | 0x0080 0350 | 0x4440 0250 | 0x4440 0350 | 0x4440 0450 | 0x4440 0550 | 0x4440 0650 |
| L3_TARG_STDERRLOG_SLVADDR | R | 32 | 0x0080 0354 | 0x4440 0254 | 0x4440 0354 | 0x4440 0454 | 0x4440 0554 | 0x4440 0654 |
| L3_TARG_STDERRLOG_INFO | R | 32 | 0x0080 0358 | 0x4440 0258 | 0x4440 0358 | 0x4440 0458 | 0x4440 0558 | 0x4440 0658 |
| L3_TARG_STDERRLOG_SLVOFSLSB | R | 32 | 0x0080 035C | 0x4440 025C | 0x4440 035C | 0x4440 045C | 0x4440 055C | 0x4440 065C |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-208. TARG Registers Summary (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset | L3F_SGX_TARG L3 Physical Address | L3F_PCIE_TARG L3 Physical Address | L3F_OCMCRAM0_TARG L3 Physical Address | L3F_VCP2_TARG L3 Physical Address | L3F_HDVICP_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_TARG_STDERRLOG_SLVOFSMSB | R | 32 | 0x0080 0360 | 0x4440 0260 | 0x4440 0360 | 0x4440 0460 | 0x4440 0560 | 0x4440 0660 |
| L3_TARG_STDERRLOG_CUSTOMINFO_INFO | R | 32 | 0x0080 0364 | 0x4440 0264 | 0x4440 0364 | 0x4440 0464 | 0x4440 0564 | 0x4440 0664 |
| L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR | R | 32 | 0x0080 0368 | 0x4440 0268 | 0x4440 0368 | 0x4440 0468 | 0x4440 0568 | 0x4440 0668 |
| L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE | R | 32 | 0x0080 036C | 0x4440 026C | 0x4440 036C | 0x4440 046C | 0x4440 056C | 0x4440 066C |
| L3_TARG_ADDRSPACESIZELOG | RW | 32 | 0x0080 0380 | 0x4440 0280 | 0x4440 0380 | 0x4440 0480 | 0x4440 0580 | 0x4440 0680 |

**Table 1-209. TARG Registers Summary**

| Register Name | Type | Register Width (Bits) | Address Offset for CLK2_SGX | L3F_ISS_TARG L3 Physical Address | L3F_L4HS0_TARG L3 Physical Address | L3F_L4HS1_TARG L3 Physical Address | L3F_MMC2_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L3_TARG_STDHOSTHDR_COREREG | R | 32 | 0x0000 0000 | 0x4440 0700 | 0x4440 0900 | 0x4440 0B00 | 0x4440 0C00 |
| L3_TARG_STDHOSTHDR_VERSIONREG | R | 32 | 0x0000 0004 | 0x4440 0704 | 0x4440 0904 | 0x4440 0B04 | 0x4440 0C04 |
| L3_TARG_STDHOSTHDR_MAINCTLREG | RW | 32 | 0x0000 0008 | 0x4440 0708 | 0x4440 0908 | 0x4440 0B08 | 0x4440 0C08 |
| L3_TARG_STDHOSTHDR_NTTPADDR_0 | R | 32 | 0x0000 0010 | 0x4440 0710 | 0x4440 0910 | 0x4440 0B10 | 0x4440 0C10 |
| L3_TARG_STDERRLOG_SVRTSTDLVL | RW | 32 | 0x0000 0040 | 0x4440 0740 | 0x4440 0940 | 0x4440 0B40 | 0x4440 0C40 |
| L3_TARG_STDERRLOG_SVRTCUSTOMLVL | RW | 32 | 0x0000 0044 | 0x4440 0744 | 0x4440 0944 | 0x4440 0B44 | 0x4440 0C44 |
| L3_TARG_STDERRLOG_MAIN | RW | 32 | 0x0000 0048 | 0x4440 0748 | 0x4440 0948 | 0x4440 0B48 | 0x4440 0C48 |
| L3_TARG_STDERRLOG_HDR | R | 32 | 0x0000 004C | 0x4440 074C | 0x4440 094C | 0x4440 0B4C | 0x4440 0C4C |
| L3_TARG_STDERRLOG_MSTADDR | R | 32 | 0x0000 0050 | 0x4440 0750 | 0x4440 0950 | 0x4440 0B50 | 0x4440 0C50 |
| L3_TARG_STDERRLOG_SLVADDR | R | 32 | 0x0000 0054 | 0x4440 0754 | 0x4440 0954 | 0x4440 0B54 | 0x4440 0C54 |
| L3_TARG_STDERRLOG_INFO | R | 32 | 0x0000 0058 | 0x4440 0758 | 0x4440 0958 | 0x4440 0B58 | 0x4440 0C58 |
| L3_TARG_STDERRLOG_SLVOFSLSB | R | 32 | 0x0000 005C | 0x4440 075C | 0x4440 095C | 0x4440 0B5C | 0x4440 0C5C |
| L3_TARG_STDERRLOG_SLVOFSMSB | R | 32 | 0x0000 0060 | 0x4440 0760 | 0x4440 0960 | 0x4440 0B60 | 0x4440 0C60 |
| L3_TARG_STDERRLOG_CUSTOMINFO_INFO | R | 32 | 0x0000 0064 | 0x4440 0764 | 0x4440 0964 | 0x4440 0B64 | 0x4440 0C64 |
| L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR | R | 32 | 0x0000 0068 | 0x4440 0768 | 0x4440 0968 | 0x4440 0B68 | 0x4440 0C68 |
| L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE | R | 32 | 0x0000 006C | 0x4440 076C | 0x4440 096C | 0x4440 0B6C | 0x4440 0C6C |
| L3_TARG_ADDRSPACESIZELOG | RW | 32 | 0x0000 0080 | 0x4440 0780 | 0x4440 0980 | 0x4440 0B80 | 0x4440 0C80 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-210. TARG Registers Summary**

| Register Name | Type | Register Width (Bits) | Address Offset | L3F_EMUSS_TARG L3 Physical Address | L3F_DMM1_TARG L3 Physical Address | L3F_DMM2_TARG L3 Physical Address | L3S_MCBSP_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L3_TARG_STDHOSTHDR_COREREG | R | 32 | 0x0000 0000 | 0x4440 0D00 | 0x4440 1200 | 0x4440 1300 | 0x4480 0200 |
| L3_TARG_STDHOSTHDR_VERSIONREG | R | 32 | 0x0000 0004 | 0x4440 0D04 | 0x4440 1204 | 0x4440 1304 | 0x4480 0204 |
| L3_TARG_STDHOSTHDR_MAINCTLREG | RW | 32 | 0x0000 0008 | 0x4440 0D08 | 0x4440 1208 | 0x4440 1308 | 0x4480 0208 |
| L3_TARG_STDHOSTHDR_NTTPADDR_0 | R | 32 | 0x0000 0010 | 0x4440 0D10 | 0x4440 1210 | 0x4440 1310 | 0x4480 0210 |
| L3_TARG_STDERRLOG_SVRTSTDLVL | RW | 32 | 0x0000 0040 | 0x4440 0D40 | 0x4440 1240 | 0x4440 1340 | 0x4480 0240 |
| L3_TARG_STDERRLOG_SVRTCUSTOMLVL | RW | 32 | 0x0000 0044 | 0x4440 0D44 | 0x4440 1244 | 0x4440 1344 | 0x4480 0244 |
| L3_TARG_STDERRLOG_MAIN | RW | 32 | 0x0000 0048 | 0x4440 0D48 | 0x4440 1248 | 0x4440 1348 | 0x4480 0248 |
| L3_TARG_STDERRLOG_HDR | R | 32 | 0x0000 004C | 0x4440 0D4C | 0x4440 124C | 0x4440 134C | 0x4480 024C |
| L3_TARG_STDERRLOG_MSTADDR | R | 32 | 0x0000 0050 | 0x4440 0D50 | 0x4440 1250 | 0x4440 1350 | 0x4480 0250 |
| L3_TARG_STDERRLOG_SLVADDR | R | 32 | 0x0000 0054 | 0x4440 0D54 | 0x4440 1254 | 0x4440 1354 | 0x4480 0254 |
| L3_TARG_STDERRLOG_INFO | R | 32 | 0x0000 0058 | 0x4440 0D58 | 0x4440 1258 | 0x4440 1358 | 0x4480 0258 |
| L3_TARG_STDERRLOG_SLVOFSLSB | R | 32 | 0x0000 005C | 0x4440 0D5C | 0x4440 125C | 0x4440 135C | 0x4480 025C |
| L3_TARG_STDERRLOG_SLVOFSMSB | R | 32 | 0x0000 0060 | 0x4440 0D60 | 0x4440 1260 | 0x4440 1360 | 0x4480 0260 |
| L3_TARG_STDERRLOG_CUSTOMINFO_INFO | R | 32 | 0x0000 0064 | 0x4440 0D64 | 0x4440 1264 | 0x4440 1364 | 0x4480 0264 |
| L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR | R | 32 | 0x0000 0068 | 0x4440 0D68 | 0x4440 1268 | 0x4440 1368 | 0x4480 0268 |
| L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE | R | 32 | 0x0000 006C | 0x4440 0D6C | 0x4440 126C | 0x4440 136C | 0x4480 026C |
| L3_TARG_ADDRSPACESIZELOG | RW | 32 | 0x0000 0080 | 0x4440 0D80 | 0x4440 1280 | 0x4440 1380 | 0x4480 0280 |

**Table 1-211. TARG Registers Summary**

| Register Name | Type | Register Width (Bits) | Address Offset for CLK2_L4PER3 | L3S_HDMI_TARG L3 Physical Address | L3S_L4FW_TARG L3 Physical Address | L3S_L4LS0_TARG L3 Physical Address | L3S_L4LS1_TARG L3 Physical Address | L3S_GPMC_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_TARG_STDHOSTHDR_COREREG | R | 32 | 0x0000 0000 | 0x4480 0300 | 0x4480 0400 | 0x4480 0500 | 0x4480 0600 | 0x4480 0700 |
| L3_TARG_STDHOSTHDR_VERSIONREG | R | 32 | 0x0000 0004 | 0x4480 0304 | 0x4480 0404 | 0x4480 0504 | 0x4480 0604 | 0x4480 0704 |
| L3_TARG_STDHOSTHDR_MAINCTLREG | RW | 32 | 0x0000 0008 | 0x4480 0308 | 0x4480 0408 | 0x4480 0508 | 0x4480 0608 | 0x4480 0708 |
| L3_TARG_STDHOSTHDR_NTTPADDR_0 | R | 32 | 0x0000 0010 | 0x4480 0310 | 0x4480 0410 | 0x4480 0510 | 0x4480 0610 | 0x4480 0710 |
| L3_TARG_STDERRLOG_SVRTSTDLVL | RW | 32 | 0x0000 0040 | 0x4480 0340 | 0x4480 0440 | 0x4480 0540 | 0x4480 0640 | 0x4480 0740 |
| L3_TARG_STDERRLOG_SVRTCUSTOMLVL | RW | 32 | 0x0000 0044 | 0x4480 0344 | 0x4480 0444 | 0x4480 0544 | 0x4480 0644 | 0x4480 0744 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-211. TARG Registers Summary (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset for CLK2_L4PER3 | L3S_HDMI_TARG L3 Physical Address | L3S_L4FW_TARG L3 Physical Address | L3S_L4LS0_TARG L3 Physical Address | L3S_L4LS1_TARG L3 Physical Address | L3S_GPMC_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_TARG_STDERRLOG_MAIN | RW | 32 | 0x0000 0048 | 0x4480 0348 | 0x4480 0448 | 0x4480 0548 | 0x4480 0648 | 0x4480 0748 |
| L3_TARG_STDERRLOG_HDR | R | 32 | 0x0000 004C | 0x4480 034C | 0x4480 044C | 0x4480 054C | 0x4480 064C | 0x4480 074C |
| L3_TARG_STDERRLOG_MSTADDR | R | 32 | 0x0000 0050 | 0x4480 0350 | 0x4480 0450 | 0x4480 0550 | 0x4480 0650 | 0x4480 0750 |
| L3_TARG_STDERRLOG_SLVADDR | R | 32 | 0x0000 0054 | 0x4480 0354 | 0x4480 0454 | 0x4480 0554 | 0x4480 0654 | 0x4480 0754 |
| L3_TARG_STDERRLOG_INFO | R | 32 | 0x0000 0058 | 0x4480 0358 | 0x4480 0458 | 0x4480 0558 | 0x4480 0658 | 0x4480 0758 |
| L3_TARG_STDERRLOG_SLVOFSLSB | R | 32 | 0x0000 005C | 0x4480 035C | 0x4480 045C | 0x4480 055C | 0x4480 065C | 0x4480 075C |
| L3_TARG_STDERRLOG_SLVOFSMSB | R | 32 | 0x0000 0060 | 0x4480 0360 | 0x4480 0460 | 0x4480 0560 | 0x4480 0660 | 0x4480 0760 |
| L3_TARG_STDERRLOG_CUSTOMINFO_INFO | R | 32 | 0x0000 0064 | 0x4480 0364 | 0x4480 0464 | 0x4480 0564 | 0x4480 0664 | 0x4480 0764 |
| L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR | R | 32 | 0x0000 0068 | 0x4480 0368 | 0x4480 0468 | 0x4480 0568 | 0x4480 0668 | 0x4480 0768 |
| L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE | R | 32 | 0x0000 006C | 0x4480 036C | 0x4480 046C | 0x4480 056C | 0x4480 066C | 0x4480 076C |
| L3_TARG_ADDRSPACESIZELOG | RW | 32 | 0x0000 0080 | 0x4480 0380 | 0x4480 0480 | 0x4480 0580 | 0x4480 0680 | 0x4480 0780 |

**Table 1-212. TARG Registers Summary**

| Register Name | Type | Register Width (Bits) | Address Offset for CLK2_C2C | L3S_MCASP1_TARG L3 Physical Address | L3S_MCASP2_TARG L3 Physical Address | L3S_MCASP3_TARG L3 Physical Address | L3S_USB_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L3_TARG_STDHOSTHDR_COREREG | R | 32 | 0x0000 0000 | 0x4480 0800 | 0x4480 0900 | 0x4480 0A00 | 0x4480 0B00 |
| L3_TARG_STDHOSTHDR_VERSIONREG | R | 32 | 0x0000 0004 | 0x4480 0804 | 0x4480 0904 | 0x4480 0A04 | 0x4480 0B04 |
| L3_TARG_STDHOSTHDR_MAINCTLREG | RW | 32 | 0x0000 0008 | 0x4480 0808 | 0x4480 0908 | 0x4480 0A08 | 0x4480 0B08 |
| L3_TARG_STDHOSTHDR_NTTPADDR_0 | R | 32 | 0x0000 0010 | 0x4480 0810 | 0x4480 0910 | 0x4480 0A10 | 0x4480 0B10 |
| L3_TARG_STDERRLOG_SVRTSTDLVL | RW | 32 | 0x0000 0040 | 0x4480 0840 | 0x4480 0940 | 0x4480 0A40 | 0x4480 0B40 |
| L3_TARG_STDERRLOG_SVRTCUSTOMLVL | RW | 32 | 0x0000 0044 | 0x4480 0844 | 0x4480 0944 | 0x4480 0A44 | 0x4480 0B44 |
| L3_TARG_STDERRLOG_MAIN | RW | 32 | 0x0000 0048 | 0x4480 0848 | 0x4480 0948 | 0x4480 0A48 | 0x4480 0B48 |
| L3_TARG_STDERRLOG_HDR | R | 32 | 0x0000 004C | 0x4480 084C | 0x4480 094C | 0x4480 0A4C | 0x4480 0B4C |
| L3_TARG_STDERRLOG_MSTADDR | R | 32 | 0x0000 0050 | 0x4480 0850 | 0x4480 0950 | 0x4480 0A50 | 0x4480 0B50 |
| L3_TARG_STDERRLOG_SLVADDR | R | 32 | 0x0000 0054 | 0x4480 0854 | 0x4480 0954 | 0x4480 0A54 | 0x4480 0B54 |
| L3_TARG_STDERRLOG_INFO | R | 32 | 0x0000 0058 | 0x4480 0858 | 0x4480 0958 | 0x4480 0A58 | 0x4480 0B58 |
| L3_TARG_STDERRLOG_SLVOFSLSB | R | 32 | 0x0000 005C | 0x4480 085C | 0x4480 095C | 0x4480 0A5C | 0x4480 0B5C |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-212. TARG Registers Summary (continued)**

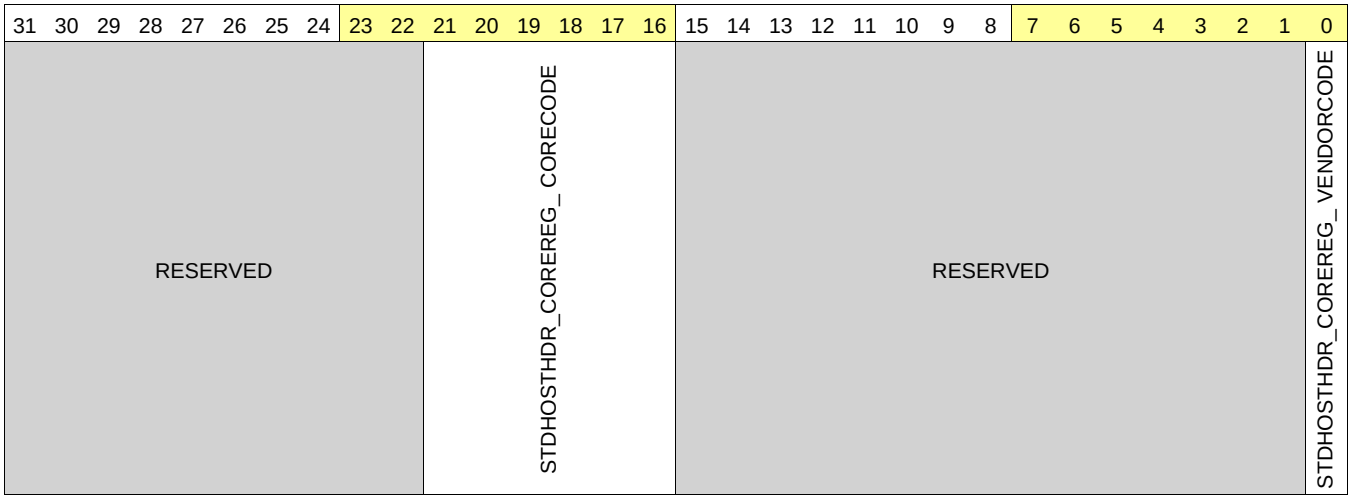
| Register Name | Type | Register Width (Bits) | Address Offset for CLK2_C2C | L3S_MCASP1_TARG L3 Physical Address | L3S_MCASP2_TARG L3 Physical Address | L3S_MCASP3_TARG L3 Physical Address | L3S_USB_TARG L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L3_TARG_STDERRLOG_SLVOFSMSB | R | 32 | 0x0000 0060 | 0x4480 0860 | 0x4480 0960 | 0x4480 0A60 | 0x4480 0B60 |
| L3_TARG_STDERRLOG_CUSTOMINFO_INFO | R | 32 | 0x0000 0064 | 0x4480 0864 | 0x4480 0964 | 0x4480 0A64 | 0x4480 0B64 |
| L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR | R | 32 | 0x0000 0068 | 0x4480 0868 | 0x4480 0968 | 0x4480 0A68 | 0x4480 0B68 |
| L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE | R | 32 | 0x0000 006C | 0x4480 086C | 0x4480 096C | 0x4480 0A6C | 0x4480 0B6C |
| L3_TARG_ADDRSPACESIZELOG | RW | 32 | 0x0000 0080 | 0x4480 0880 | 0x4480 0980 | 0x4480 0A80 | 0x4480 0B80 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.2.5.1.2.2 L3 TARG Registers Description

**Table 1-213. L3_TARG_STDHOSTHDR_COREREG**

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0000 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |

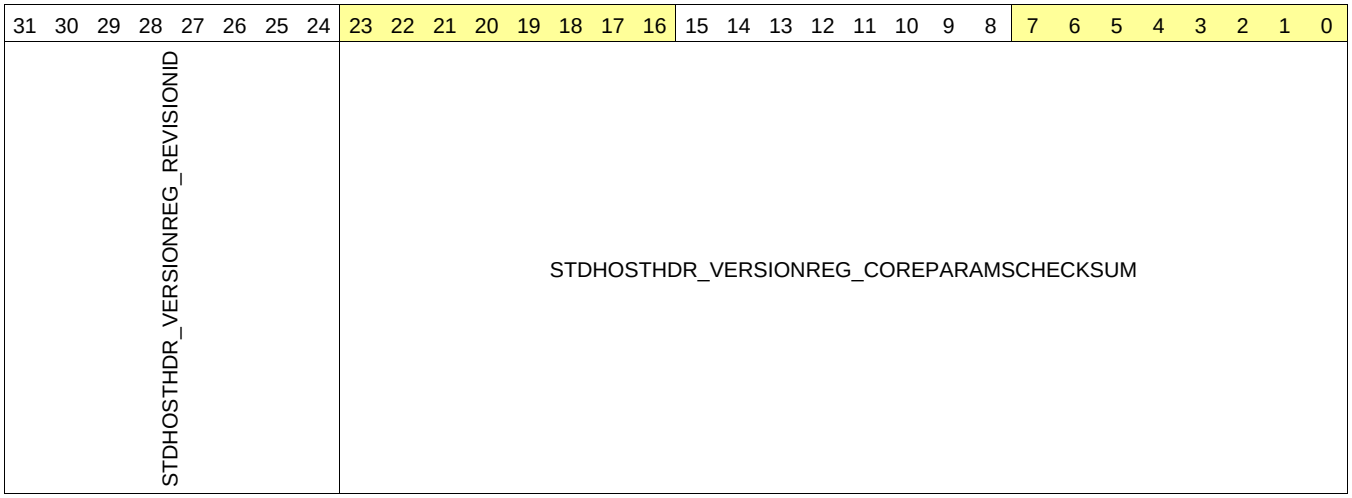



| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:22 | RESERVED | | R | 0x000 |
| 21:16 | STDHOSTHDR_COREREG_ CORECODE | The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x13. | R | 0x13 |
| 15:1 | RESERVED | | R | 0x0000 |
| 0 | STDHOSTHDR_COREREG_ VENDORCODE | The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1.<br>Read 0x0: Third-party vendor.<br>Read 0x1: | R | 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-214. L3_TARG_STDHOSTHDR_VERSIONREG

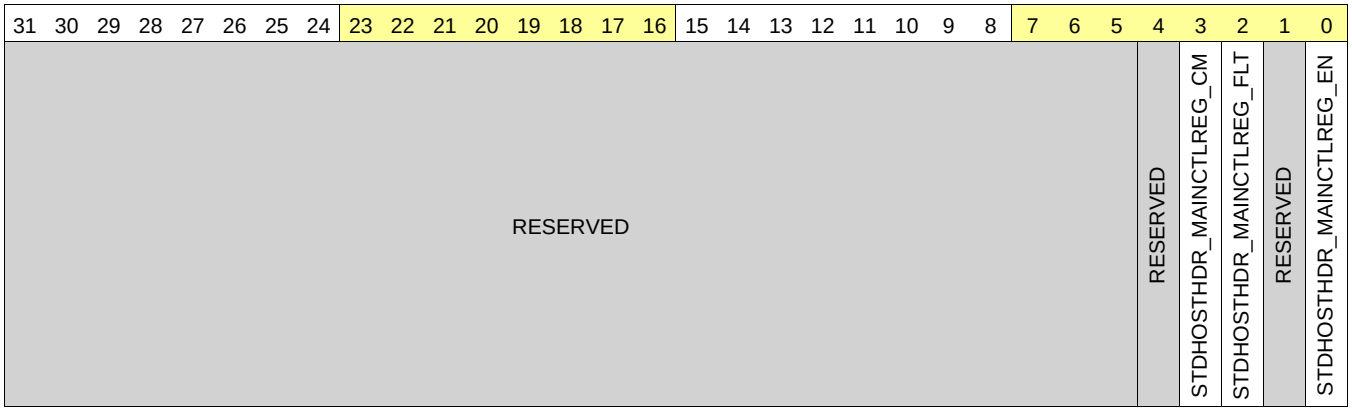
| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0004 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:24 | STDHOSTHDR_VERSIONREG_REVISIONID | The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0. | R | 0x00 |
| 23:0 | STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM | Reserved. Type: Reserved. Reset value: Reserved. | R | 0x000000 |

### Table 1-215. L3_TARG_STDHOSTHDR_MAINCTLREG

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0008 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | RW | | |



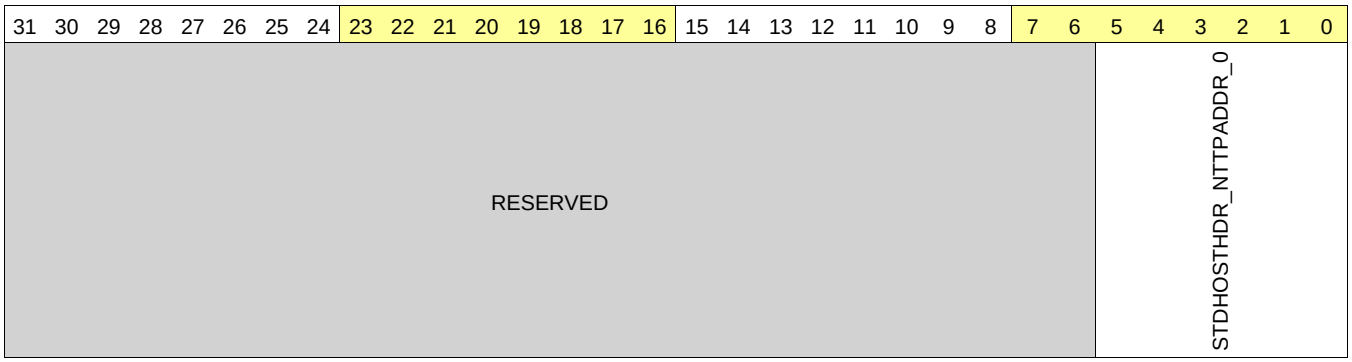


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:5 | RESERVED | Reserved | R | 0x0000000 |
| 4 | RESERVED | Reserved | R | 0 |
| 3 | STDHOSTHDR_MAINCTLREG_CM | Reserved for internal testing. Must be set to 0. Type: Control. Reset value: 0x0. | R | 0 |
| 2 | STDHOSTHDR_MAINCTLREG_FLT | Asserted when a Fault condition is detected: if the unit includes Error Logging, Flt is asserted when the FltCnt register field indicates a Fault, and deasserted when FltCnt is reset. If no Error Logging is implemented, this bit becomes unit-dependent. In all cases, Flt bit and Flt pin (service network) have the same logical level. Type: Status. Reset value: X. | R | 0 |
| 1 | RESERVED | Reserved | R | 0 |
| 0 | STDHOSTHDR_MAINCTLREG_EN | Sets the global core enable. Note: A disabled master does not generate any NTTP requests, and a disabled slave replies with an error packet to any request it receives. Type: Control. Reset value: 0x1. | RW | 1 |

## Table 1-216. L3_TARG_STDHOSTHDR_NTTPADDR_0

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0010 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |

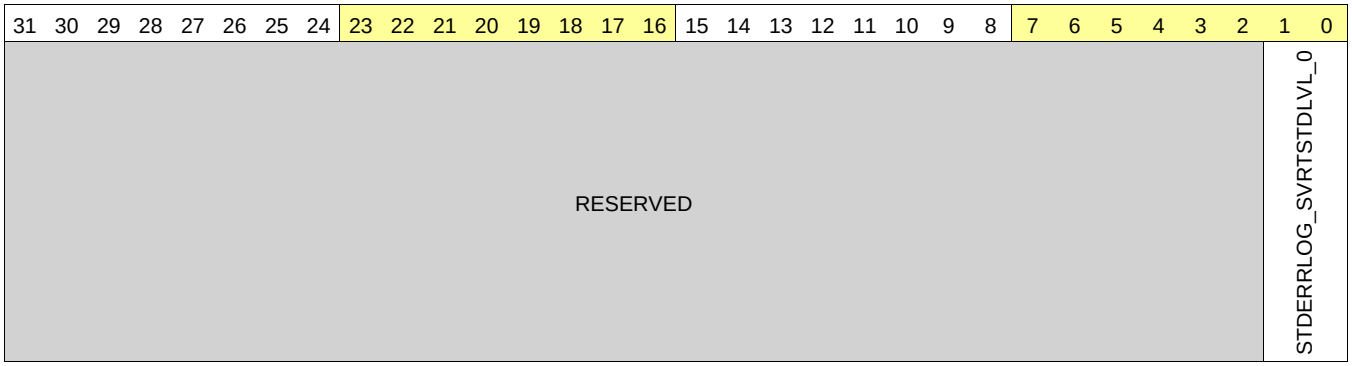



| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | RESERVED | Reserved | R | 0x0000000 |
| 5:0 | STDHOSTHDR_NTTPADDR_0 | Sets the RX port address. Type: Control. Reset value: 0x15. | R | 0x03 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-217. L3_TARG_STDERRLOG_SVRTSTDLVL

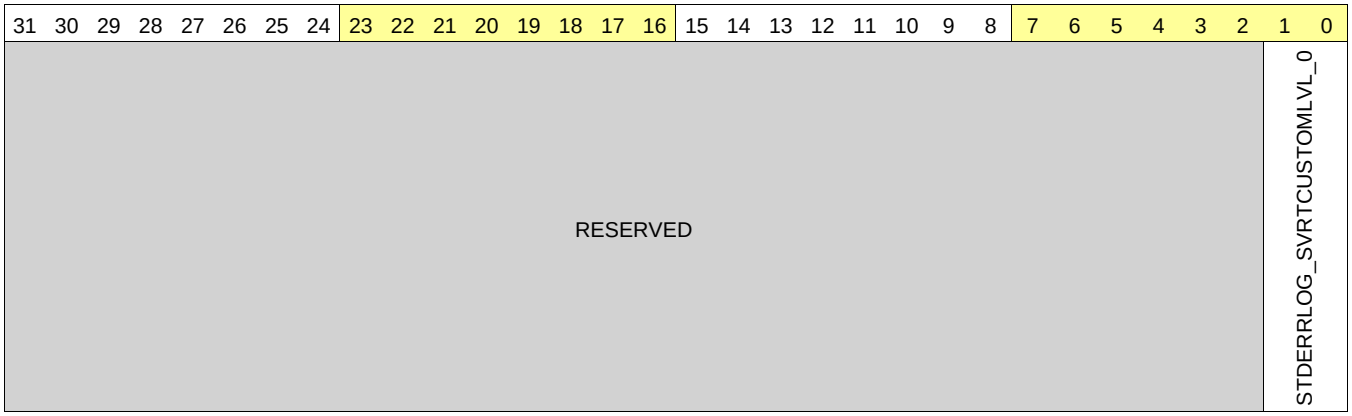
| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0040 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | RW | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | RESERVED | Reserved | R | 0x0000 0000 |
| 1:0 | STDERRLOG_SVRTSTDLVL_0 | Severity level parameters Type: Control. Reset value: 0x2.<br>0x0: Error logging is disabled.<br>0x1: Errors are logged with severity level Error.<br>0x2: Errors are logged with severity level Fault. | RW | 0x2 |

### Table 1-218. L3_TARG_STDERRLOG_SVRTCUSTOMLVL

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0044 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | RW | | |





Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | RESERVED | Reserved | R | 0x0000 0000 |
| 1:0 | STDERRLOG_SVRTCUSTOMLVL_0 | Severity level parameters Type: Control. Reset value: 0x2.<br>0x0: Error logging is disabled.<br>0x1: Errors are logged with severity level Error.<br>0x2: Errors are logged with severity level Fault. | RW | 0x2 |

## Table 1-219. L3_TARG_STDERRLOG_MAIN

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 048 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | RW | | |

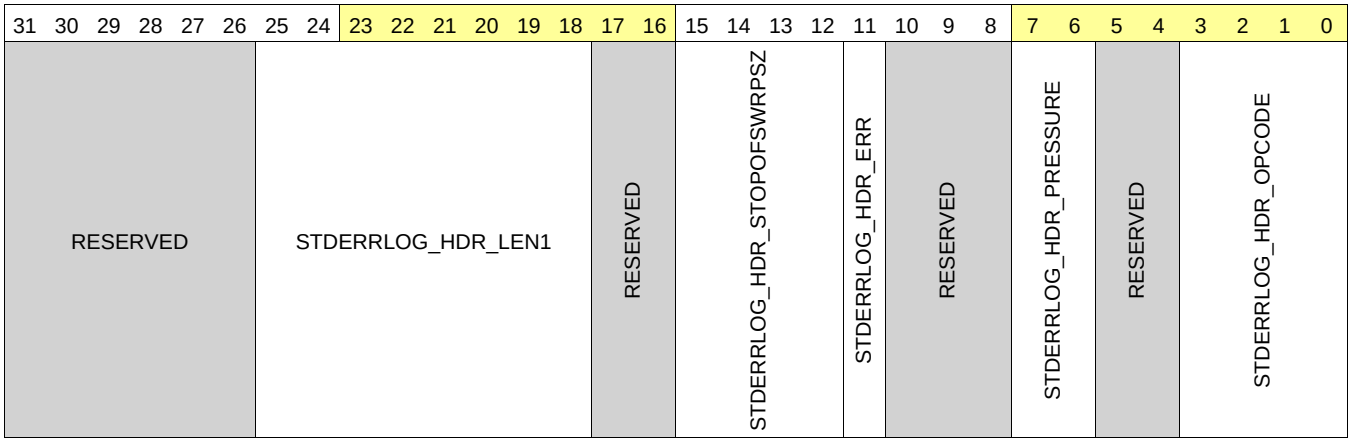
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STDERRLOG_MAIN_CLRLOG | RESERVED | | | | | | | | | | | STDERRLOG_MAIN_FLTCNT | STDERRLOG_MAIN_ERRCNT | RESERVED | | | | | | | | | | | | | | | | | STDERRLOG_MAIN_ERRTYPE | STDERRLOG_MAIN_ERRLOGVLD |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31 | STDERRLOG_MAIN_CLRLOG | Clears Error Logging Valid bit when written to 1. Type: Give_AutoCleared. Reset value: 0x0. | RW | 0 |
| 30:20 | RESERVED | Reserved | R | 0x000 |
| 19 | STDERRLOG_MAIN_FLTCNT | Asserted when at least one error with severity level FAULT is detected. Reset when written to 1. Type: Take. Reset value: 0x0. | RW | 0 |
| 18 | STDERRLOG_MAIN_ERRCNT | Asserted when at least one error with severity level ERROR is detected. Reset when written to 1. Type: Take. Reset value: 0x0. | RW | 0 |
| 17:2 | RESERVED | Reserved | R | 0x0000 |
| 1 | STDERRLOG_MAIN_ERRTYPE | Indicates logging type. Type: Status. Reset value: X.<br>Read 0x0: Logged Error format is standard (header and necker recorded).<br>Read 0x1: Logged Error format is module dependent. CustomInfo register(s) may be implemented to store additional information. | R | 0 |
| 0 | STDERRLOG_MAIN_ERRLOGVLD | Error Logging Valid. Asserted when logging information is valid(indicates that an error has been logged). Type: Status. Reset value: X. | R | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

## Table 1-220. L3_TARG_STDERRLOG_HDR

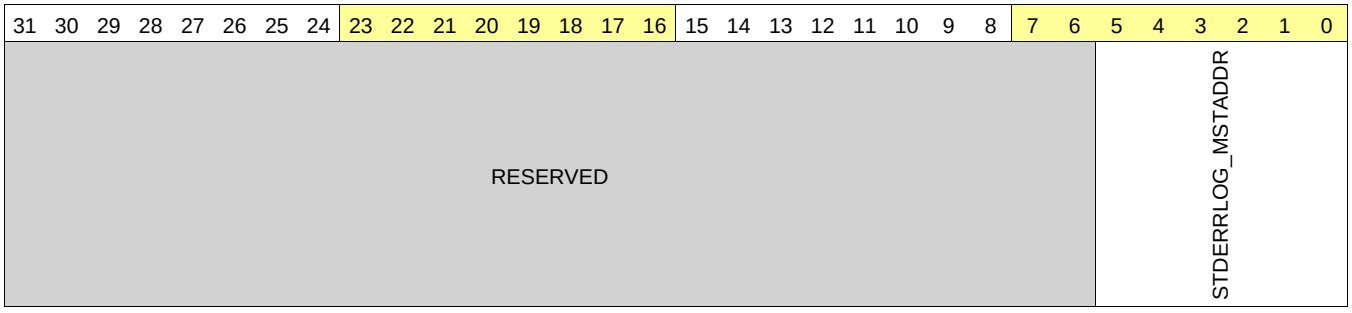
| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 004C | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:26 | RESERVED | Reserved | R | 0x00 |
| 25:18 | STDERRLOG_HDR_LEN1 | This field contains the number of payload cell(s) minus one of the logged packet. Type: Status. Reset value: X. | R | 0x00 |
| 17:16 | RESERVED | Reserved | R | 0x0 |
| 15:12 | STDERRLOG_HDR_STOPOFSWRPSZ | StopOfs or WrapSize field of the logged packet (meaning depends on Wrp bit of logged opcode). Type: Status. Reset value: X. | R | 0x0 |
| 11 | STDERRLOG_HDR_ERR | Err bit of the logged packet. Type: Status. Reset value: X. | R | 0 |
| 10:8 | RESERVED | Reserved | R | 0x0 |
| 7:6 | STDERRLOG_HDR_PRESSURE | Pressure field of the logged packet. Type: Status. Reset value: X. | R | 0 |
| 5:4 | RESERVED | Reserved | R | 0x0 |
| 3:0 | STDERRLOG_HDR_OPCODE | Opcode of the logged packet. Type: Status. Reset value: X. | R | 0x0 |

## Table 1-221. L3_TARG_STDERRLOG_MSTADDR

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0050 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |





Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | RESERVED | Reserved | R | 0x000000 |
| 5:0 | STDERRLOG_MSTADDR | Master Address field of the logged packet . Type: Status. Reset value: X. | R | 0x00 |

## Table 1-222. L3_TARG_STDERRLOG_SLVADDR

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0054 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STDERRLOG_SLVADDR | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | RESERVED | Reserved | R | 0x0000000 |
| 5:0 | STDERRLOG_SLVADDR | Slave Address field of the logged packet. Type: Status. Reset value: X. | R | 0x00 |

## Table 1-223. L3_TARG_STDERRLOG_INFO

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0058 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STDERRLOG_INFO | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:8 | RESERVED | Reserved | R | 0x000000 |
| 7:0 | STDERRLOG_INFO | Info field of the logged packet. Type: Status. Reset value: X. | R | 0x00 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-224. L3_TARG_STDERRLOG_SLVOFSLSB

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 005C | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STDERRLOG_SLVOFSLSB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | STDERRLOG_SLVOFSLSB | LSB of the slave offset field, concatenated with start offset field of the logged packet. Type: Status. Reset value: X. | R | 0x0000 0000 |

### Table 1-225. L3_TARG_STDERRLOG_SLVOFSMSB

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0060 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |

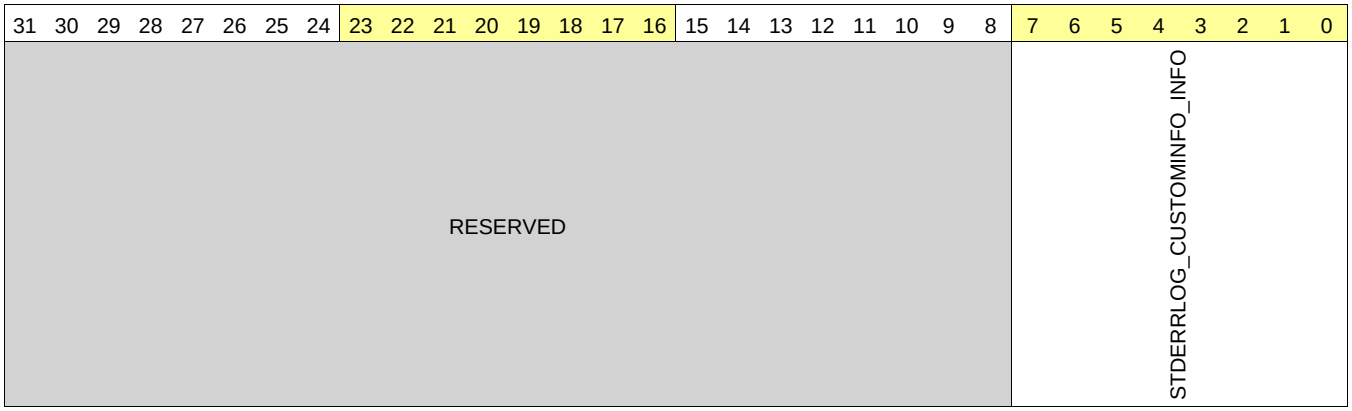
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | STDERRLOG_SLVOFSMSB |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | STDERRLOG_SLVOFSMSB | MSB of the slave offset field of the logged packet (according to NTTP packet format, this register field may exceed the actual slave offset size. Unused bits are stuck at 0, if any). Type: Status. Reset value: X. | R | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-226. L3_TARG_STDERRLOG_CUSTOMINFO_INFO

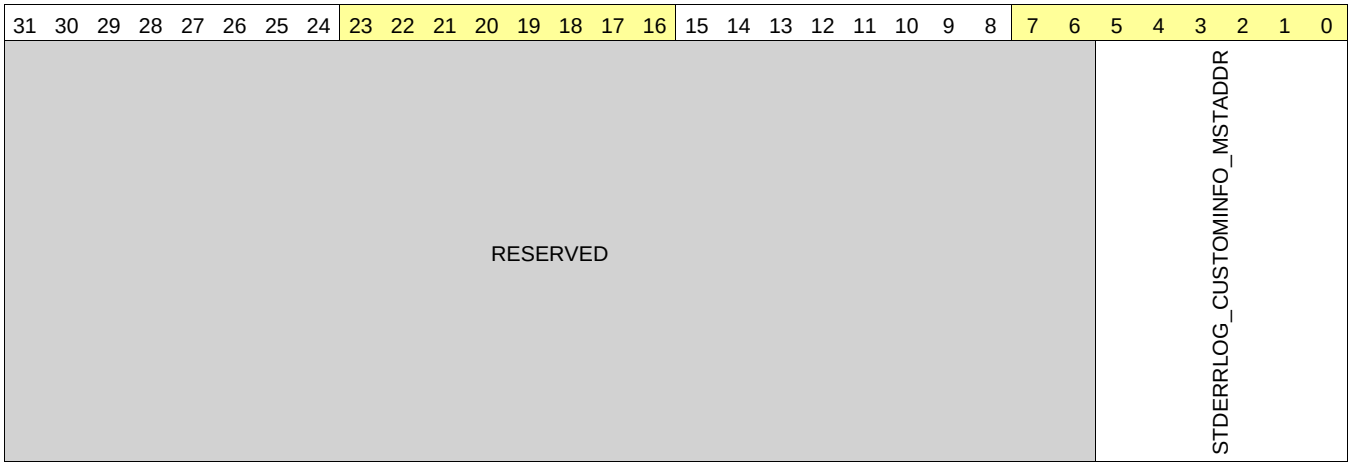
| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0064 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:8 | RESERVED | Reserved | R | 0x000000 |
| 7:0 | STDERRLOG_CUSTOMINFO_INFO | Info field of the response packet. Type: Status. Reset value: X. | R | 0x00 |

### Table 1-227. L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0068 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |

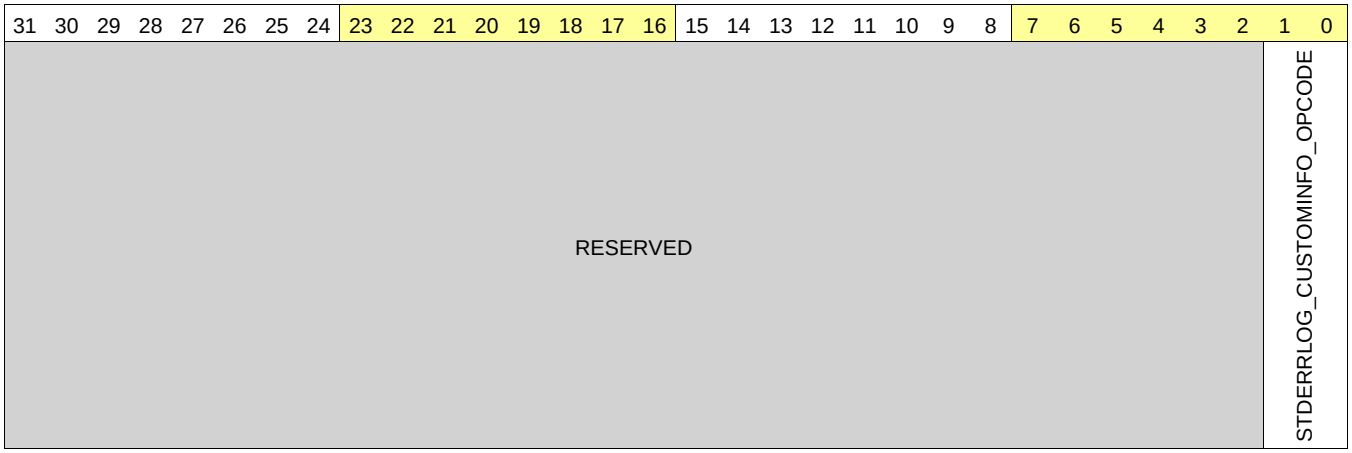



| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | RESERVED | Reserved | R | 0x000000 |
| 5:0 | STDERRLOG_CUSTOMINFO_MSTADDR | MstAddr field of the response packet (see ). Type: Status. Reset value: X. | R | 0x00 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-228. L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE

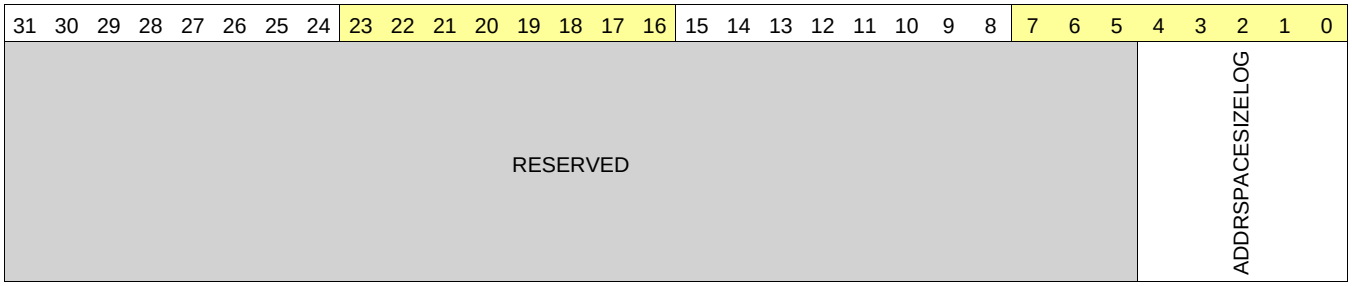
| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 006C | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | RESERVED | Reserved | R | 0x0000 0000 |
| 1:0 | STDERRLOG_CUSTOMINFO_OPCODE | Opcode of the response packet. Type: Status. Reset value: X. | R | 0x0 |

### Table 1-229. L3_TARG_ADDRSPACESIZELOG

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0080 | | |
| **Physical Address** | See Table 1-206 through Table 1-212 | **Instance** | See Table 1-206 through Table 1-212 |
| **Description** | | | |
| **Type** | RW | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:5 | RESERVED | Reserved | R | 0x0000000 |
| 4:0 | ADDRSPACESIZELOG | The address space size is equal to 2**AddrSpaceSizeLog * 4K in bytes. Type: Control. Reset value: 0x1F. | RW | 0x1F |

#### *1.12.2.5.1.3 L3 FLAGMUX Register Summary and Description*


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-230. FLAGMUX Instance Summary**

| Module Name | Base Address | Size |
|---|---|---|
| L3F_FLAGMUX_CLK1 | 0x4400 0E00 | 4KiB |
| L3F_FLAGMUX_CLK2 | 0x4440 0E00 | 4KiB |
| L3S_FLAGMUX | 0x4480 0C00 | 4KiB |
| FLAGMUX_TOP | 0x4400 0F00 | 4KiB |

#### *1.12.2.5.1.3.1 L3 FLAGMUX Register Summary*

**Table 1-231. FLAGMUX Registers Summary**

| Register Name | Type | Register Width (Bits) | Address Offset for L3F_CLK1 | L3F_FLAGMUX_CLK1 L3 Physical Address | Address Offset for L3F_CLK2 | L3F_FLAGMUX_CLK2 L3 Physical Address |
|---|---|---|---|---|---|---|
| L3_FLAGMUX_STDHOSTHDR_CORREREG | R | 32 | 0x0000 0E00 | 0x4400 0E00 | 0x0040 0E00 | 0x4440 0E00 |
| L3_FLAGMUX_STDHOSTHDR_VERSIONREG | R | 32 | 0x0000 0E04 | 0x4400 0E04 | 0x0040 0E04 | 0x4440 0E04 |
| L3_FLAGMUX_MASK0 | RW | 32 | 0x0000 0E08 | 0x4400 0E08 | 0x0040 0E08 | 0x4440 0E08 |
| L3_FLAGMUX_REGERR0 | R | 32 | 0x0000 0E0C | 0x4400 0E0C | 0x0040 0E0C | 0x4440 0E0C |
| L3_FLAGMUX_MASK1 | RW | 32 | 0x0000 0E10 | 0x4400 0E10 | 0x0040 0E10 | 0x4440 0E10 |
| L3_FLAGMUX_REGERR1 | RW | 32 | 0x0000 0E14 | 0x4400 0E14 | 0x0040 0E14 | 0x4440 0E14 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-232. FLAGMUX Registers Summary**

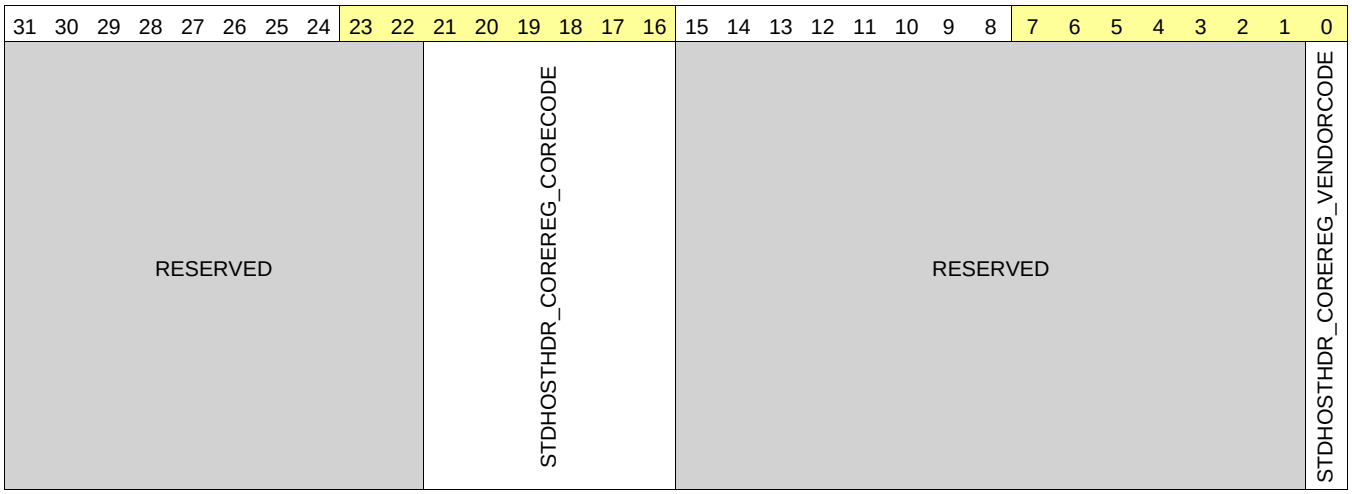
| Register Name | Type | Register Width (Bits) | Address Offset for L3S | CLK2L3S L3 Physical Address | Address Offset for FLAGMUX_TOP | FLAGMUX_TOP L3 Physical Address |
|---|---|---|---|---|---|---|
| L3_FLAGMUX_STDHOSTHDR_COREREG | R | 32 | 0x0080 1000 | 0x4480 0C00 | 0x0000 0F00 | 0x4400 0F00 |
| L3_FLAGMUX_STDHOSTHDR_VERSIONREG | R | 32 | 0x0080 1004 | 0x4480 0C04 | 0x0000 0F04 | 0x4400 0F04 |
| L3_FLAGMUX_MASK0 | RW | 32 | 0x0080 1008 | 0x4480 0C08 | 0x0000 0F08 | 0x4400 0F08 |
| L3_FLAGMUX_REGERR0 | R | 32 | 0x0080 100C | 0x4480 0C0C | 0x0000 0F10 | 0x4400 0F0C |
| L3_FLAGMUX_MASK1 | RW | 32 | 0x0080 1010 | 0x4480 0C10 | 0x0000 0F40 | 0x4400 0F10 |
| L3_FLAGMUX_REGERR1 | RW | 32 | 0x0080 1014 | 0x4480 0C14 | 0x0000 0F44 | 0x4400 0F14 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.2.5.1.3.2 L3 FLAGMUX Register Description

**Table 1-233. L3_FLAGMUX_STDHOSTHDR_COREREG**

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-231. | | |
| **Physical Address** | See Table 1-231 and Table 1-232. | **Instance** | See Table 1-231 and Table 1-232. |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:22 | RESERVED | Reserved | R | 0x000 |
| 21:16 | STDHOSTHDR_COREREG_CORECODE | The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37. | R | 0x37 |
| 15:1 | RESERVED | Reserved | R | 0x0000 |
| 0 | STDHOSTHDR_COREREG_VENDORCODE | The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1.<br>Read 0x0: Third-party vendor.<br>Read 0x1: | R | 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-234. L3_FLAGMUX_STDHOSTHDR_VERSIONREG

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-231. | | |
| **Physical Address** | See Table 1-231 and Table 1-232. | **Instance** | See Table 1-231 and Table 1-232. |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STDHOSTHDR_VERSIONREG_REVISIONID | | | | | | | | STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:24 | STDHOSTHDR_VERSIONREG_REVISIONID | The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0. | R | 0x00 |
| 23:0 | STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM | Reserved. Type: Reserved. Reset value: Reserved. | R | 0x000000 |

### Table 1-235. L3_FLAGMUX_MASK0

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-231. | | |
| **Physical Address** | See Table 1-231 and Table 1-232. | **Instance** | See Table 1-231 and Table 1-232. |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | MASK0 | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:11 | RESERVED | Reserved | R | 0x0000 |
| 10:0 | MASK0 | Mask flag inputs 0 Type: Control. Reset value: 0x3FFFF. | RW | see Table 1-239 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-236. L3_FLAGMUX_REGERR0

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-231. | | |
| **Physical Address** | See Table 1-231 and Table 1-232. | **Instance** | See Table 1-231 and Table 1-232. |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | REGERR0 | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:11 | RESERVED | Reserved | R | 0x0000 |
| 10:0 | REGERR0 | Flag inputs 0 Type: Status. Reset value: X. | R | 0x0 |

### Table 1-237. L3_FLAGMUX_MASK1

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-231. | | |
| **Physical Address** | See Table 1-231 and Table 1-232. | **Instance** | See Table 1-231 and Table 1-232. |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | MASK1 | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:11 | RESERVED | Reserved | R | 0x0000 |
| 10:0 | MASK1 | Mask flag inputs 1 Type: Control. Reset value: 0x3FFFF. | RW | see Table 1-239 |

### Table 1-238. L3_FLAGMUX_REGERR1

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-231. | | |
| **Physical Address** | See Table 1-231 and Table 1-232. | **Instance** | See Table 1-231 and Table 1-232. |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | REGERR1 | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:11 | RESERVED | Reserved | R | 0x0000 |
| 10:0 | REGERR1 | Flag inputs 1 Type: Status. Reset value: X. | R | 0x0 |

### Table 1-239. Reset Values for FLAGMUX_MASKi

| Instance | FLAGMUX_MASKi/REGERRi [] Bit Field Size | Reset value of FLAGMUX_MASKi[]MASKi |
|---|---|---|
| L3F_CLK1 | [10:0] | 0x7FF |
| L3F_CLK2 | [12:0] | 0x1FFF |
| L3S | [9:0] | 0x3FF |
| FLAGMUX_TOP | [2:0] | 0x1F |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.12.2.5.1.4 L3 BW Regulator Register Summary and Description

**Table 1-240. BW_REGULATOR Instance Summary**

| Module Name | Base Address | Size |
|---|---|---|
| L3F_TPTC_RD4_BW_REGULATOR | 0x4400 1F00 | 4KiB |
| L3F_TPTC_WR4_BW_REGULATOR | 0x4400 2000 | 4KiB |
| L3F_TPTC_RD2_BW_REGULATOR | 0x4400 2100 | 4KiB |
| L3F_TPTC_WR2_BW_REGULATOR | 0x4400 2200 | 4KiB |
| L3F_MMU_BW_REGULATOR | 0x4400 2300 | 4KiB |
| L3M_HDVICP_BW_REGULATOR | 0x4440 1C00 | 4KiB |
| L3M_SGX_BW_REGULATOR | 0x4440 2400 | 4KiB |
| L3M_ISS_BW_REGULATOR | 0x4440 2500 | 4KiB |

##### 1.12.2.5.1.4.1 L3 BW Regulator Register Summary


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-241. BW_REGULATOR Registers Summary**

| Register Name | Type | Register Width (Bits) | Address OffsetLATOR | L3F_TPTC_RD4_BW_REGULATOR L3 Physical Address | L3F_TPTC_WR4_BW_REGULATOR L3 Physical AddresATOR | L3F_TPTC_RD2_BW_REGULATOR L3 Physical Addres | L3F_TPTC_WR2_BW_REGULATOR L3 Physical Addres | L3F_MMU_BW_REGULATOR L3 Physical Addres |
|---|---|---|---|---|---|---|---|---|
| L3_BW_R_STDHOSTHDR_COREREG | R | 32 | 0x0000 0000 | 0x4400 1F00 | 0x4400 2000 | 0x4400 2100 | 0x4400 2200 | 0x4400 2300 |
| L3_BW_R_STDHOSTHDR_VERSIONREG | R | 32 | 0x0000 0004 | 0x4400 1F04 | 0x4400 2004 | 0x4400 2104 | 0x4400 2204 | 0x4400 2304 |
| L3_BW_R_BANDWIDTH | RW | 32 | 0x0000 0008 | 0x4400 1F08 | 0x4400 2008 | 0x4400 2108 | 0x4400 2208 | 0x4400 2308 |
| L3_BW_R_WATERMARK | RW | 32 | 0x0000 000C | 0x4400 1F0C | 0x4400 200C | 0x4400 210C | 0x4400 220C | 0x4400 230C |
| L3_BW_R_PRESS | R | 32 | 0x0000 0010 | 0x4400 1F10 | 0x4400 2010 | 0x4400 2110 | 0x4400 2210 | 0x4400 2310 |
| L3_BW_R_CLEARHISTORY | RW | 32 | 0x0000 0014 | 0x4400 1F14 | 0x4400 2014 | 0x4400 2114 | 0x4400 2214 | 0x4400 2314 |

**Table 1-242. BW_REGULATOR Registers Summary**

| Register Name | Type | Register Width (Bits) | Address Offset | L3M_HDVICP_BW_REGULATOR L3 Physical Address | L3M_SGX_BW_REGULATOR L3 Physical Address | L3M_ISS_BW_REGULATOR L3 Physical Address |
|---|---|---|---|---|---|---|
| L3_BW_R_STDHOSTHDR_COREREG | R | 32 | 0x0040 0000 | 0x4440 1C00 | 0x4440 2400 | 0x4440 2500 |
| L3_BW_R_STDHOSTHDR_VERSIONREG | R | 32 | 0x0040 0004 | 0x4440 1C04 | 0x4440 2404 | 0x4440 2504 |
| L3_BW_R_BANDWIDTH | RW | 32 | 0x0040 0008 | 0x4440 1C08 | 0x4440 2408 | 0x4440 2508 |
| L3_BW_R_WATERMARK | RW | 32 | 0x0040 000C | 0x4440 1C0C | 0x4440 240C | 0x4440 250C |
| L3_BW_R_PRESS | R | 32 | 0x0040 0010 | 0x4440 1C10 | 0x4440 2410 | 0x4440 2510 |
| L3_BW_R_CLEARHISTORY | RW | 32 | 0x0040 0014 | 0x4440 1C14 | 0x4440 2414 | 0x4440 2514 |

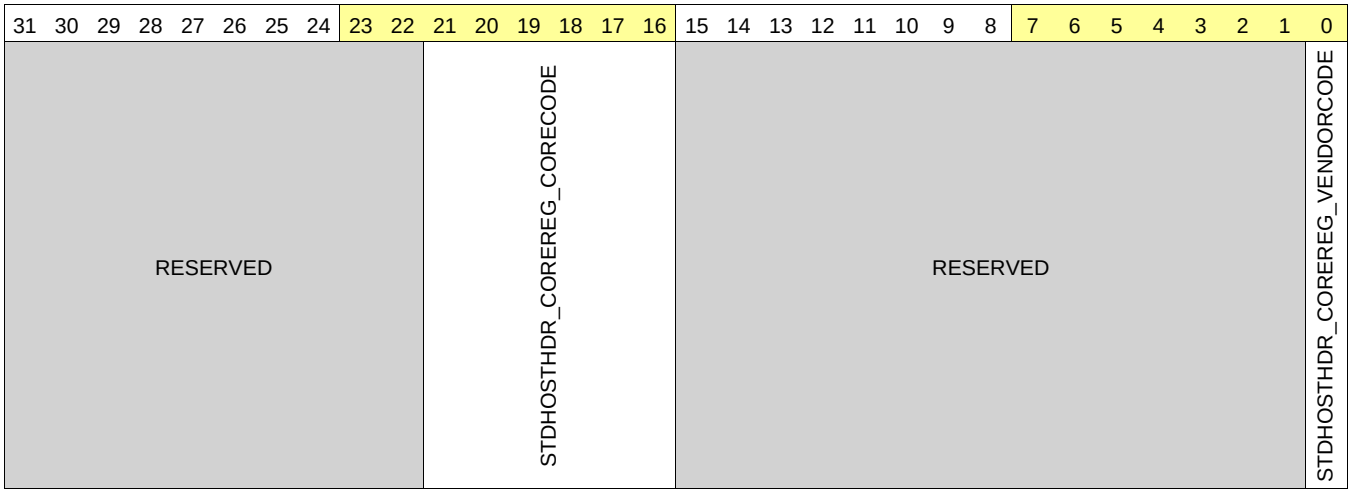
#### *1.12.2.5.1.4.2 L3 BW Regulator Register Description*



Copyright © 2011–2016, Texas Instruments Incorporated

## Table 1-243. L3_BW_R_STDHOSTHDR_COREREG

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-241. | | |
| **Physical Address** | See Table 1-241 andTable 1-242. | **Instance** | See Table 1-241 andTable 1-242. |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:22 | RESERVED | Reserved | R | 0x000 |
| 21:16 | STDHOSTHDR_COREREG_CORECODE | The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x31. | R | 0x31 |
| 15:1 | RESERVED | Reserved | R | 0x0000 |
| 0 | STDHOSTHDR_COREREG_VENDORCODE | The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1.<br>Read 0x0: Third-party vendor.<br>Read 0x1: | R | 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-244. L3_BW_R_STDHOSTHDR_VERSIONREG

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-241. | | |
| **Physical Address** | See Table 1-241 andTable 1-242. | **Instance** | See Table 1-241 andTable 1-242. |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STDHOSTHDR_VERSIONREG_REVISIONID | | | | | | | | STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:24 | STDHOSTHDR_VERSIONREG_REVISIONID | The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0. | R | 0x00 |
| 23:0 | STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM | Reserved. Type: Reserved. Reset value: Reserved. | R | 0x000000 |

### Table 1-245. L3_BW_R_BANDWIDTH

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-241. | | |
| **Physical Address** | See Table 1-241 andTable 1-242. | **Instance** | See Table 1-241 andTable 1-242. |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | BANDWIDTH | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:17 | RESERVED | Reserved | R | 0x0000 |
| 16:0 | BANDWIDTH | Bandwidth, in bytes per second. Type: Control. Reset value: 0x0. | RW | 0x0000 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-246. L3_BW_R_WATERMARK

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-241. | | |
| **Physical Address** | See Table 1-241 andTable 1-242. | **Instance** | See Table 1-241 andTable 1-242. |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | WATERMARK | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:12 | RESERVED | Reserved | R | 0x00000 |
| 11:0 | WATERMARK | Peak permissible bandwidth, in bytes. Type: Control. Reset value: 0x1. | RW | 0x001 |

### Table 1-247. L3_BW_R_PRESS

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-241. | | |
| **Physical Address** | See Table 1-241 andTable 1-242. | **Instance** | See Table 1-241 andTable 1-242. |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | PRESS_LOW | | PRESS_HIGH | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:4 | RESERVED | Reserved | R | 0x0000 0000 |
| 3:2 | PRESS_LOW | Pressure value inserted if the measured bandwidth is over the watermark. The pressure is bar graph encoded. Type: Control. Reset value: 0x0. | R | 0 |
| 1:0 | PRESS_HIGH | Pressure value inserted if the measured bandwidth is under the watermark. The pressure is bar graph encoded. Type: Control. Reset value: 0x1. | R | 0x3 |

### Table 1-248. L3_BW_R_CLEARHISTORY

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-241. | | |
| **Physical Address** | See Table 1-241 andTable 1-242. | **Instance** | See Table 1-241 andTable 1-242. |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | CLEARHISTORY |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | CLEARHISTORY | Write a 1 clear the traffic counter Type: Give_AutoCleared. Reset value: 0x0. | RW | 0 |

#### 1.12.2.5.1.5 L3 STATCOLL Register Summary and Description

**Table 1-249. STATCOLL Instance Summary**

| Module Name | Base Address | Size |
|---|---|---|
| L3F_STATCOLL0 | 0x4400 1000 | 512 bytes |
| L3F_STATCOLL1 | 0x4400 1400 | 512 bytes |
| L3F_STATCOLL2 | 0x4400 1800 | 512 bytes |
| L3M_STATCOLL3 | 0x4440 1000 | 512 bytes |
| L3S_STATCOLL4 | 0x4480 0E00 | 512 bytes |

##### 1.12.2.5.1.5.1 L3 STATCOLL Register Summary


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-250. STATCOLL Register Summary**

| Register Name | Type | Register Width (Bits) | Address Offset for L3F_STATCOLL0 | L3F_STATCOLL0 L3 Physical Address | Address Offset for L3F_STATCOLL1 | L3F_STATCOLL1 L3 Physical Address | Address Offset for L3_STATCOLL2 | L3_STATCOLL2 L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_STCOL_STDHOSTHDR_COREREG | R | 32 | 0x0000 1000 | 0x4400 1000 | 0x0000 1400 | 0x4400 1400 | 0x0000 1800 | 0x4400 1800 |
| L3_STCOL_STDHOSTHDR_VERSIONREG | R | 32 | 0x0000 1004 | 0x4400 1004 | 0x0000 1404 | 0x4400 1404 | 0x0000 1804 | 0x4400 1804 |
| L3_STCOL_EN | RW | 32 | 0x0000 1008 | 0x4400 1008 | 0x0000 1408 | 0x4400 1408 | 0x0000 1808 | 0x4400 1808 |
| L3_STCOL_SOFTEN | RW | 32 | 0x0000 100C | 0x4400 100C | 0x0000 140C | 0x4400 140C | 0x0000 180C | 0x4400 180C |
| L3_STCOL_TRIGEN | RW | 32 | 0x0000 1010 | 0x4400 1010 | 0x0000 1410 | 0x4400 1410 | 0x0000 1810 | 0x4400 1810 |
| L3_STCOL_REQEVT | RW | 32 | 0x0000 1014 | 0x4400 1014 | 0x0000 1414 | 0x4400 1414 | 0x0000 1814 | 0x4400 1814 |
| L3_STCOL_RSPEVT | RW | 32 | 0x0000 1018 | 0x4400 1018 | 0x0000 1418 | 0x4400 1418 | 0x0000 1818 | 0x4400 1818 |
| L3_STCOL_EVTMUX_SEL_k | RW | 32 | 0x0000 101C + (0x4*k) | 0x4400 101C + (0x4*k) | 0x0000 141C + (0x4*k) | 0x4400 141C + (0x4*k) | 0x0000 181C + (0x4*k) | 0x4400 181C + (0x4*k) |
| L3_STCOL_DUMP_IDENTIFIER | R | 32 | 0x0000 1024 | 0x4400 1024 | 0x0000 143C | 0x4400 143C | 0x0000 183C | 0x4400 183C |
| L3_STCOL_DUMP_COLLECTTIME | RW | 32 | 0x0000 1028 | 0x4400 1028 | 0x0000 1440 | 0x4400 1440 | 0x0000 1840 | 0x4400 1840 |
| L3_STCOL_DUMP_SLVADDR | R | 32 | 0x0000 102C | 0x4400 102C | 0x0000 1444 | 0x4400 1444 | 0x0000 1844 | 0x4400 1844 |
| L3_STCOL_DUMP_MSTADDR | R | 32 | 0x0000 1030 | 0x4400 1030 | 0x0000 1448 | 0x4400 1448 | 0x0000 1848 | 0x4400 1848 |
| L3_STCOL_DUMP_SLVOFS | RW | 32 | 0x0000 1034 | 0x4400 1034 | 0x0000 144C | 0x4400 144C | 0x0000 184C | 0x4400 184C |
| L3_STCOL_DUMP_MANUAL | RW | 32 | 0x0000 1038 | 0x4400 1038 | 0x0000 1450 | 0x4400 1450 | 0x0000 1850 | 0x4400 1850 |
| L3_STCOL_DUMP_SEND | RW | 32 | 0x0000 103C | 0x4400 103C | 0x0000 1454 | 0x4400 1454 | 0x0000 1854 | 0x4400 1854 |
| L3_STCOL_FILTER_i_GLOBALEN | RW | 32 | 0x0000 1040 + (0x80*i) + (0x34*m) | 0x4400 1040 + (0x80*i) + (0x34*m) | 0x0000 1458 + (0x54*i) | 0x4400 1458 + (0x54*i) | 0x0000 1858 + (0x54*i) | 0x4400 1858 + (0x54*i) |
| L3_STCOL_FILTER_i_EN_m | RW | 32 | 0x0000 1044 + (0x80*i) + (0x34*m) | 0x4400 1044 + (0x80*i) + (0x34*m) | 0x0000 145C + (0x54*i) | 0x4400 145C + (0x54*i) | 0x0000 185C + (0x54*i) | 0x4400 185C + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_MSTADDR | RW | 32 | 0x0000 1048 + (0x80*i) + (0x34*m) | 0x4400 1048 + (0x80*i) + (0x34*m) | 0x0000 1460 + (0x54*i) | 0x4400 1460 + (0x54*i) | 0x0000 1860 + (0x54*i) | 0x4400 1860 + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_REQUSERINFO | RW | 32 | N/A | N/A | 0x0000 1464 + (0x54*i) | 0x4400 1464 + (0x54*i) | 0x0000 1864 + (0x54*i) | 0x4400 1864 + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_RSPUSERINFO | RW | 32 | N/A | N/A | 0x0000 1468 + (0x54*i) | + (0x54*i) | 0x0000 1868 + (0x54*i) | 0x4400 1868 + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_RD | RW | 32 | 0x0000 104C + (0x80*i) + (0x34*m) | 0x4400 104C + (0x80*i) + (0x34*m) | 0x0000 146C + (0x54*i) | 0x4400 146C + (0x54*i) | 0x0000 186C + (0x54*i) | 0x4400 186C + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_WR | RW | 32 | 0x0000 1050 + (0x80*i) + (0x34*m) | 0x4400 1050 + (0x80*i) + (0x34*m) | 0x0000 1470 + (0x54*i) | 0x4400 1470 + (0x54*i) | 0x0000 1870 + (0x54*i) | 0x4400 1870 + (0x54*i) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-250. STATCOLL Register Summary (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset for L3F_STATCOLL0 | L3F_STATCOLL0 L3 Physical Address | Address Offset for L3F_STATCOLL1 | L3F_STATCOLL1 L3 Physical Address | Address Offset for L3_STATCOLL2 | L3_STATCOLL2 L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_STCOL_FILTER_i_MASK_m_ERR | RW | 32 | 0x0000 1054 + (0x80*i) + (0x34*m) | 0x4400 1054 + (0x80*i) + (0x34*m) | 0x0000 1474 + (0x54*i) | 0x4400 1474 + (0x54*i) | 0x0000 1874 + (0x54*i) | 0x4400 1874 + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_USERINFO | RW | 32 | 0x0000 105C + (0x80*i) + (0x34*m) | 0x4400 105C + (0x80*i) + (0x34*m) | N/A | N/A | N/A | N/A |
| L3_STCOL_FILTER_i_MASK_m_SLVADDR | RW | 32 | N/A | N/A | 0x0000 1478 + (0x54*i) | 0x4400 1478 + (0x54*i) | 0x0000 1878 + (0x54*i) | 0x4400 1878 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_MSTADDR | RW | 32 | 0x0000 1060 + (0x80*i) + (0x34*m) | 0x4400 1060 + (0x80*i) + (0x34*m) | 0x0000 147C + (0x54*i) | 0x4400 147C + (0x54*i) | 0x0000 187C + (0x54*i) | 0x4400 187C + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_REQUSERINFO | RW | 32 | N/A | N/A | 0x0000 1480 + (0x54*i) | 0x4400 1480 + (0x54*i) | 0x0000 1880 + (0x54*i) | 0x4400 1880 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_RESPUSERINFO | RW | 32 | N/A | N/A | 0x0000 1484 + (0x54*i) | 0x4400 1484 + (0x54*i) | 0x0000 1884 + (0x54*i) | 0x4400 1884 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_RD | RW | 32 | 0x0000 1064 + (0x80*i) + (0x34*m) | 0x4400 1064 + (0x80*i) + (0x34*m) | 0x0000 1488 + (0x54*i) | 0x4400 1488 + (0x54*i) | 0x0000 1888 + (0x54*i) | 0x4400 1888 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_WR | RW | 32 | 0x0000 1068 + (0x80*i) + (0x34*m) | 0x4400 1068 + (0x80*i) + (0x34*m) | 0x0000 148C + (0x54*i) | 0x4400 148C + (0x54*i) | 0x0000 188C + (0x54*i) | 0x4400 188C + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_ERR | RW | 32 | 0x0000 106C + (0x80*i) + (0x34*m) | 0x4400 106C + (0x80*i) + (0x34*m) | 0x0000 1490 + (0x54*i) | 0x4400 1490 + (0x54*i) | 0x0000 1890 + (0x54*i) | 0x4400 1890 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_SLVADDR | RW | 32 | N/A | N/A | 0x0000 1494 + (0x54*i) | 0x4400 1494 + (0x54*i) | 0x0000 1894 + (0x54*i) | 0x4400 1894 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_USERINFO | RW | 32 | 0x0000 1074 + (0x80*i) + (0x34*m) | 0x4400 1074 + (0x80*i) + (0x34*m) | N/A | N/A | N/A | N/A |
| L3_STCOL_OP_i_THRESHOLD_MINVAL | RW | 32 | 0x0000 10AC + (0x80*i) + (0x34*m) | 0x4400 10AC + (0x80*i) + (0x34*m) | 0x0000 1498 + (0x54*i) | 0x4400 1498 + (0x54*i) | 0x0000 1898 + (0x54*i) | 0x4400 + (0x54*i) |
| L3_STCOL_OP_i_THRESHOLD_MAXVAL | RW | 32 | 0x0000 10B0 + (0x80*i) + (0x34*m) | 0x4400 10B0 + (0x80*i) + (0x34*m) | 0x0000 149C + (0x54*i) | 0x4400 149C + (0x54*i) | 0x0000 189C + (0x54*i) | 0x4400 189C + (0x54*i) |
| L3_STCOL_OP_i_EVTINFOSEL | RW | 32 | 0x0000 10B4 + (0x80*i) + (0x34*m) | 0x4400 10B4 + (0x80*i) + (0x34*m) | 0x0000 14A0 + (0x54*i) | 0x4400 14A0 + (0x54*i) | 0x0000 18A0 + (0x54*i) | 0x4400 18A0 + (0x54*i) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-250. STATCOLL Register Summary (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset for L3F_STATCOLL0 | L3F_STATCOLL0 L3 Physical Address | Address Offset for L3F_STATCOLL1 | L3F_STATCOLL1 L3 Physical Address | Address Offset for L3_STATCOLL2 | L3_STATCOLL2 L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L3_STCOL_OP_i_SEL | RW | 32 | 0x0000 10B8 + (0x80*i) + (0x34*m) | 0x4400 10B8 + (0x80*i) + (0x34*m) | 0x0000 14A4 + (0x54*i) | 0x4400 14A4 + (0x54*i) | 0x0000 18A4 + (0x54*i) | 0x4400 18A4 + (0x54*i) |

**Table 1-251. STATCOLL Register Summary**

| Register Name | Type | Register Width (Bits) | Address Offset for L3M_STATCOLL3 | L3M_STATCOLL3 L3 Physical Address | Address Offset for L3S_STATCOLL4 | L3S_STATCOLL4 L3 Physical Address |
|---|---|---|---|---|---|---|
| L3_STCOL_STDHOSTHDR_COREREG | R | 32 | 0x0040 1000 | 0x4440 1000 | 0x0080 0E00 | 0x4480 0E00 |
| L3_STCOL_STDHOSTHDR_VERSIONREG | R | 32 | 0x0040 1004 | 0x4440 1004 | 0x0080 0E04 | 0x4480 0E04 |
| L3_STCOL_EN | RW | 32 | 0x0040 1008 | 0x4440 1008 | 0x0080 0E08 | 0x4480 0E08 |
| L3_STCOL_SOFTEN | RW | 32 | 0x0040 100C | 0x4440 100C | 0x0080 0E0C | 0x4480 0E0C |
| L3_STCOL_TRIGEN | RW | 32 | 0x0040 1010 | 0x4440 1010 | 0x0080 0E10 | 0x4480 0E10 |
| L3_STCOL_REQEVT | RW | 32 | 0x0040 1014 | 0x4440 1014 | 0x0080 0E14 | 0x4480 0E14 |
| L3_STCOL_RSPEVT | RW | 32 | 0x0040 1018 | 0x4440 1018 | 0x0080 0E18 | 0x4480 0E18 |
| L3_STCOL_EVTMUX_SEL_k | RW | 32 | 0x0040 101C + (0x4*k) | 0x4440 101C + (0x4*k) | 0x0080 0E1C + (0x4*i) | 0x4480 0E1C + (0x4*i) |
| L3_STCOL_DUMP_IDENTIFIER | R | 32 | 0x0040 102C | 0x4440 102C | 0x0080 0E2C | 0x4480 0E2C |
| L3_STCOL_DUMP_COLLECTTIME | RW | 32 | 0x0040 1030 | 0x4440 1030 | 0x0080 0E30 | 0x4480 0E30 |
| L3_STCOL_DUMP_SLVADDR | R | 32 | 0x0040 1034 | 0x4440 1034 | 0x0080 0E34 | 0x4480 0E34 |
| L3_STCOL_DUMP_MSTADDR | R | 32 | 0x0040 1038 | 0x4440 1038 | 0x0080 0E38 | 0x4480 0E38 |
| L3_STCOL_DUMP_SLVOFS | RW | 32 | 0x0040 103C | 0x4440 103C | 0x0080 0E3C | 0x4480 0E3C |
| L3_STCOL_DUMP_MANUAL | RW | 32 | 0x0040 1040 | 0x4440 1040 | 0x0080 0E40 | 0x4480 0E40 |
| L3_STCOL_DUMP_SEND | RW | 32 | 0x0040 1044 | 0x4440 1044 | 0x0080 0E44 | 0x4480 0E44 |
| L3_STCOL_FILTER_i_GLOBALEN | RW | 32 | 0x0040 1048 + (0x54*i) | 0x4440 1048 + (0x54*i) | 0x0080 0E48 + (0x54*i) | 0x4480 0E48 + (0x54*i) |
| L3_STCOL_FILTER_i_EN_m | RW | 32 | 0x0040 104C + (0x54*i) | 0x4440 104C + (0x54*i) | 0x0080 0E4C + (0x54*i) | 0x4480 0E4C + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_MSTADDR | RW | 32 | 0x0040 1050 + (0x54*i) | 0x4440 1050 + (0x54*i) | 0x0080 0E50 + (0x54*i) | 0x4480 0E50 + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_REQUSERINFO | RW | 32 | 0x0040 1054 + (0x54*i) | 0x4440 1054 +(0x54*i) | 0x0080 0E54 + (0x54*i) | 0x4480 0E54 + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_RSPUSERINFO | RW | 32 | 0x0040 1058 + (0x54*i) | 0x4440 1058 + (0x54*i) | 0x0080 0E58 + (0x54*i) | 0x4480 0E58 + (0x54*i) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-251. STATCOLL Register Summary (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset for L3M_STATCOLL3 | L3M_STATCOLL3 L3 Physical Address | Address Offset for L3S_STATCOLL4 | L3S_STATCOLL4 L3 Physical Address |
|---|---|---|---|---|---|---|
| L3_STCOL_FILTER_i_MASK_m_RD | RW | 32 | 0x0040 105C + (0x54*i) | 0x4440 105C + (0x54*i) | 0x0080 0E5C + (0x54*i) | 0x4480 0E5C + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_WR | RW | 32 | 0x0040 1060 + (0x54*i) | 0x4440 1060 + (0x54*i) | 0x0080 0E60 + (0x54*i) | 0x4480 0E60 + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_ERR | RW | 32 | 0x0040 1064 + (0x54*i) | 0x4440 1064 + (0x54*i) | 0x0080 0E64 + (0x54*i) | 0x4480 0E64 + (0x54*i) |
| L3_STCOL_FILTER_i_MASK_m_USERINFO | RW | 32 | N/A | NA | N/A | N/A |
| L3_STCOL_FILTER_i_MASK_m_SLVADDR | RW | 32 | 0x0040 1068 + (0x54*i) | 0x4440 1068 + (0x54*i) | 0x0080 0E68 + (0x54*i) | 0x4480 0E68 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_MSTADDR | RW | 32 | 0x0040 106C + (0x54*i) | 0x4440 106C + (0x54*i) | 0x0080 0E6C + (0x54*i) | 0x4480 0E6C + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_REQUSERINFO | RW | 32 | 0x0040 1070 + (0x54*i) | 0x4440 1070 + (0x54*i) | 0x0080 0E70 + (0x54*i) | 0x4480 0E70 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_RESPUSERINFO | RW | 32 | 0x0040 1074 + (0x54*i) | 0x4440 1074 + (0x54*i) | 0x0080 0E74 + (0x54*i) | 0x4480 0E74 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_RD | RW | 32 | 0x0040 1078 + (0x54*i) | 0x4440 1078 + (0x54*i) | 0x0080 0E78 + (0x54*i) | 0x4480 0E78 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_WR | RW | 32 | 0x0040 107C + (0x54*i) | 0x4440 107C + (0x54*i) | 0x0080 0E7C + (0x54*i) | 0x4480 0E7C + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_ERR | RW | 32 | 0x0040 1080 + (0x54*i) | 0x4440 1080 + (0x54*i) | 0x0080 0E80 + (0x54*i) | 0x4480 0E80 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_SLVADDR | RW | 32 | 0x0040 1084 + (0x54*i) | 0x4440 1084 + (0x54*i) | 0x0080 0E84 + (0x54*i) | 0x4480 0E84 + (0x54*i) |
| L3_STCOL_FILTER_i_MATCH_m_USERINFO | RW | 32 | N/A | N/A | N/A | N/A |
| L3_STCOL_OP_i_THRESHOLD_MINVAL | RW | 32 | 0x0040 1088 + (0x54*i) | 0x4440 1088 + (0x54*i) | 0x0080 0E88 + (0x54*i) | 0x4480 0E88 + (0x54*i) |
| L3_STCOL_OP_i_THRESHOLD_MAXVAL | RW | 32 | 0x0040 108C + (0x54*i) | 0x4440 108C + (0x54*i) | 0x0080 0E8C + (0x54*i) | 0x4480 0E8C + (0x54*i) |
| L3_STCOL_OP_i_EVTINFOSEL | RW | 32 | 0x0040 1090 + (0x54*i) | 0x4440 1090 + (0x54*i) | 0x0080 0E90 + (0x54*i) | 0x4480 0E90 + (0x54*i) |
| L3_STCOL_OP_i_SEL | RW | 32 | 0x0040 1094 + (0x54*i) | 0x4440 1094 + (0x54*i) | 0x0080 0E94 + (0x54*i) | 0x4480 0E94 + (0x54*i) |

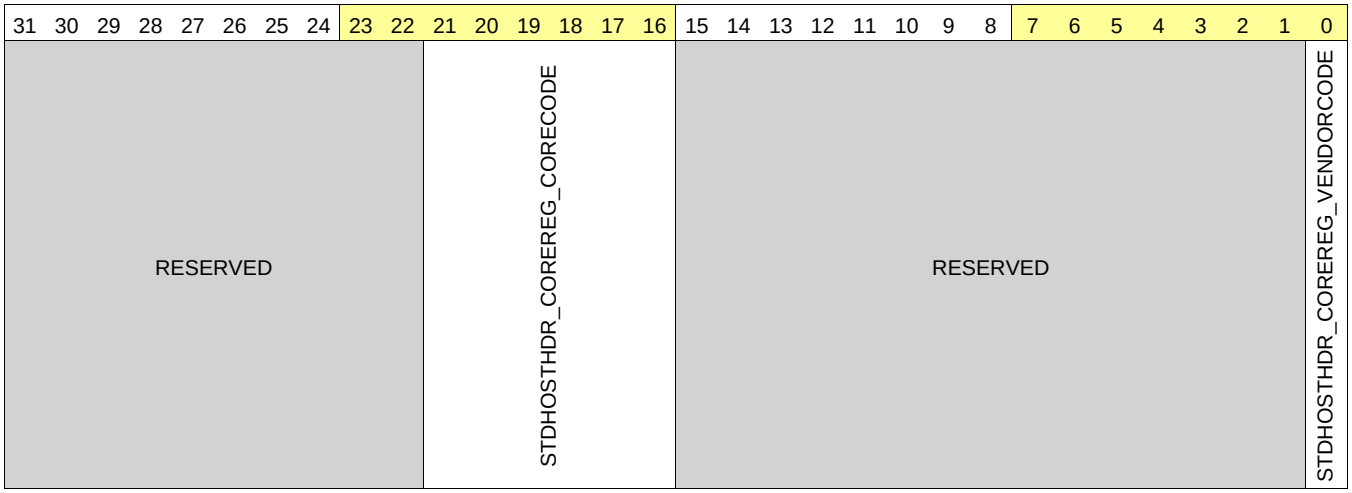
### *1.12.2.5.1.5.2 L3 STATCOLL Register Description*



Copyright © 2011–2016, Texas Instruments Incorporated

## Table 1-252. L3_STCOL_STDHOSTHDR_COREREG

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:22 | RESERVED | Reserved | R | 0x000 |
| 21:16 | STDHOSTHDR_COREREG_CORECODE | The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x3A. (When the instance is CLK3_FLAGMUX_STATCOLL reset value is 0x37) | R | 0x3A |
| 15:1 | RESERVED | Reserved | R | 0x0000 |
| 0 | STDHOSTHDR_COREREG_VENDORCODE | The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1.<br>Read 0x0: Third-party vendor.<br>Read 0x1: | R | 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-253. L3_STCOL_STDHOSTHDR_VERSIONREG

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STDHOSTHDR_VERSIONREG_REVISIONID | | | | | | | | STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:24 | STDHOSTHDR_VERSIONREG_REVISIONID | The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x1. | R | 0x0 |
| 23:0 | STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM | Reserved. Type: Reserved. Reset value: Reserved. | R | 0x000000 |

### Table 1-254. L3_STCOL_EN

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | EN |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | EN | Enable performance monitoring, this will also shut down the clock if En = 0 Type: Control. Reset value: 0x0. | RW | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-255. L3_STCOL_SOFTEN

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SOFTEN |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | SOFTEN | Software enable for performance monitoring Type: Control. Reset value: 0x0. | RW | 0 |

### Table 1-256. L3_STCOL_TRIGEN

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | TRIGEN |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | TRIGEN | TrigEn when set, it enable the external trigger start and stop Type: Control. Reset value: 0x0. | RW | 0 |

### Table 1-257. L3_STCOL_REQEVT

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | REQEVT | | | |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:4 | RESERVED | Reserved | R | 0x0000000 |
| 3:0 | REQEVT | Req event select Type: Control. Reset value: 0x0.<br>0x0: Collect is disabled default value<br>0x1: Collect all event: hit always (cycle)<br>0x2: Collect transfers: actually used cycle for transferring aN NTTP word<br>0x3: Collect wait cycle: transfer has been delayed by source<br>0x4: Collect busy: transfer has been delayed by destination<br>0x5: Collect packet: new packet start<br>0x6: Collect data: data cycle transfer, write for requests, read for responses<br>0x7: Collect idles: transfer is not initiated by source<br>0x8: Collect latency: hit when actually detecting debug bit on response links | RW | 0x0 |

### Table 1-258. L3_STCOL_RSPEVT

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

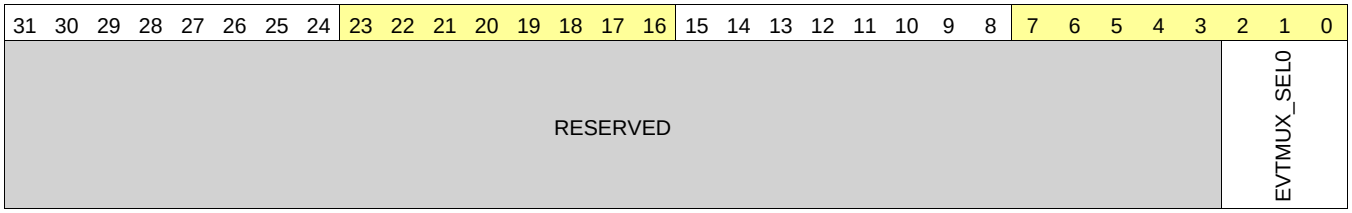
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | RSPEVT | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:4 | RESERVED | Reserved | R | 0x0000000 |
| 3:0 | RSPEVT | Rsp event select Type: Control. Reset value: 0x0.<br>0x0: Collect is disabled default value<br>0x1: Collect all event: hit always (cycle)<br>0x2: Collect transfers: actually used cycle for transferring a NTTP word<br>0x3: Collect wait cycle: transfer has been delayed by source<br>0x4: Collect busy: transfer has been delayed by destination<br>0x5: Collect packet: new packet start<br>0x6: Collect data: data cycle transfer, write for requests, read for responses<br>0x7: Collect idles: transfer is not initiated by source<br>0x8: Collect latency: hit when actually detecting debug bit on response links | RW | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-259. L3_STCOL_EVTMUX_SEL_k

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |



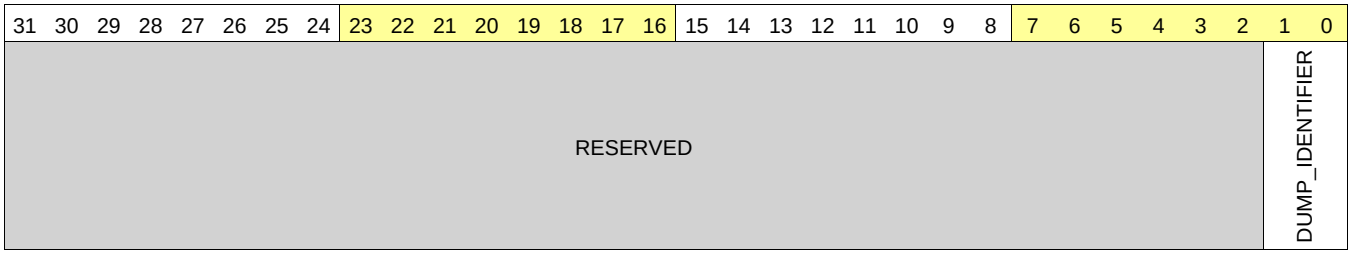

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:3 | RESERVED | Reserved | R | 0x0000 0000 |
| 2:0 | EVTMUX_SEL0Table 1-260 | The select of the mux 0 Type: Control. Reset value: 0x0. | RW | 0x0 |

### Table 1-260. L3_STCOL_EVTMUX_SEL_i[] EVTMUX_SELi Width

| Instance | Bits |
|---|---|
| L3F_STATCOLL0 | [1:0] |
| L3F_STATCOLL1 | [3:0] |
| L3F_STATCOLL2 | [3:0] |
| L3M_STATCOLL3 | [3:0] |
| L3S_STATCOLL4 | [2:0] |

### Table 1-261. L3_STCOL_DUMP_IDENTIFIER

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | R | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | RESERVED | Reserved | R | 0x0000 0000 |
| 1:0 | DUMP_IDENTIFIER | Probe identifier Type: Control. Reset value: 0x0. | R | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-262. L3_STCOL_DUMP_COLLECTTIME

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DUMP_COLLECTTIME | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | DUMP_COLLECTTIME | Number of cycle to wait between two statistics frame Type: Control. Reset value: 0x0. | RW | 0x0000 |

### Table 1-263. L3_STCOL_DUMP_SLVADDR

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | DUMP_SLVADDR | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | RESERVED | Reserved | R | 0x0000000 |
| 5:0 | DUMP_SLVADDR | Dump slave address Type: Control. Reset value: 0x19. | R | 0x1F |

### Table 1-264. L3_STCOL_DUMP_MSTADDR

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | DUMP_MSTADDR | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:10 | RESERVED | Reserved | R | 0x000000 |
| 9:0 | DUMP_MSTADDR | Dump master address Type: Control. (Reset value: STCOL0 - 0x3C, STCOL1 - 0x3D, STCOL2 - 0x3E, STCOL3 - 0x3F, STCOL4 - 0x37) | R | ---- |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-265. L3_STCOL_DUMP_SLVOFS

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DUMP_SLVOFS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | DUMP_SLVOFS | Dump slave offset Type: Control. Reset value: 0x800. | RW | 0x0000 0800 |

### Table 1-266. L3_STCOL_DUMP_MANUAL

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | DUMP_MANUAL |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | DUMP_MANUAL | Define the dump mode: if != 0 the dump is controlled by the Send register. Type: Control. Reset value: 0x0. | RW | 0 |

### Table 1-267. L3_STCOL_DUMP_SEND

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | DUMP_SEND |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | DUMP_SEND | In manual mode, is used to send the dump content and initialize the counters.<br>Type: Give_AutoCleared.<br>Reset value: 0x0.<br>• Dumping can be performed only if monitoring is enabled<br>• For “one shot metrics dump” the DUMP_SEND command has to be issued before disabling monitoring | RW | 0 |

**Table 1-268. L3_STCOL_FILTER_i_GLOBALEN**

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | FILTER_i_GLOBALEN |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | FILTER_i_GLOBALEN | Filter global enable Type: Control. Reset value: 0x0. | RW | 0 |

**Table 1-269. L3_STCOL_FILTER_i_EN_m**

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

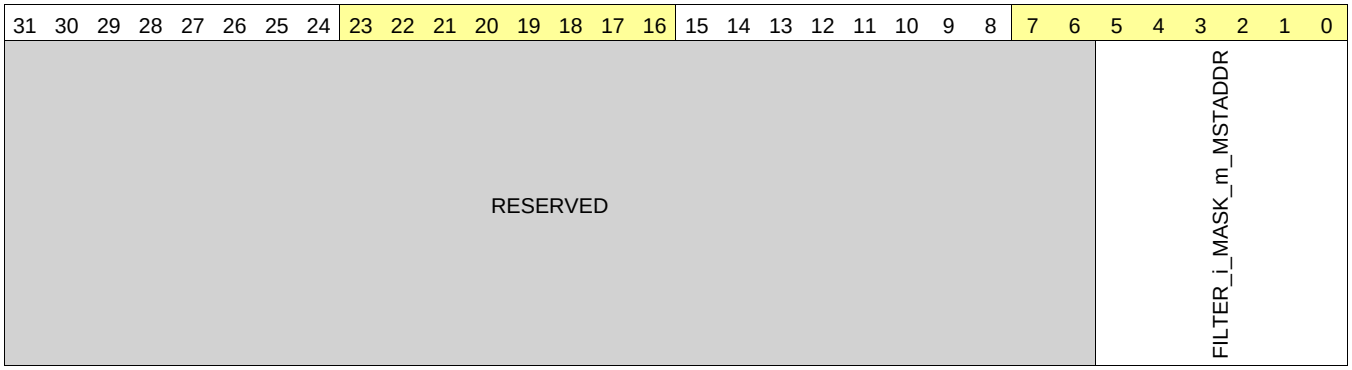
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | FILTER_i_EN0 |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | FILTER_i_EN0 | Enable filter stage 0 Type: Control. Reset value: 0x0. | RW | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-270. L3_STCOL_FILTER_i_MASK_m_MSTADDR

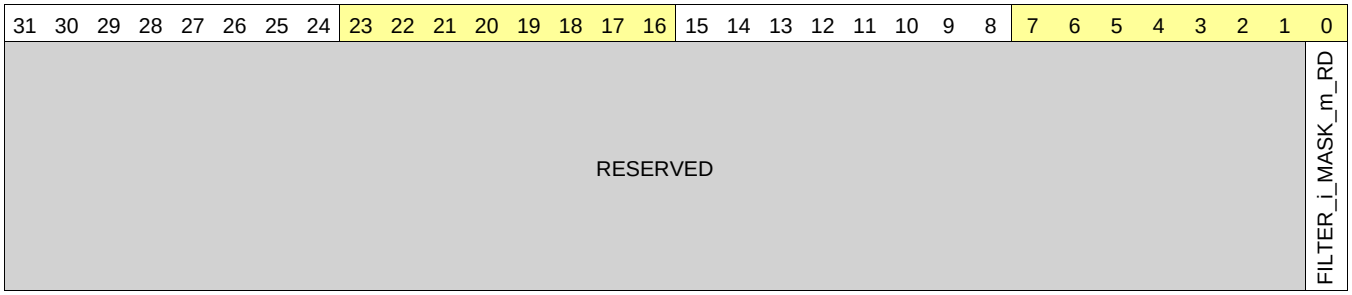
| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | RESERVED | Reserved | R | 0x000000 |
| 5:0 | FILTER_i_MASK_m_MSTADDR | Mask/Match of MstAddr Type: Control. Reset value: 0x0. | RW | 0x00 |

### Table 1-271. L3_STCOL_FILTER_i_MASK_m_RD

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

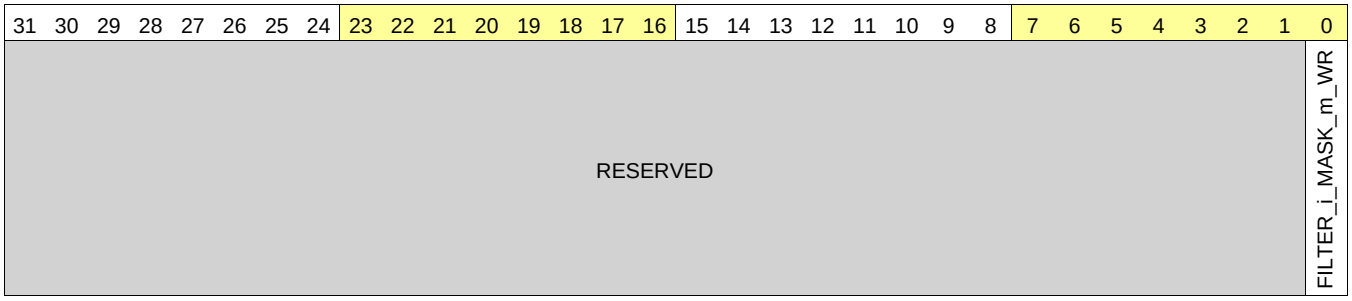



| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | FILTER_i_MASK_m_RD | Mask/Match of Rd Type: Control. Reset value: 0x0. | RW | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-272. L3_STCOL_FILTER_i_MASK_m_WR

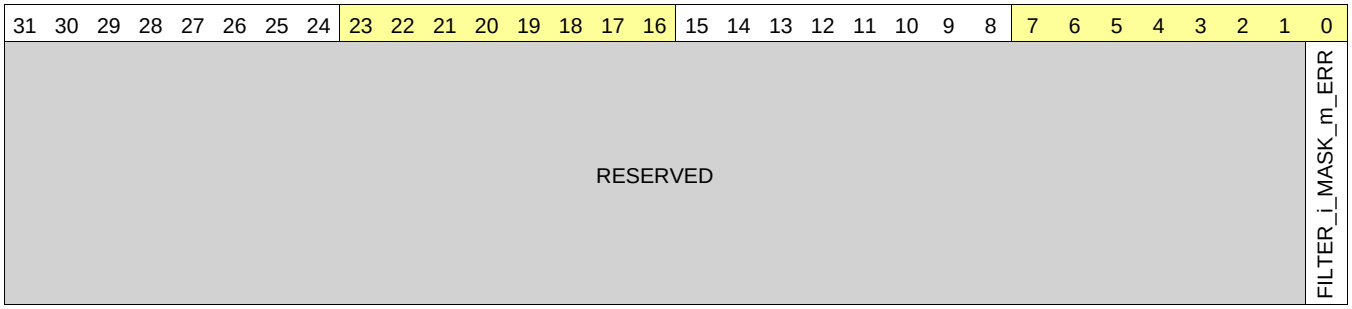
| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | FILTER_i_MASK_m_WR | Mask/Match of Wr Type: Control. Reset value: 0x0. | RW | 0 |

### Table 1-273. L3_STCOL_FILTER_i_MASK_m_ERR

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | FILTER_i_MASK_m_ERR | Mask/Match of Err Type: Control. Reset value: 0x0. | RW | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-274. L3_STCOL_FILTER_i_MASK_m_USERINFO

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | FILTER_i_MASK_m_USERINFO | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:18 | RESERVED | Reserved | R | 0x0000 |
| 17:0 | FILTER_i_MASK_m_USERINFO | Mask/Match of UserInfo Type: Control. Reset value: 0x0. | RW | 0x00000 |

### Table 1-275. L3_STCOL_FILTER_i_MASK_m_SLVADDR

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | FILTER_i_MASK_m_SLVADDR | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:5 | RESERVED | Reserved | R | 0x0000 0000 |
| 4:0 | FILTER_i_MASK_m_SLVADDR | Mask/Match of SlvAddr Type: Control. Reset value: 0x0. | RW | 0x00 |

### Table 1-276. L3_STCOL_FILTER_i_MASK_m_REQUSERINFO

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

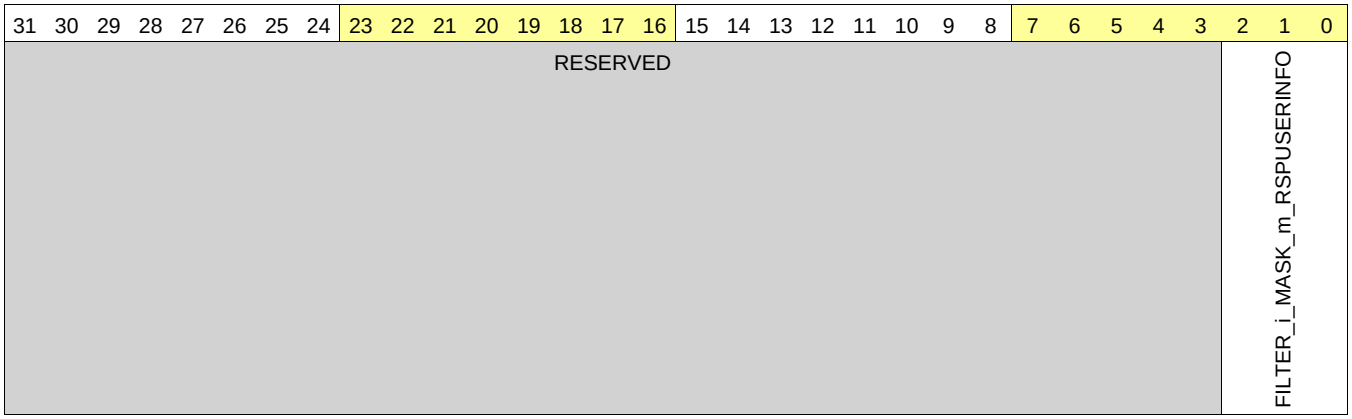
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | FILTER0_MASK0_REQUSERINFO | | | | | | | | | | | | | | | | | | | | | | | | | |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:24 | RESERVED | | R | 0x0000 0000 |
| 23:0 | FILTER_i_MASK_m_REQUSERINFO | Mask/Match of ReqUserInfo Type: Control. Reset value: 0x0. | RW | 0x00 |

### Table 1-277. L3_STCOL_FILTER_i_MASK_m_RSPUSERINFO

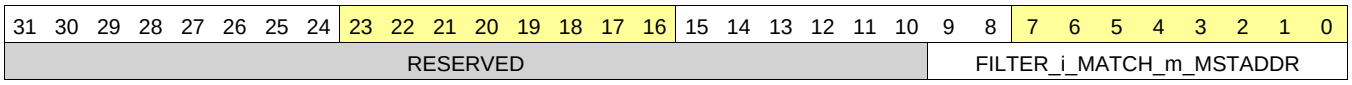
| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:3 | RESERVED | Reserved | R | 0x0000 0000 |
| 2:0 | FILTER_i_MASK_m_RSPUSERINFO | Mask/Match of RspUserInfo Type: Control. Reset value: 0x0. | RW | 0x0 |

### Table 1-278. L3_STCOL_FILTER_i_MATCH_m_MSTADDR

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |



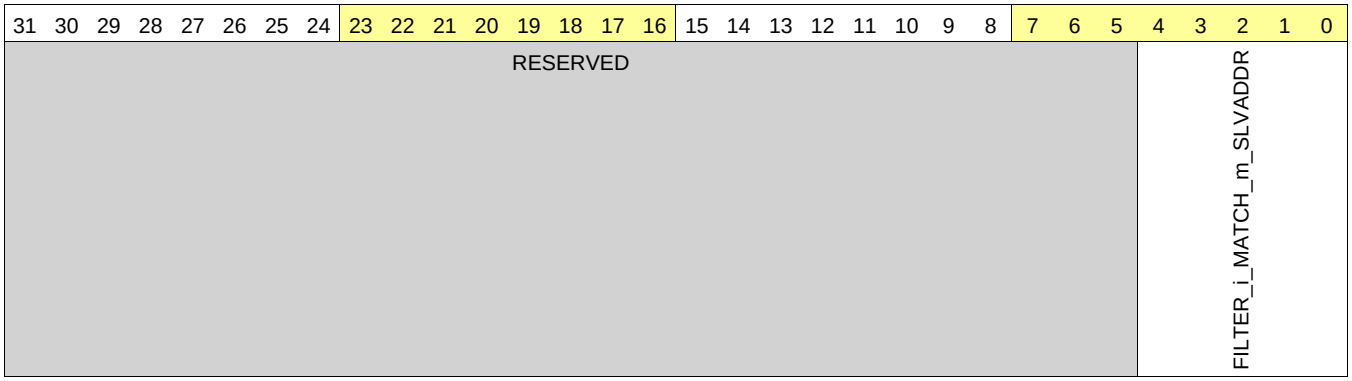

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:10 | RESERVED | Reserved | R | 0x000000 |
| 9:0 | FILTER_i_MATCH_m_MSTADDR | Mask/Match of MstAddr Type: Control. Reset value: 0x0. | RW | 0x00 |

### Table 1-279. L3_STCOL_FILTER_i_MATCH_m_SLVADDR

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |



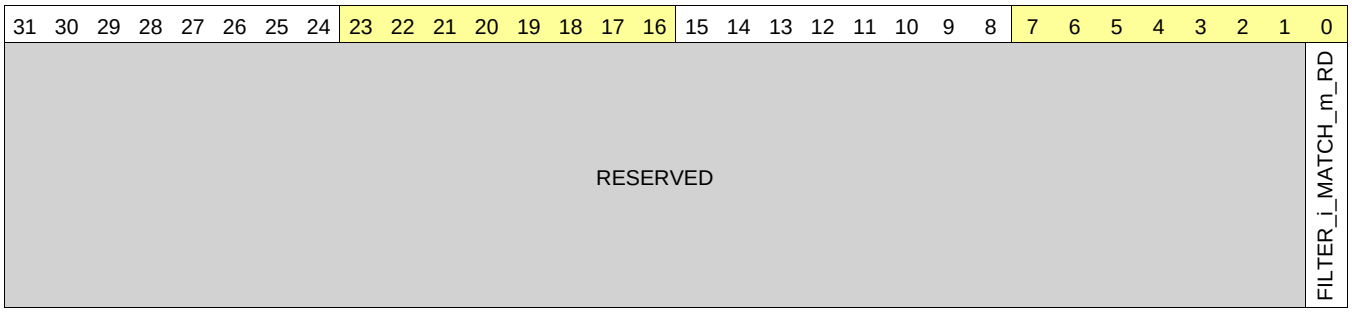
Copyright © 2011–2016, Texas Instruments Incorporated




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:5 | RESERVED | Reserved | R | 0x0000 0000 |
| 4:0 | FILTER_i_MATCH_m_SLVADDR | Mask/Match of SlvAddr Type: Control. Reset value: 0x0. | RW | 0x00 |

**Table 1-280. L3_STCOL_FILTER_i_MATCH_m_RD**

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

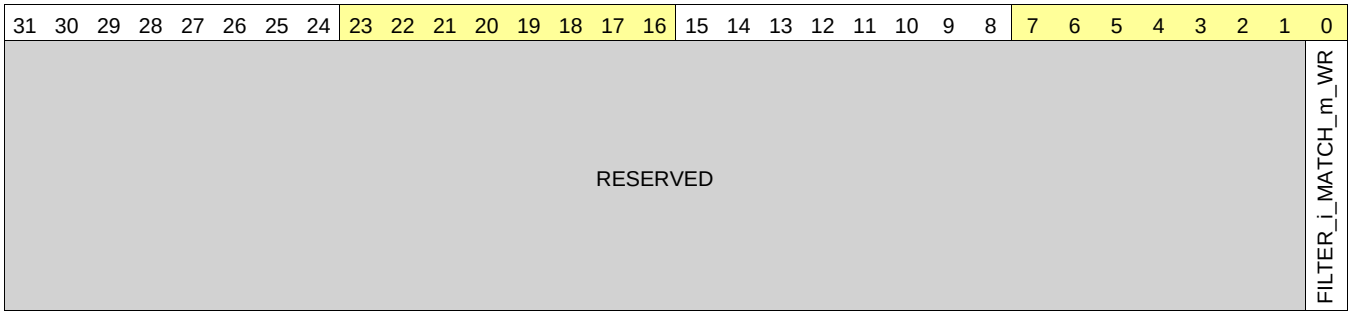



| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | FILTER_i_MATCH_m_RD | Mask/Match of Rd Type: Control. Reset value: 0x0. | RW | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-281. L3_STCOL_FILTER_i_MATCH_m_WR

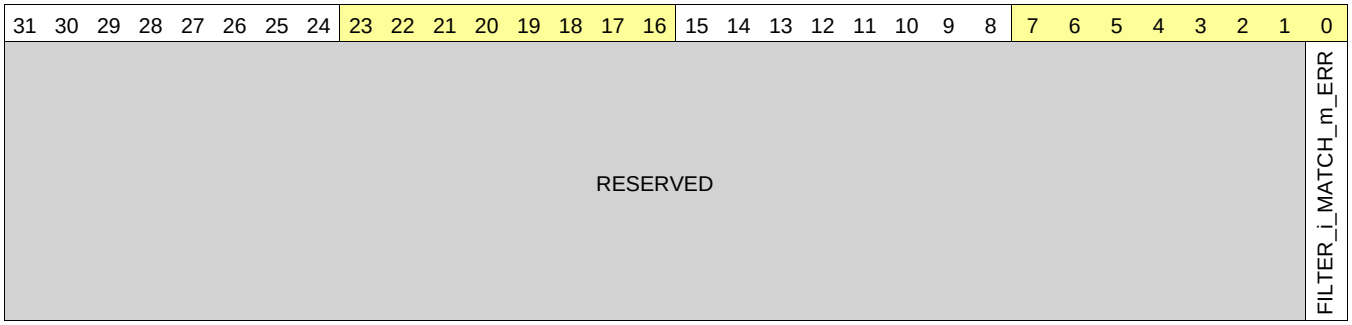
| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | FILTER_i_MATCH_m_WR | Mask/Match of Wr Type: Control. Reset value: 0x0. | RW | 0 |

### Table 1-282. L3_STCOL_FILTER_i_MATCH_m_ERR

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:1 | RESERVED | Reserved | R | 0x0000 0000 |
| 0 | FILTER_i_MATCH_m_ERR | Mask/Match of Err Type: Control. Reset value: 0x0. | RW | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-283. L3_STCOL_FILTER_i_MATCH_m_USERINFO

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | FILTER_i_MATCH_m_USERINFO | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:18 | RESERVED | Reserved | R | 0x0000 |
| 17:0 | FILTER_i_MATCH_m_USERINFO | Mask/Match of UserInfo Type: Control. Reset value: 0x0. | RW | 0x00000 |

### Table 1-284. L3_STCOL_FILTER_i_MATCH_m_REQUSERINFO

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | FILTER_i_MASK_m_REQUSERINFO | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:24 | RESERVED | Reserved | R | 0x0000 0000 |
| 23:0 | FILTER_i_MASK_m_REQUSERINFO | Mask/Match of ReqUserInfo Type: Control. Reset value: 0x0. | RW | 0x00 |

### Table 1-285. L3_STCOL_FILTER_i_MATCH_m_RSPUSERINFO

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | FILTER_i_MASK_m_RSPUSERINFO | |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:3 | RESERVED | | R | 0x0000 0000 |
| 2:0 | FILTER_i_MASK_m_RSPUSERINFO | Mask/Match of RspUserInfo Type: Control. Reset value: 0x0. | RW | 0x0 |

### Table 1-286. L3_STCOL_OP_i_THRESHOLD_MINVAL

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | OP_i_THRESHOLD_MINVAL | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:12 | RESERVED | Reserved | R | 0x00000 |
| 11:0 | OP_i_THRESHOLD_MINVAL | Min value Type: Control. Reset value: 0x0. | RW | 0x000 |

### Table 1-287. L3_STCOL_OP_i_THRESHOLD_MAXVAL

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | OP_i_THRESHOLD_MAXVAL | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:12 | RESERVED | Reserved | R | 0x00000 |
| 11:0 | OP_i_THRESHOLD_MAXVAL | Max value Type: Control. Reset value: 0x0. | RW | 0x000 |

### Table 1-288. L3_STCOL_OP_i_EVTINFOSEL

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | OP_i_EVTINFOSEL | |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | RESERVED | Reserved | R | 0x0000 0000 |
| 1:0 | OP_i_EVTINFOSEL | Select event info data to add to counter (len/press or latency) Type: Control.<br>Reset value: 0x0.<br>0x0: Select len from event info list<br>0x1: Select pressure if available from event info list<br>0x2: Select latency if available from event info list | RW | 0x0 |

## Table 1-289. L3_STCOL_OP_i_SEL

| | | | |
|---|---|---|---|
| **Address Offset** | See Table 1-250. | | |
| **Physical Address** | See Table 1-250 and Table 1-251 | **Instance** | See Table 1-250 and Table 1-251 |
| **Description** | | | |
| **Type** | RW | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | OP_i_SEL | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:4 | RESERVED | Reserved | R | 0x0000000 |
| 3:0 | OP_i_SEL | Select logical operation Type: Control. Reset value: 0x0.<br>0x0: Increment counter on each mask/match filter hit<br>0x1: Increment counter on each min/max level hit<br>0x2: Add to counter the selected event info value (len/press or latency)<br>0x3: Increment counter when all filter event hits (And(Fi))<br>0x4: Increment counter if any of filter event hits (Or(Fi))<br>0x5: Add to counter the number of current request event that hit<br>0x6: Add to counter the number of current response event that hit<br>0x7: Add to counter the number of all event that hit<br>0x8: Increment counter on each selected external event hit | RW | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.3 L4 Interconnects

This section details the device L4 interconnects.

#### 1.12.3.1 L4 Interconnect Overview

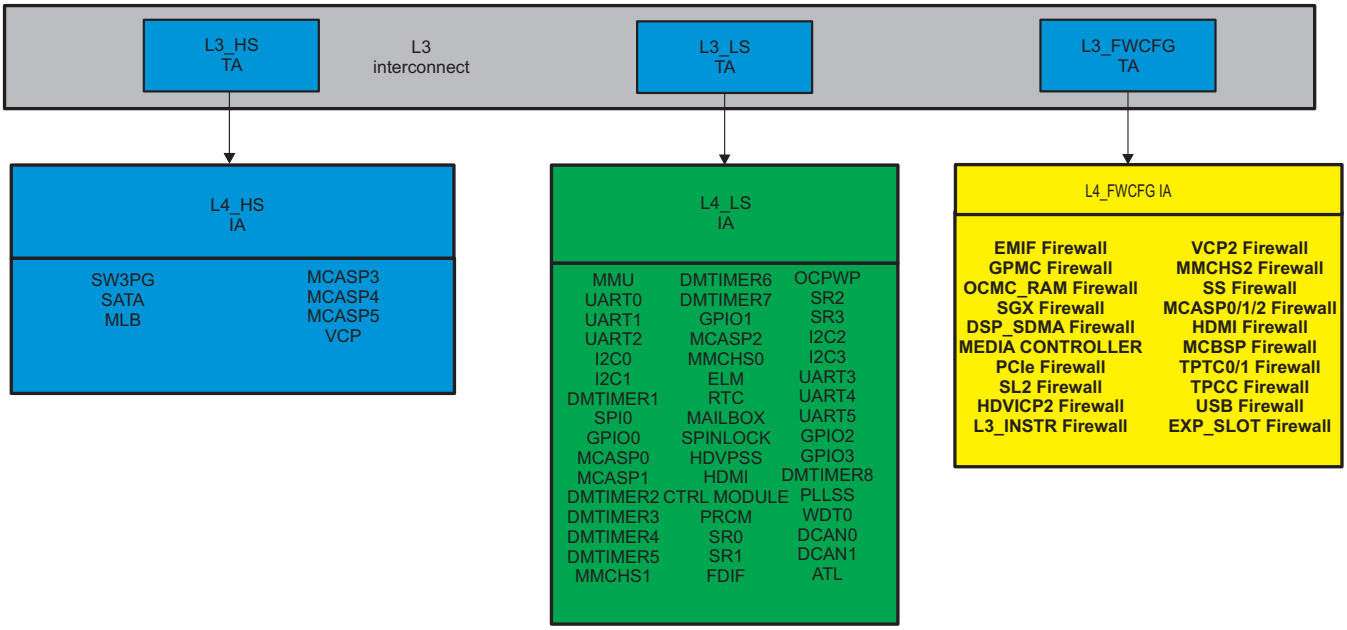
The device uses two separate L4 interconnect structures to connect peripheral modules, and one additional L4 interconnect for the firewalls. All L4s handle transfers with peripherals but are located in distinct power domains.

Figure 1-110 is an overview of the L4 interconnects and the peripherals attached to them.

The L4 interconnect is composed of the following interconnects:

- L4_HS: Configuration peripheral interconnect, for configuration of high speed masters. Located in the 220MHz clock domain.
- L4_LS: Main peripheral interconnect for peripherals that requre DMA accesses. This is a configuration interconnect for the low speed masters, located in the 110 MHz clock domain.
- L4_FWCFG: Firewall configuration interconnect located in the 110 MHz clock domain.

**Figure 1-110. L4 Interconnect Overview**




The main features of the L4 interconnects are:

- From one to two 32-bit initiator ports for each L4 interconnect instance.
- 8-, 16-, or 32-bit data, single, or burst transactions
- Little endian
- Nonblocking architecture with fair arbitration between threads
- Target interfaces: Fully synchronous or divided synchronous
- Protection logic that provides user-configurable access control to targets by each initiator
- L4LS and L4HS are each with its own PRCM interface
- All peripherals are protected by firewall

#### 1.12.3.2 L4 Interconnect Integration

Table 1-290 and Table 1-291 summarize the integration of the module in the device.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-290. Integration Attributes**

| Module Instance | Attributes |
|---|---|
| | Power Domain |
| L4_HS | Always on power domain |
| L4_LS | Always on power domain |
| L4_FWCFG | Always on power domain |

**Table 1-291. Clocks and Resets**

| Clocks | | | |
|---|---|---|---|
| **Module Instance** | **Destination Signal Name** | **Source** | **Description** |
| L4_HS | L4HS_CLK | PRCM module | Functional and interface clock |
| L4_LS | L4LS_CLK | PRCM module | Functional and interface clock |
| L4_FWCFG | L4FWCFG_CLK | PRCM module | Functional and interface clock |
| **Resets** | | | |
| **Module Instance** | **Destination Signal Name** | **Source** | **Description** |
| L4_HS | L4HS_RST | PRCM module | Reset of L4_HS interconnect |
| L4_LS | L4LS_RST | PRCM module | Reset of L4_LS interconnect |
| L4_FWCFG | L4FWCFG_RST | PRCM module | Reset of L4_FWCFG interconnect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.3.3 L4 Interconnect Functional Description

#### *1.12.3.3.1 Module Distribution*

IAs and TAs provide the interface to connect the different modules to their associated interconnect.

Table 1-292 through Table 1-297 list all the modules and subsystems with their associated agents. The agents are listed for each L4 interconnect domain.

##### *1.12.3.3.1.1 L4_HS Interconnect Agents*

Table 1-292 lists the L4 HS TAs.

**Table 1-292. L4 HS TAs**

| Module Target Name | Description |
|---|---|
| L4HS_TA_SW3PG | SW3PG L4 HS target port |
| L4HS_TA_SATA | SATA L4 HS target port |
| L4HS_TA_MLB | MLB L4 HS target port |
| L4HS_TA_MCASP3 | MCASP3 L4 HS target port |
| L4HS_TA_MCASP4 | MCASP4 L4 HS target port |
| L4HS_TA_MCASP5 | MCASP5 L4 HS target port |
| L4HS_TA_VCP | VCP L4 HS target port |

Two ports communicate beteween L3 and the L4 HS interconnect to allow L3 Initiators to access L4HS targets. Table 1-293 lists the L4HS initiator TAs.

For the list of initiators authorized to access the L4 peripheral peripherals, see *Connectivity Matrix*.

**Table 1-293. L4 HS IAs**

| Module Iniator Name | Description |
|---|---|
| L4HS_IA_0 | L3 enhanced direct memory access (EDMA) RD interconnect port |
| L4HS_IA_1 | L3 EDMA WR interconnect port |

##### *1.12.3.3.1.2 L4_LS Interconnect Agents*

The L4_LS interconnect handles only transfers to peripherals in the core power domain. Table 1-294 lists the TAs.

**Table 1-294. L4_LS TAs**

| Module Target Name | Description |
|---|---|
| L4LS_TA_MMU | System MMU L4LS target port |
| L4LS_TA_UART0 | UART0 L4LS target port |
| L4LS_TA_UART1 | UART1 L4LS target port |
| L4LS_TA_UART2 | UART2 L4LS target port |
| L4LS_TA_I2C0 | I2C0 L4LS target port |
| L4LS_TA_I2C1 | I2C1 L4 LS target port |
| L4LS_TA_DMTIMER1 | TIMER1 L4 LS target port |
| L4LS_TA_SPI0 | SPI0 L4LS target port |
| L4LS_TA_GPIO0 | GPIO0 L4LS target port |
| L4LS_TA_MCASP0 | MCASP0 L4LS target port |
| L4LS_TA_MCASP1 | MCASP1 L4LS target port |
| L4LS_TA_DMTIMER2 | TIMER2 L4LS target port |
| L4LS_TA_DMTIMER3 | TIMER3 L4LS target port |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-294. L4_LS TAs (continued)**

| Module Target Name | Description |
|---|---|
| L4LS_TA_DMTIMER4 | TIMER4 L4LS target port |
| L4LS_TA_DMTIMER5 | TIMER5 L4LS target port |
| L4LS_TA_DMTIMER6 | TIMER6 L4LS target port |
| L4LS_TA_DMTIMER7 | TIMER7 L4LS target port |
| L4LS_TA_GPIO1 | GPIO1 L4LS target port |
| L4LS_TA_MCASP2 | MCASP2 L4LS target port |
| L4LS_TA_MMCHS0 | MMCHS0 L4LS target port |
| L4LS_TA_ELM | ELM L4LS target port |
| L4LS_TA_RTC | RTC L4LS target port |
| L4LS_TA_MAILBOX | MAILBOX L4LS target port |
| L4LS_TA_SPINLOCK | SPINLOCK L4LS target port |
| L4LS_TA_HDVPSS | HDVPSS L4LS target port |
| L4LS_TA_HDMIPHY | HDMIPHY L4LS target port |
| L4LS_TA_CONTROL | Control Module L4LS target port |
| L4LS_TA_PRCM | PRCM L4LS target port |
| L4LS_TA_SR0 | Smart Reflex 0 L4LS target port |
| L4LS_TA_SR1 | Smart Reflex 1 L4 LS target port |
| L4LS_TA_OCPWP | OCPWP L4 LS target port |
| L4LS_TA_SR2 | Smart Reflex 2 L4 LS target port |
| L4LS_TA_SR3 | Smart Reflex 3 L4 LS target port |
| L4LS_TA_I2C2 | I2C2 L4 LS target port |
| L4LS_TA_I2C3 | I2C3 L4 LS target port |
| L4LS_TA_SPI1 | SPI1 L4 LS target port |
| L4LS_TA_SPI2 | SPI2 L4 LS target port |
| L4LS_TA_SPI3 | SPI3 L4 LS target port |
| L4LS_TA_UART3 | UART3 L4 LS target port |
| L4LS_TA_UART4 | UART4 L4 LS target port |
| L4LS_TA_UART5 | UART5 L4 LS target port |
| L4LS_TA_GPIO2 | GPIO2 L4 LS target port |
| L4LS_TA_GPIO3 | GPIO3 L4 LS target port |
| L4LS_TA_DMTIMER8 | TIMER8 L4 LS target port |
| L4LS_TA_PLLSS | PLL Subsystem L4 LS target port |
| L4LS_TA_WDT0 | Watchdog timer 0 L4 LS target port |
| L4LS_TA_DCAN0 | DCAN0 L4 LS target port |
| L4LS_TA_DCAN1 | DCAN1 L4 LS target port |
| L4LS_TA_ATL | ATL L4 LS target port |
| L4LS_TA_FDIF | FDIF L4 LS target port |
| L4LS_TA_MMCHS1 | MMCHS1 L4 LS target port |

For the list of initiators authorized to access the L4_LS peripherals, see *Connectivity Matrix*.

**Table 1-295. L4_LS IAs**

| Module Iniator Name | Description |
|---|---|
| L4LS_IA_0 | L3 interconnect port |
| L4LS_IA_1 | L3 interconnect port |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.12.3.3.1.3 L4_FWCFG Interconnect Agents

The L4 FWCFG interconnect handles transfers only to peripherals in the FWCFG. Table 1-296 lists the TAs. Table 1-297 lists the L4FWCFG initiator agents.

**Table 1-296. L4 FWCFG TAs**

| Module Target Name | Description |
|---|---|
| L4FWCFG_TA_EMIF | EMIF firewall |
| L4FWCFG_TA_GPMC | GPMC firewall |
| L4FWCFG_TA_OCMC_RAM | OCMC RAM firewall |
| L4FWCFG_TA_SGX | SGX firewall |
| L4FWCFG_TA_DSP_SDMA | DSP SDMA firewall |
| L4FWCFG_TA_MEDIA_CONTROLLER | Media Controller firewall |
| L4FWCFG_TA_PCIE | PCIe firewall |
| L4FWCFG_TA_SL2 | HDVICP2-0 SL2 firewall |
| L4FWCFG_TA_HDVICP2 | HDVICP2-0 CFG firewall |
| L4FWCFG_TA_VCP2 | VCP2 firewall |
| L4FWCFG_TA_MMCHS2 | MMCHS2 firewall |
| L4FWCFG_TA_ISS | ISS firewall |
| L4FWCFG_TA_MCASP0/1/2 | MCASP0/1/2 firewall |
| L4FWCFG_TA_HDMI | HDMI firewall |
| L4FWCFG_TA_MCBSP | MCBSP firewall |
| L4FWCFG_TA_TPTC0/1 | TPTC0/1 firewall |
| L4FWCFG_TA_TPCC | TPCC firewall |
| L4FWCFG_TA_USB | USB firewall |
| L4FWCFG_TA_EXP_SLOT | Expansion slot firewall |
| L4FWCFG_TA_L3_INSTR | L3 Instrumentation firewall |

**Table 1-297. L4_FWCFG IAs**

| Module Iniator Name | Description |
|---|---|
| L4FWCFG_IA_0 | L3 interconnect port |

#### 1.12.3.3.2 Power Management

Each L4 has a single power management interface connected to a power control bridge. Power control bridge is managing the different internal power interfaces. It has na external slave interface to the PRCM, an internal master interface to L4 power interface and one interface to each power bridge. The power disconnect bridge is responsible for returning OCP responses or asking for target wake up. The bridge is under the control of the PRCM.

As part of the Always On power domain L4 interconnects do not support idle state or retention.

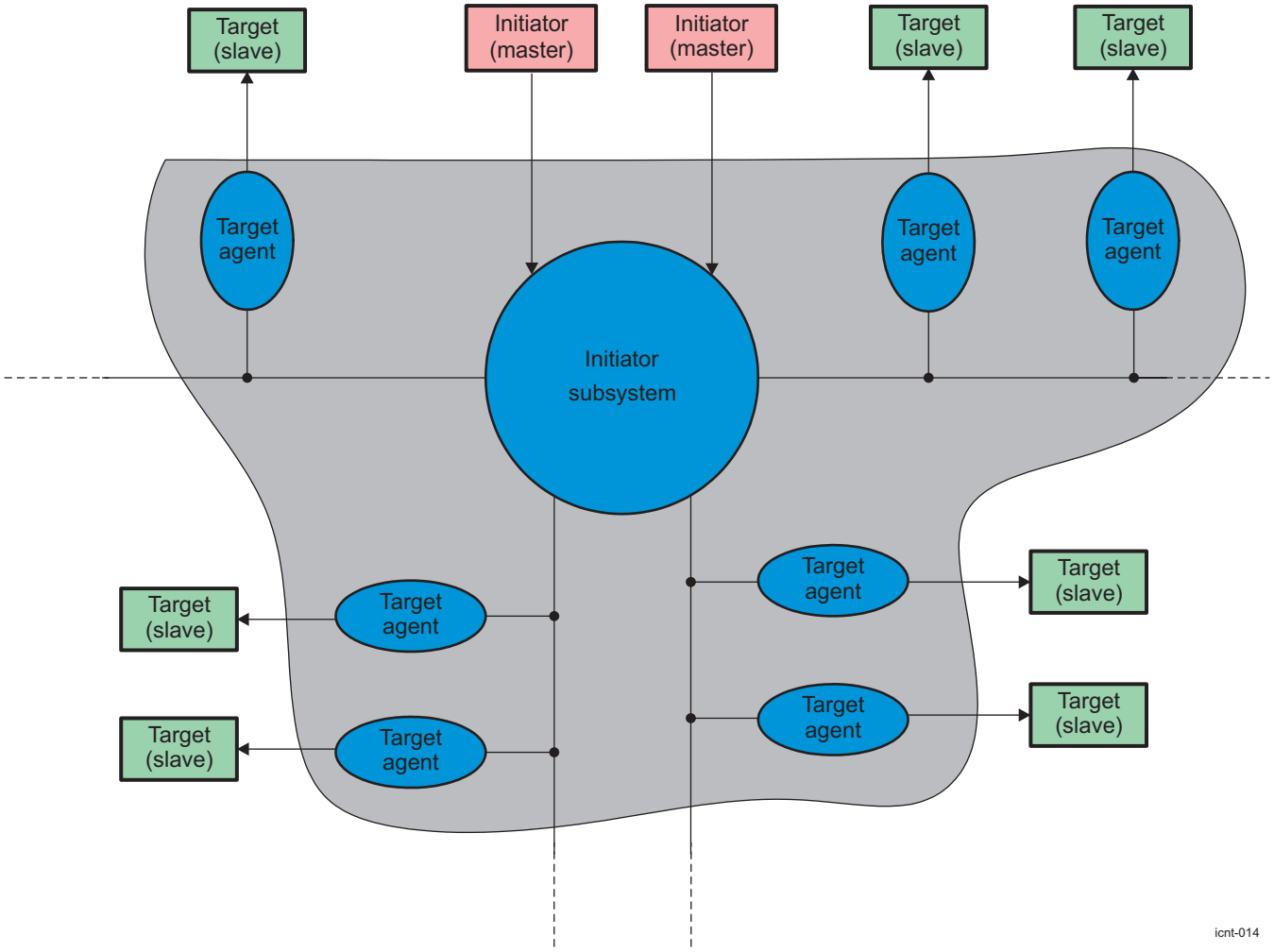
For more information about L4 interconnects power managemnet, clocking and resets see *Chapter: Power Reset and Clock management*.

#### 1.12.3.3.3 L4 Firewalls

Figure 1-111 is an internal view of the L4 interconnects in the overall interconnect. This architecture, with one initiator subsystem centralizing all initiator master requests and distributing them to all target modules (peripherals), enables the L4 interconnect firewall functions to be centralized at the L4 initiator subsystem level. The L4 firewall filters the accesses based on the configurable protection groups defined in the L4 address protection (AP) registers. Each module or TA is assigned to a protection group. The configuration is also defined in the L4 AP and is programmable on a module-per-module basis.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-111. L4 Initiator-Target Connectivity**




**NOTE:** As Figure 1-111 shows, targets are attached to branches. Branches do not impact the function of the L4 interconnect but are present to simplify timing closure and reduce active power consumption.

Because of the large address spaces and the number of peripherals connected to an L4 interconnect, two parameters are used to set up access permission:

- Programmable groups for initiators
  - Eight protection groups for L4_HS and L4_LS interconnect
- Segments divided into regions of 4KB
  - 29 regions for the L4_HS interconnect
  - 128 regions for the L4_LS interconnect
  - 47 regions for the L4_FWCFG interconnect.

Protection group members are TAs with the same protection settings. A region is programmed to allow access to a unique selectable protection group. For better protection, different regions are grouped into protection group regions and associated with a protection group member.

All L4 interconnect information, such as numbers of protection groups, regions, and segments, is available in the L4_LA_INITIATOR_INFO_L register.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.12.3.3.3.1 Protection Group

A protection group is defined by its initiators and REQ_INFO is allowed. Two registers define these two settings:

- The CONNID_BIT_VECTOR field L4_AP_PROT_GROUP_MEMBERS_k_L is a 1-bit vector that defines which initiator belongs to a group. A protection group is accessible by an initiator when the bit position corresponding to its ConnID is set to 1 in the CONNID_BIT_VECTOR field. Table 1-298 lists all the ConnID available at the L4 levels.
- The ENABLE field L4_AP_PROT_GROUP_ROLES_k_L lists all possible REQ_INFO combinations associated with the L4_AP_PROT_GROUP_MEMBERS_k_L register. Setting a Req bit in this register determines the initiators type of access. Two REQ_INFOs are used in L4 interconnects: MReqDebug and MReqSupervisor.

**NOTE:** Permissions are identical for read and write accesses in L4 interconnect.

k indicates the protection group number.

L indicates the region number.

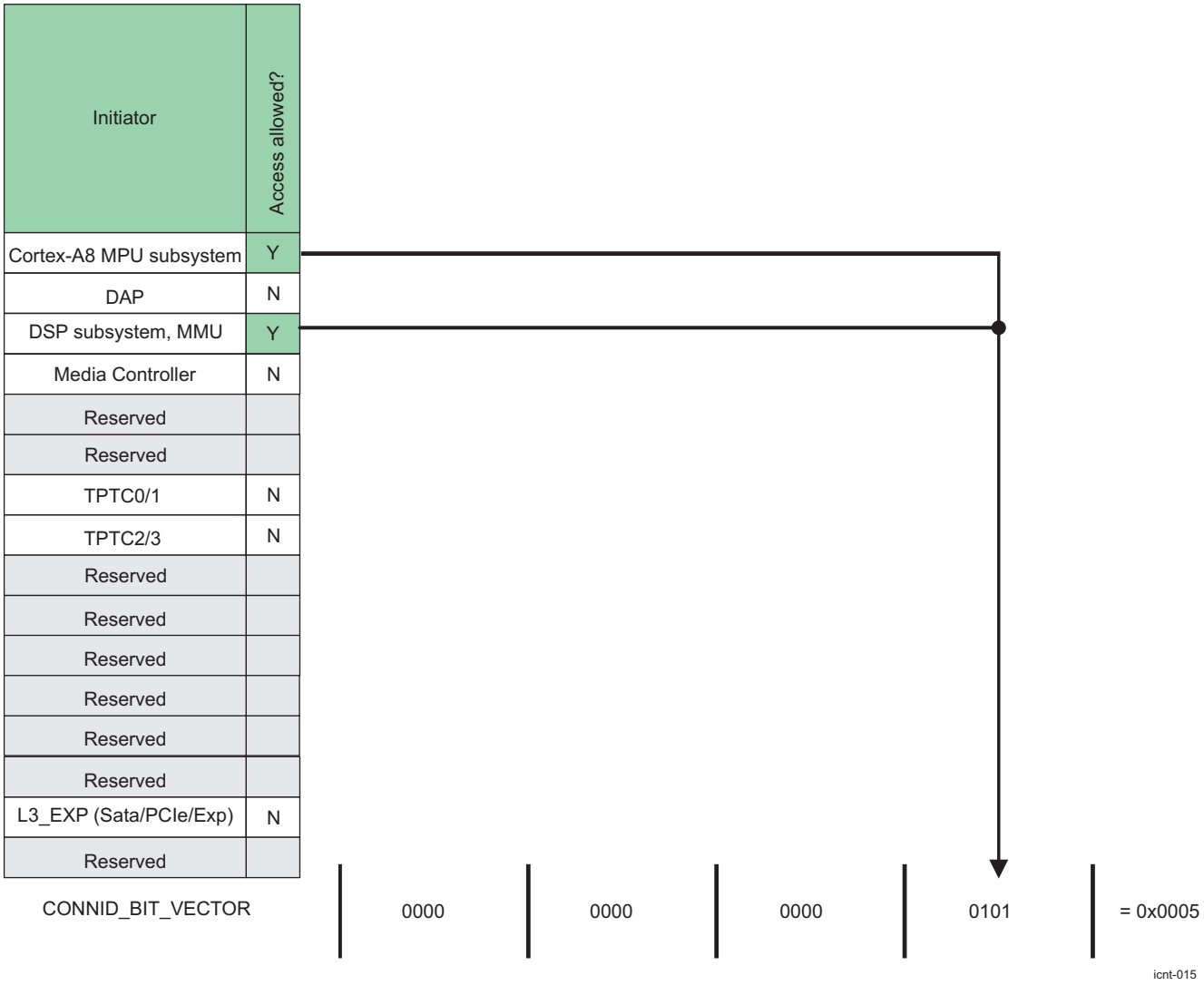
**Table 1-298. L4 ConnID Definition**

| ConnID | Initiator |
|---|---|
| 0 | Modena MPU Cortex A8 |
| 1 | DAP/ IEEE1500 |
| 2 | DSP (GEM_MDMA, GEM_CFG), MMU |
| 3 | Media Controller |
| 4 | Reserved |
| 5 | Reserved |
| 6 | TPTC0/1 |
| 7 | TPTC2/3 |
| 8 | Reserved |
| 9 | Reserved |
| A | Reserved |
| B | Reserved |
| C | Reserved |
| D | Reserved |
| E | L3_EXP (Sata/PCIe/Exp) |
| F | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

Figure 1-112 is an example of CONNID_BIT_VECTOR.

**Figure 1-112. Example of CONNID_BIT_VECTOR L4_AP_PROT_GROUP_MEMBERS_k**




Setting bits 0 and 2 in the PROT_GROUP_ID_1 register defines a group initiator that can access targets in protection group 1, and includes the following:

- Cortex-A8 MPU subsystem
- DSP subsystem

Protection group 1 can be applied to multiple protection regions with no limitation. Each protection region that is configured with protection group 1 enables permission access only to the two initiators.

The L4_AP_REGION_l_H.PROT_GROUP_ID field determines the region to which the protection group member is attached.

The values of CONNID_BIT_VECTOR and ENABLE fields are exported by the control module at reset or are user writable (see Figure 1-112 for more information).

Table 1-299 and Table 1-301 list the default configuration of the various groups for each L4 interconnect. For each group, some modules/regions are associated with it with default initiator members.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-299. L4_LS AP Registers Default Configuration**

| Group | Default Modules Associated With Group | Register | Modifiability | Default Value |
|---|---|---|---|---|
| Group 0<br>AP registers | AP registers | L4_AP_PROT_GROUP_ROLES_0_L | No | Exported |
| | | L4_AP_PROT_GROUP_MEMBERS_0_L | No | Exported |
| Group 1 | Reserved | L4_AP_PROT_GROUP_ROLES_1_L | No | Exported |
| | | L4_AP_PROT_GROUP_MEMBERS_1_L | No | Exported |
| Group 2 | Reserved | L4_AP_PROT_GROUP_ROLES_2_L | No | Exported |
| | | L4_AP_PROT_GROUP_MEMBERS_2_L | No | Exported |
| Group 3–7<br>Free | L4_LS peripherals | L4_AP_PROT_GROUP_ROLES_k_L<br>L4_AP_PROT_GROUP_ROLES_k_L | Yes | 0xFFFF FFFF (all) |

**Table 1-300. L4_HS AP Registers Default Configuration**

| Group | Default Modules Associated With Group | Register | Modifiability | Default Value |
|---|---|---|---|---|
| Group 0<br>AP registers | AP registers | L4_AP_PROT_GROUP_ROLES_0_L | No | Exported |
| | | L4_AP_PROT_GROUP_MEMBERS_0_L | No | Exported |
| Group 1-7 | L4_HS peripherals | L4_AP_PROT_GROUP_ROLES_k_L | No | 0xFFFF FFFF (all) |
| | | L4_AP_PROT_GROUP_MEMBERS_k_L | No | 0xFFFF (all) |

**Table 1-301. L4_FWCFG AP Registers Default Configuration**

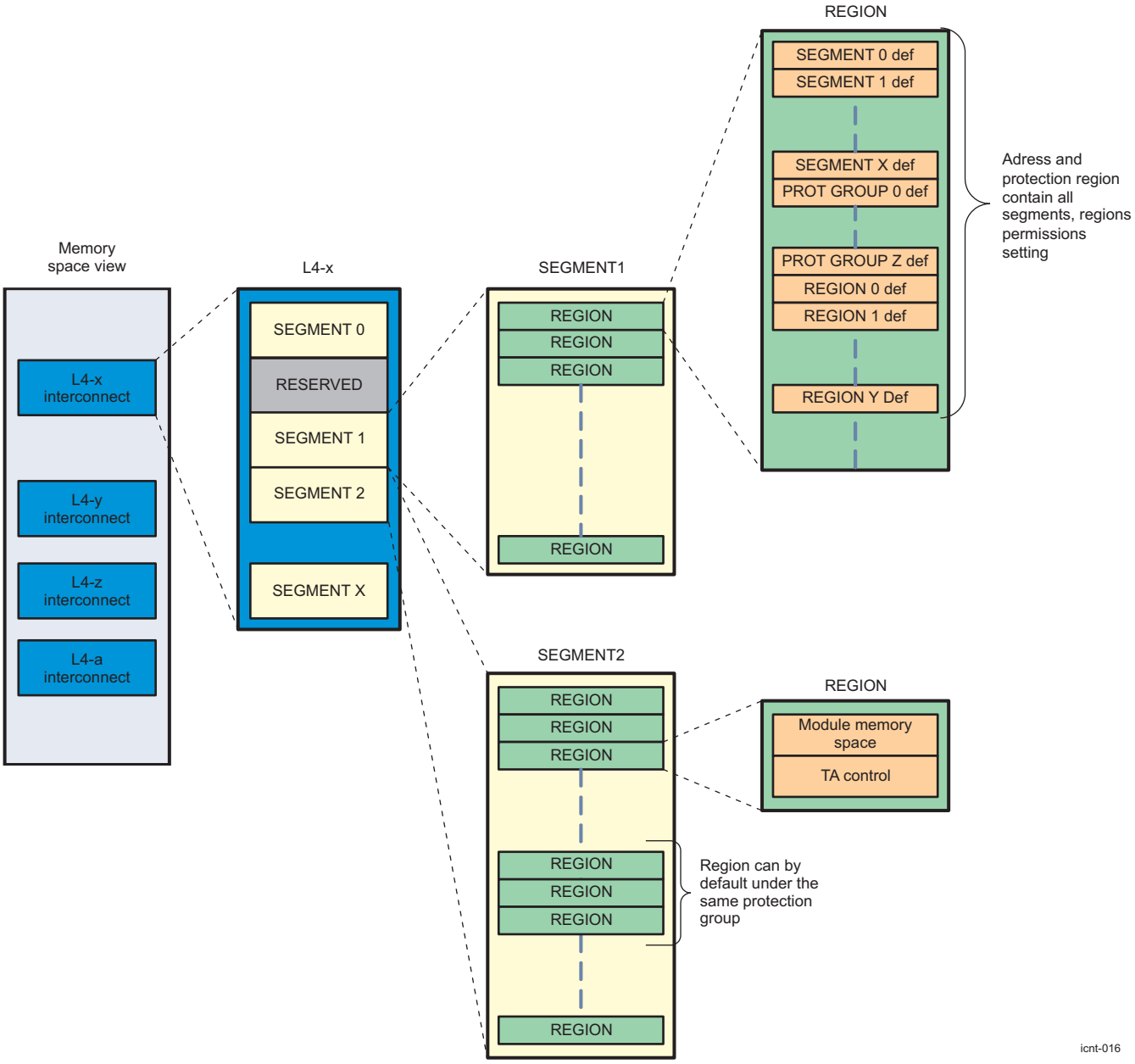
| Group | Default Modules Associated With Group | Register | Modifiability | Default Value |
|---|---|---|---|---|
| Group 0<br>AP registers | AP registers | L4_AP_PROT_GROUP_ROLES_0_L | No | Exported |
| | | L4_AP_PROT_GROUP_MEMBERS_0_L | No | Exported |
| Group 1 | Reserved | L4_AP_PROT_GROUP_ROLES_1_L | No | Exported |
| | | L4_AP_PROT_GROUP_MEMBERS_1_L | No | Exported |
| Group 2-7 | FWCFG peripherals | L4_AP_PROT_GROUP_ROLES_k_L | No | 0xFFFF FFFF (all) |
| | | L4_AP_PROT_GROUP_MEMBERS_k_L | No | 0xFFFF (all) |

#### *1.12.3.3.3.2 Segments and Regions*

The protection mechanism for L4 interconnects is based on a hierarchical segmentation, as shown in Figure 1-113. By default, some regions are attached to specific protection group members. This specificity lets users set up the permission access to certain types of modules requiring the same access protection without managing region allocation.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-113. L4 Segmentation**




All interconnect address spaces are covered by regions. Table 1-302 through Table 1-304 list the module mapping with their addresses, region numbers, and default protection group allocated to them.

**NOTE:** Module refers to the configuration registers of the module.

TA refers to the interconnect configuration registers of the TA associated with the module.

**Table 1-302. Region Allocations for L4_HS Interconnect**

| Module | Region | Description |
|---|---|---|
| L4_LS CONFIG | 0 | AP |
| | 1 | L4HS_IA_0 |
| | 2 | Link agent |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-302. Region Allocations for L4_HS Interconnect (continued)**

| Module | Region | Description |
|---|---|---|
| SW3PG | 5 | CFG |
| | 6 | Target port |
| SATA | 7 | CFG |
| | 8 | Target port |
| MLB | 9 | CFG |
| | 10 | Target port |
| MCASP3 | 12 | CFG |
| | 13 | Target port |
| | 14 | Data |
| MCASP4 | 15 | CFG |
| | 16 | Data Target port |
| | 17 | Data |
| MCASP5 | 18 | CFG |
| | 19 | Data Target port |
| | 20 | Data |
| VCP2 | 21 | CFG |
| | 22 | Target port |
| MCASP3 | 25 | CFG Target port |
| MCASP4 | 26 | CFG Target port |
| MCASP5 | 27 | CFG Target port |
| SW4PG | 28 | Target port |

**Table 1-303. Region Allocations for L4_LS Interconnect**

| Module | Region | Description |
|---|---|---|
| L4LS CONFIG | 0 | AP |
| | 1 | CFG_IA_0 |
| | 2 | Link agent |
| UART0 | 3 | Target port |
| UART1 | 4 | Target port |
| | 5 | Module |
| UART2 | 6 | Target port |
| | 7 | Module |
| I2C0 | 8 | Target port |
| | 9 | Module |
| I2C1 | 10 | Target port |
| | 11 | Module |
| UART0 | 14 | Module |
| DMTIMER1 | 15 | Target port |
| | 16 | Module |
| SPI0 | 17 | Target port |
| | 18 | Module |
| GPIO0 | 19 | Target port |
| | 20 | Module |
| WATCHDOG Timer | 21 | Target port |
| | 22 | Module |
| MMCHS0 | 23 | Target port |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-303. Region Allocations for L4_LS Interconnect (continued)**

| Module | Region | Description |
|---|---|---|
| | 24 | Module |
| RTC | 25 | Target port |
| | 26 | Module |
| PRCM | 27 | Target port |
| | 28 | Module |
| MCASP1 | 29 | Target port |
| | 30 | Module |
| MCASP2 | 31 | Target port |
| | 32 | Module |
| MCASP0 | 33 | Target port |
| | 34 | Module |
| IEEE1500 | 35 | Target port |
| | 36 | Module |
| OCPWP | 37 | Target port |
| | 38 | Module |
| L4LS_IA1 | 39 | CFG_IA_1 |
| EFUSE_CTRL | 40 | Module |
| | 41 | Target port |
| MMU | 42 | Target port |
| | 43 | Module |
| SMARTCARD0 | 44 | Target port |
| | 45 | Module |
| DMTIMER2 | 46 | Target port |
| | 47 | Module |
| DMTIMER3 | 48 | Target port |
| | 49 | Module |
| DMTIMER4 | 50 | Target port |
| | 51 | Module |
| DMTIMER5 | 52 | Target port |
| | 53 | Module |
| DMTIMER6 | 54 | Target port |
| | 55 | Module |
| DMTIMER7 | 56 | Target port |
| | 57 | Module |
| GPIO1 | 58 | Target port |
| | 59 | Module |
| ELM | 60 | Target port |
| | 61 | Module |
| SPARE1 | 62 | Target port |
| | 63 | Module |
| MAILBOX | 64 | Target port |
| | 65 | Module |
| Spinlock | 66 | Target port |
| | 67 | Module |
| PRCM | 68 | Target port |
| Smart Reflex 0 | 69 | Target port |
| | 70 | Module |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-303. Region Allocations for L4_LS Interconnect (continued)**

| Module | Region | Description |
|---|---|---|
| Smart Reflex 1 | 71 | Target port |
| | 72 | Module |
| HDVPSS | 73 | Target port |
| | 74 | Module |
| HDMI | 75 | Target port |
| | 76 | Module |
| Smart Reflex 2 | 77 | Target port |
| | 78 | Module |
| Smart Reflex 3 | 79 | Target port |
| | 80 | Module |
| I2C2 | 81 | Target port |
| | 82 | Module |
| I2C3 | 83 | Target port |
| | 84 | Module |
| SPI1 | 85 | Target port |
| SPI3 | 86 | Target port |
| | 87 | Module |
| GPIO2 | 88 | Target port |
| | 89 | Module |
| GPIO3 | 90 | Target port |
| | 91 | Module |
| ATL | 92 | Target port |
| | 93 | Module |
| UART3 | 94 | Target port |
| | 95 | Module |
| UART4 | 96 | Target port |
| | 97 | Module |
| UART5 | 98 | Target port |
| | 99 | Module |
| SMARTCARD1 | 100 | Target port |
| | 101 | Module |
| Control | 102 | Target port |
| | 103 | Module |
| SPI1 | 104 | Module |
| SPI2 | 105 | Target port |
| | 106 | Module |
| SPARE2 | 107 | Target port |
| | 108 | Module |
| PATA | 109 | Target port |
| | 110 | Module |
| DCAN0 | 111 | Target port |
| | 112 | Module |
| DCAN1 | 113 | Target port |
| | 114 | Module |
| FDIF | 115 | Target port |
| | 116 | Module |
| MMCHS1 | 117 | Target port |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-303. Region Allocations for L4_LS Interconnect (continued)**

| Module | Region | Description |
|---|---|---|
| | 118 | Module |
| | 119 | Target port |
| DMTIMER8 | 120 | Target port |
| | 121 | Module |
| SYNCTIMER32K | 122 | Target port |
| | 123 | Target port |
| PLLSS | 124 | Target port |
| | 125 | Module |

**Table 1-304. Region Allocations for L4_FWCFG Interconnect**

| Module | Region | Description |
|---|---|---|
| L4 FW CONFIG | 0 | AP |
| | 1 | FWCFG_IA_0 |
| | 2 | Link agent |
| EMIF_FW | 3 | Target port |
| | 4 | Module |
| GPMC_FW | 5 | Target port |
| | 6 | Module |
| OCMCRAM0_FW | 7 | Target port |
| | 8 | Module |
| SGX_FW | 9 | Target port |
| | 10 | Module |
| GEMSDMA_FW | 11 | Target port |
| | 12 | Module |
| MEDIACONTROLLER_FW | 13 | Target port |
| | 14 | Module |
| PCIE_FW | 15 | Target port |
| | 16 | Module |
| SL2_FW | 17 | Target port |
| | 18 | Module |
| HDVICP_FW | 19 | Target port |
| | 20 | Module |
| MCASP_FW | 21 | Target port |
| | 22 | Module |
| HDMI_FW | 25 | Target port |
| | 26 | Module |
| EMUSS_FW | 27 | Target port |
| | 28 | Module |
| TPTC_FW | 29 | Target port |
| | 30 | Module |
| TPCC_FW | 31 | Target port |
| | 32 | Module |
| MCBSP_FW | 33 | Target port |
| | 34 | Module |
| USB_TARG_FW | 37 | Target port |
| | 38 | Module |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-304. Region Allocations for L4_FWCFG Interconnect (continued)**

| Module | Region | Description |
|---|---|---|
| VCP2_FW | 39 | Target port |
| | 40 | Module |
| ISS_FW | 41 | Target port |
| | 42 | Module |
| MMCHS2_FW | 43 | Target port |
| | 44 | Module |
| EXP_FW | 45 | Target port |
| | 46 | Module |

#### 1.12.3.3.3 L4 Address and Protection Register Settings

Table 1-305 lists the settings of the AP registers relative to an L4 interconnect firewall. These values are computed based on the physical implementation of each L4 interconnect.

**Table 1-305. L4 Address and Protection Register Description Overview**

| Register Type | Register Name | Bits | Field | Description |
|---|---|---|---|---|
| Segment | L4_AP_SEGMENT_i_L (1) | 23:0 | BASE | Segment base address |
| | L4_AP_SEGMENT_i_H (1) | 4:0 | SIZE | Segment size equals to $2^{SIZE}$ |
| Protection groups | L4_AP_PROT_GROUP_MEMBERS_k_L (2) | 15:0 | CONNID_BIT_VECTOR | For L4ConnID, see Table 1-298. |
| | L4_AP_PROT_GROUP_ROLES_k_L (2) | 31:0 | ENABLE | See for REQ_INFO description(3) |
| | L4_AP_PROT_GROUP_ROLES_k_H | 31:0 | ENABLE | See for REQ_INFO description(3) |
| Region setting | L4_AP_REGION_l_L (4) | 23:0 | BASE | Defines the base address of region with respect to its segment base address |
| | L4_AP_REGION_l_H (4) | 31:28 | MADDRSPACE | Target OCP MAddrSpace value |
| | | 26:24 | SEGMENT_ID | Segment ID number of the region |
| | | 22:20 | PROT_GROUP_ID | Protection group member attached to the region |
| | | 18:17 | BYTE_DATA_WIDTH_EXP | Determines the number of byte in an access |
| | | 14:8 | PHY_TARGET_ID | Physical target ID |
| | | 6:1 | SIZE | Size of the region equals to $2^{SIZE}$ |
| | | 0 | ENABLE | Enables the region protection |

(1) i = 0 to 1 forHS_AP
i = 0 to 2 for LS_AP
(2) k = 0 to 7 for HS_AP
k = 0 to 7 for LS_AP
(3) For L4 interconnects, only MReqDebug and MReqSupervisor are available.
(4) l = 0 to 28 for HS_AP
l = 0 to 127 for LS_AP

The value of L4_AP_PROT_GROUP_ROLES_k is exported at the chip boundary. Those values are latched in the register at reset or by using the load firewall mechanism, which allows to latch exported values simply by writing into a register of the control module, this avoids resetting the interconnect in this case. The values of the other protection registers can be modified on the fly.

#### 1.12.3.3.4 L4 Error Detection and Reporting

##### 1.12.3.3.4.1 IA and TA Error Detection and Logging

The L4 interconnect provides mechanisms for handling internally detected errors or errors reported by modules attached to the L4 target ports.


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** L4_IA denotes the IA for all L4 interconnects: L4_LS, L4_HS and FWCFG.

L4_TA denotes the TA for all L4 interconnects: L4_LS, L4_HS and FWCFG.

The L4 interconnects handle four types of errors:

- No target core found or address hole, detected and logged at IA
- Unsupported command, detected and logged at IA
- Protection violation, detected and logged at IA (see Section 1.12.3.3.3, *L4 Firewalls*)
- Target does not service a request before a time-out expires. The error is detected and logged at TA (see Section 1.12.3.3.4.2, *Time-Out*).

Table 1-306 lists the value of the L4_IA_ERROR_LOG_L[25:24] CODE bit field stored when an error occurs.

**Table 1-306. L4 CODE Bit Field Definition**

| CODE (bits 1:0) | Error Type | REQ_INFO | Secondary | ConnID | CMD |
|---|---|---|---|---|---|
| 0 | No error | | | | |
| 1 | Unsupported command | x | x | x | x |
| 2 | Address hole | x | x | x | x |
| 3 | Protection violation | x | | x | x |

- No target core found/address hole: This error indicates that a request was addressed to a hole in the L4 address map. When this error occurs, an in-band error response is returned to the L3 level. The error is also logged into the L4_IA_AGENT_STATUS_L[27] INBAND_ERROR bit. Additionally, an address hole error code is logged to the L4_IA_ERROR_LOG_L[25:24] CODE bit field.
- Unsupported command: This error indicates that the command type of the request is not supported by the accessed target register. The error is logged into the L4_IA_AGENT_STATUS_L[27] INBAND ERROR bit. An unsupported command error code is written to the L4_IA_ERROR_LOG_L[25:24] CODE bit field for the initiator interface.
- Protection violation: This error indicates that a request is not issued from an allowed initiator member or is issued with the inappropriate ReqInfo qualifiers associated with the target region. This error is reported using an in-band error and is written to the L4_IA_AGENT_STATUS_L[27] INBAND_ERROR bit. A protection violation error code is saved into the L4_IA_ERROR_LOG_L[25:24] CODE bit field for the same initiator interface. A protection violation is also logged in the L4_IA_AGENT_STATUS_L[31] PROT_ERROR_SECONDARY or [30] PROT_ERROR_PRIMARY bit when in debug or applicative mode, respectively.

  The L4_IA_ERROR_LOG_L[30] SECONDARY bit indicates whether the error occurred in application or debug.

  The L4_IA_ERROR_LOG_H[15:0] REQ_INFO bit field returns the type access (REQ_INFO qualifier) that caused the error.

  The L4_IA_ERROR_LOG_L[13:8] CONNID bit field returns the ID of the initiator that caused the error.

  The L4_IA_ERROR_LOG_ADDR_L[31:0] register logs the address for error conditions.

  A L4_IA_ERROR_LOG_L[31] MULTI bit is asserted when multiple errors are detected. In this case, the error code corresponds to the first error that occurs.

#### *1.12.3.3.4.2 Time-Out*

A time-out mechanism can be enabled at the interconnect level and on a per-target basis. If the mechanism is enabled for a TA and interconnect and commands are not accepted or responses are not returned within the expected delay, the L4 interconnect generates an error event.


Copyright © 2011–2016, Texas Instruments Incorporated

The error is logged in the L4_TA_AGENT_STATUS_L[10:8] REQ_TIMEOUT bit. The affected TA enters an error state that causes it to send an error response to any new request targeted at it. To recover from this state, the TA must be reset by the system software. The time-out is counted starting from the moment a command is presented to the target, regardless of the target response to this command.

The L4 interconnect implements a centralized time-base circuit that broadcasts a set of four periodic pulse signals to all connected TAs. The time-base circuit offers four possible sets of four time-base signals. Selection is done by programming the L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE bit field.

The selected time-base signals are available at any TA. Each TA can be programmed to refer to one of these four time-base signals, using the L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT bit field. These four signals are referred to as 1X time-base, 4X time-base, 16X time-base, and 64X time-base.

Table 1-307 lists all values in number of L4 clock cycles.

**Table 1-307. L4 Time-Out Link and TA Programming**

| | L4_TA_AGENT_CONTROL_L REQ_TIMEOUT[2:0] | | | | |
|---|---|---|---|---|---|
| **L4_LA_NETWORK_CONTROL_L TIMEOUT_BASE[2:0]** | **0** | **1** | **2** | **3** | **4** |
| 0 | All L4 time-out features are disabled. | | | | |
| 1 | Locally disabled | 64 | 256 | 1024 | 4096 |
| 2 | | 256 | 1024 | 4096 | 16,384 |
| 3 | | 1024 | 4096 | 16,384 | 65,536 |
| 4 | | 4096 | 16,384 | 65,536 | 262,144 |

The default reset value is 0x2 for REQ_TIMEOUT and 0x4 for TIMEOUT_BASE, implying 16,384 clock cycles.

A time-out condition is detected when the command acceptance or the response is not received after a delay of one to three time-base periods.

On detection of a time-out condition, the TA logs the error in L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT bit, and it also reports the error to the corresponding IA, which forward it to the L3 interconnects.

After the time-out is detected and logged, the behavior of the attached module is ignored. A new request targeting the module arriving at the timed-out TA receives an error response. If the request is addressed to the agent internal registers, it is processed normally.

To recover from a time-out error, software is assumed to first reset the faulty module and then the TA using the L4_TA_AGENT_CONTROL_L[0] OCP_RESET bit.

#### *1.12.3.3.4.3 Error Reporting*

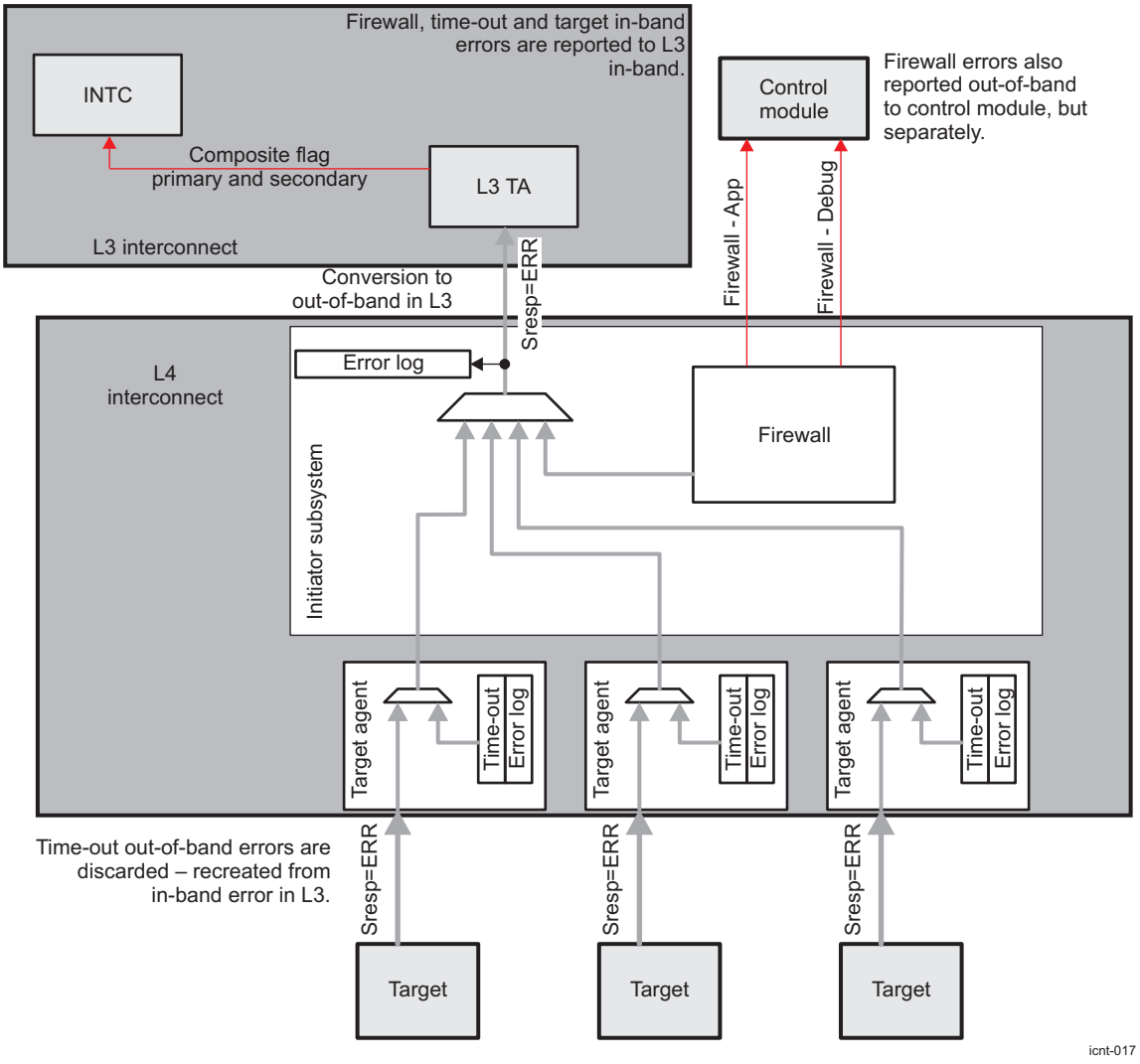
Figure 1-114 shows the error-reporting scheme used in the L4 interconnects. All L4 in-band errors are reported to L3 Interconnect with SRESP = ERR, as well as being stored in the appropriate registers. Any target error response is send back to the initiator via L3 interconnect.

Protection errors are always reported out-of-band to the Control Module. This asynchronous reporting uses separate flags for debug requests and for non-debug reuqests. Out-of-band firewall error reporting can be disabled by software from the L4_IA_AGENT_CONTROL register.

By setting the register L4_LA_FLAG_MASK_i_L registers errors comming form different initiator ports can be masked independently. The default settings of these registers are 0x3 so that all errors can be reported. In case of error, the L4_LA_FLAG_STATUS_i_L registers can be read to determine the error source.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-114. L4 Error Reporting**




### *1.12.3.3.4.4 Error Recovery*

Writing 1 to the L4_TA_AGENT_CONTROL_L[0] OCP_RESET bit initiates the software reset period. Software reset must be asserted for at least 16 cycles of the target module interface clock, which can be a divided clock with respect to the L4 clock.

During the software reset period:

- Requests sent to the target module receive error responses. Therefore, if the faulty request is part of a DMA transfer, it is necessary to stop the DMA to prevent unwanted errors.
- Requests sent to the TA register block are processed as usual.
- The L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT status bit is cleared.

Writing 0 to the L4_TA_AGENT_CONTROL_L[0] OCP_RESET bit terminates the software reset period.

The attached module must then be reset to complete the recovery.


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.3.4 L4 Interconnect Programming Guide

#### 1.12.3.4.1 L4 Interconnect Low-level Programming Models

This section describes the low-level hardware programming sequences for configuring and using the L4 interconnect module.

##### 1.12.3.4.1.1 Global Initialization

###### 1.12.3.4.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the L4 interconnect module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the L4 interconnect. For more information, see Section 1.12.3.2, *L4 Interconnect Integration* .

**Table 1-308. Global Initialization of Surrounding Modules**

| Surrounding Modules | Comments |
|---|---|
| PRCM | For more information about the configuration of the module, see the *PRCM* chapter. |
| Control module | For more information about the configuration of the module, see the *Control Module* chapter. |
| MPU INTC | The MPU INTC must be configured to enable the interrupts from L4 interconnect module. See the *ARM Interrupt Controller (AINTC)* chapter. |
| EDMA | For more information about the configuration of the EDMA, see the *EDMA* chapter. |
| L3 interconnect | For more information about the interconnect configuration, see the *L3 Interconnect* section. |

##### 1.12.3.4.1.2 Operational Modes Configuration

###### 1.12.3.4.1.2.1 L4 Interconnect Error Analysis Mode

###### 1.12.3.4.1.2.1.1 Main Sequence: L4 Interconnect Error Analysis Mode

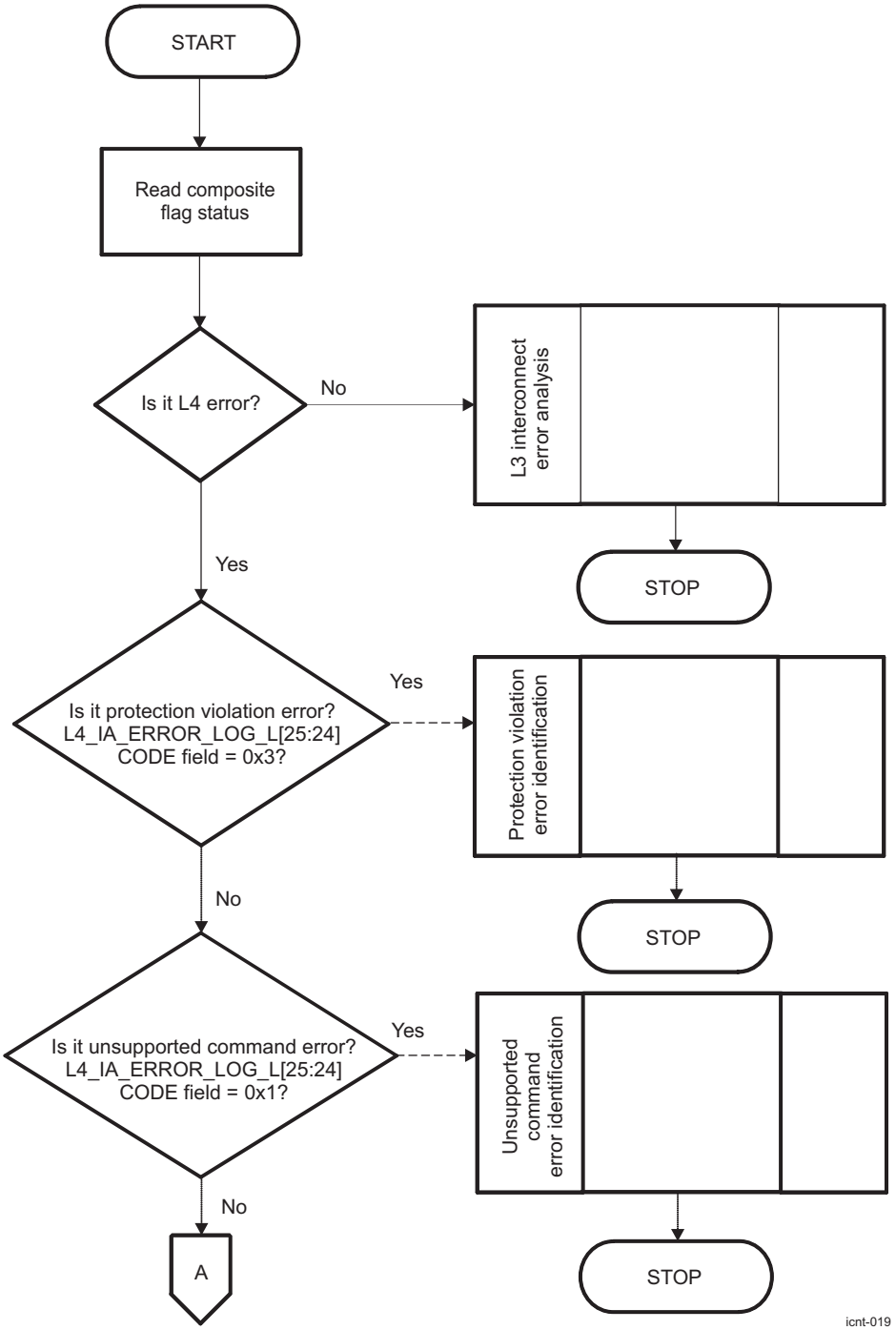
The information required to analyze an error source is logged in several registers.The number of registers to access depends on the error source.

Figure 1-115 and Figure 1-116 show the software sequence required in most cases.

Table 1-309 and Table 1-310 show the main sequence for error analysis mode and its subprocess call summary, respectively.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-115. Typical Error Analysis Sequence**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 1-116. Typical Error Analysis Sequence**

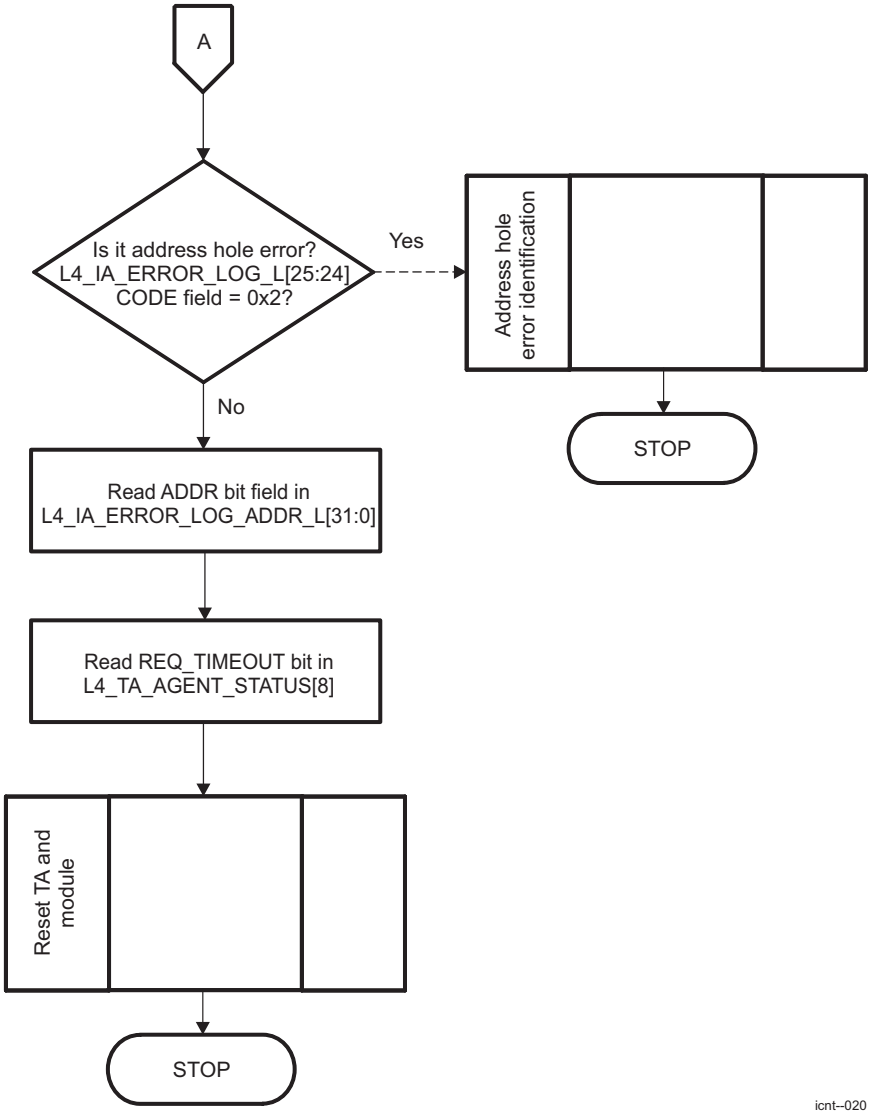



**Table 1-309. Main Sequence – Error Analysis Mode**

| Register Name | Register Name | Register Name |
|---|---|---|
| L4_IA_ERROR_LOG_L | L4_IA_ERROR_LOG_ADDR_L | L4_IA_AGENT_STATUS_L |
| L4_IA_AGENT_CONTROL_L | L4_TA_AGENT_STATUS_L | |

**Table 1-310. Subprocess Call Summary for Main Sequence – Error Analysis Mode**

| Subprocess | Cross-Reference |
|---|---|
| L3 interconnect error analysis | , *L3 Interconnect Error Handling* |
| L4 interconnect protection violation error identification | Section 1.12.3.4.1.2.1.2, *Subsequence: L4 Interconnect Protection Violation Error Identification* |
| L4 interconnect unsupported command/address hole error identification | Section 1.12.3.4.1.2.1.3, *Subsequence: L4 Interconnect Unsupported Command/Address Hole Error Identification* |
| L4 interconnect reset TA and module | Section 1.12.3.4.1.2.1.4, *Subsequence: L4 Interconnect Reset TA and Module* |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.12.3.4.1.2.1.2 Subsequence: L4 Interconnect Protection Violation Error Identification

This procedure describes the protection violation error identification (see Table 1-311).

**Table 1-311. Protection Violation Error Identification**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Read multiple errors detection. | L4_IA_ERROR_LOG_L[31] MULTI | |
| Read initiator ID. | L4_IA_ERROR_LOG_L[13:8] CONNID | |
| Read command that cause the error. | L4_IA_ERROR_LOG_L[3:0] CMD | |
| Read address of request that caused the error. | L4_IA_ERROR_LOG_ADDR_L[31:0] ADDR | |
| **IF**: Is it a primary error? | L4_IA_AGENT_STATUS_L[30] PROT_ERROR_PRIMARY | =0x1 |
| Write 1 to clear IA status bit. | L4_IA_AGENT_STATUS_L[30] PROT_ERROR_PRIMARY | 0x1 |
| **ELSE** | | |
| Write 1 to clear IA status bit | L4_IA_AGENT_STATUS_L[31] PROT_ERROR_SECONDARY | 0x1 |
| **ENDIF** | | |
| Write 1 to clear multiple errors detection. | L4_IA_ERROR_LOG_L[31] MULTI | 0x1 |
| Write 1 to clear in-band error status. | L4_IA_AGENT_STATUS_L[27] INBAND_ERROR | 0x1 |

#### 1.12.3.4.1.2.1.3 Subsequence: L4 Interconnect Unsupported Command/Address Hole Error Identification

This procedure describes the identification of unsupported command/address hole error (see Table 1-312).

**Table 1-312. Unsupported Command/Address Hole Error Identification**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Read multiple errors detection. | L4_IA_ERROR_LOG_L[31] MULTI | |
| Read initiator ID. | L4_IA_ERROR_LOG_L[13:8] CONNID | |
| Read command that caused the error. | L4_IA_ERROR_LOG_L[3:0] CMD | |
| Read address of request that caused the error. | L4_IA_ERROR_LOG_ADDR_L[31:0] ADDR | |
| Read secondary status. | L4_IA_ERROR_LOG_L[30] SECONDARY | |
| Write 1 to clear secondary status. | L4_IA_ERROR_LOG_L[30] SECONDARY | 0x1 |
| Write 1 to clear multiple errors detection. | L4_IA_ERROR_LOG_L[31] MULTI | 0x1 |
| Write 1 to clear inband error status. | L4_IA_AGENT_STATUS_L[27 INBAND_ERROR | 0x1 |

#### 1.12.3.4.1.2.1.4 Subsequence: L4 Interconnect Reset TA and Module

This procedure resets the TA and module (see Table 1-313).

**Table 1-313. Reset TA and Module**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Reset TA. | L4_TA_AGENT_CONTROL_L[0] OCP_RESET | 0x1 |
| Wait until target module clock = 16 cycles. | | |
| Write 0 to clear TA time - out status. | L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT | 0x0 |
| Write 0 to clear TA reset. | L4_TA_AGENT_CONTROL_L[0] OCP_RESET | 0x0 |
| Reset the attached module.(1) | | |

(1) For more information, see the respective module chapter.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 1.12.3.4.1.2.2 L4 Interconnect Time-Out Configuration Mode

##### 1.12.3.4.1.2.2.1 Main Sequence: L4 Interconnect Time-Out Configuration Mode

This procedure describes the time-out configuration sequence (see Table 1-314).

**Table 1-314. Time-Out Configuration**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Disable time-out. | L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE | 0x0 |
| Clear TA time-out error status.(1) | L4_TA_AGENT_STATUS_L[10:8] REQ_TIMEOUT | 0x1 |
| Set time-out at TA level.(1) | L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT | xxx |
| Set time-out base. | L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE | xxx |

(1) Must be done for each TA.

#### 1.12.3.4.1.2.3 L4 Interconnect Firewall Configuration Mode

##### 1.12.3.4.1.2.3.1 Main Sequence: L4 Interconnect Firewall Configuration Mode

This procedure describes the firewall configuration sequence (see Table 1-315).

**Table 1-315. Firewall Configuration**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Define the members of protection group k.(1) | L4_AP_PROT_GROUP_MEMBERS_k_L[15:0] CONNID_BIT_VECTOR | xxx |
| Define the access type of a protection group k.(1) | L4_AP_PROT_GROUP_ROLES_k_L[15:0] ENABLE | xx |
| Set region affiliation to protection group.(2) | L4_AP_REGION_l_L[22:20] PROT_GROUP_ID | xxx |

(1) Must be done for each protection group.
(2) Must be done for each region.


Copyright © 2011–2016, Texas Instruments Incorporated

### 1.12.3.5 L4 Interconnects Register Manual

Table 1-316 and Table 1-317 list all L4 register blocks for IA, TA, AP, and LA. Each module instance is shown with the module register mapping and bit and bit field definitions.

#### *1.12.3.5.1 L4 Interconnects Instance Summary*

**Table 1-316. L4LS Instance Summary**

| Module Name | L3 Base Address | Size |
|---|---|---|
| CENTAURUS_L4LS_AP | 0x4800 0000 | 2KBytes |
| L4LS_LA | 0x4800 0800 | 512Bytes |
| L4LS_IA_IP0 | 0x4800 1000 | 128Bytes |
| L4LS_TA_MMU | 0x4801 1000 | 64Bytes |
| L4LS_TA_UART0 | 0x4802 1000 | 64Bytes |
| L4LS_TA_UART1 | 0x4802 3000 | 64Bytes |
| L4LS_TA_UART2 | 0x4802 5000 | 64Bytes |
| L4LS_TA_I2C0 | 0x4802 9000 | 64Bytes |
| L4LS_TA_I2C1 | 0x4802 B000 | 64Bytes |
| L4LS_TA_DMTIMER1 | 0x4802 F000 | 64Bytes |
| L4LS_TA_SPI0 | 0x4803 1000 | 64Bytes |
| L4LS_TA_GPIO0 | 0x4803 3000 | 64Bytes |
| L4LS_TA_MCASP0 | 0x4803 A000 | 64Bytes |
| L4LS_TA_MCASP1 | 0x4803 E000 | 64Bytes |
| L4LS_TA_DMTIMER2 | 0x4804 1000 | 64Bytes |
| L4LS_TA_DMTIMER3 | 0x4804 3000 | 64Bytes |
| L4LS_TA_DMTIMER4 | 0x4804 5000 | 64Bytes |
| L4LS_TA_DMTIMER5 | 0x4804 7000 | 64Bytes |
| L4LS_TA_DMTIMER6 | 0x4804 9000 | 64Bytes |
| L4LS_TA_DMTIMER7 | 0x4804 B000 | 64Bytes |
| L4LS_TA_GPIO1 | 0x4804 D000 | 64Bytes |
| L4LS_TA_MCASP2 | 0x4805 2000 | 64Bytes |
| L4LS_TA_MMCHS0 | 0x4807 0000 | 64Bytes |
| L4LS_TA_ELM | 0x4809 0000 | 64Bytes |
| L4LS_TA_RTC | 0x480C 1000 | 64Bytes |
| L4LS_TA_MAILBOX | 0x480C 9000 | 64Bytes |
| L4LS_TA_SPINLOCK | 0x480C B000 | 64Bytes |
| L4LS_TA_HDVPSS | 0x4812 0000 | 64Bytes |
| L4LS_TA_HDMIPHY | 0x4812 3000 | 64Bytes |
| L4LS_TA_CONTROL | 0x4816 0000 | 64Bytes |
| L4LS_TA_PRCM | 0x4818 3000 | 64Bytes |
| L4LS_TA_SR0 | 0x4818 9000 | 64Bytes |
| L4LS_TA_SR1 | 0x4818 B000 | 64Bytes |
| L4LS_TA_OCPWP | 0x4818 D000 | 64Bytes |
| L4LS_TA_SR2 | 0x4819 1000 | 64Bytes |
| L4LS_TA_SR3 | 0x4819 3000 | 64Bytes |
| L4LS_TA_I2C2 | 0x4819 D000 | 64Bytes |
| L4LS_TA_I2C3 | 0x4819 F000 | 64Bytes |
| L4LS_TA_SPI1 | 0x481A 1000 | 64Bytes |
| L4LS_TA_SPI2 | 0x481A 3000 | 64Bytes |
| L4LS_TA_SPI3 | 0x481A 5000 | 64Bytes |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-316. L4LS Instance Summary (continued)**

| Module Name | L3 Base Address | Size |
|---|---|---|
| L4LS_TA_UART3 | 0x481A 7000 | 64Bytes |
| L4LS_TA_UART4 | 0x481A 9000 | 64Bytes |
| L4LS_TA_UART5 | 0x481A B000 | 64Bytes |
| L4LS_TA_GPIO2 | 0x481A D000 | 64Bytes |
| L4LS_TA_GPIO3 | 0x481A F000 | 64Bytes |
| L4LS_TA_DMTIMER8 | 0x481C 2000 | 64Bytes |
| L4LS_TA_PLLSS | 0x481C 6000 | 64Bytes |
| L4LS_TA_WDT0 | 0x481C 8000 | 64Bytes |
| L4LS_TA_PATA | 0x481C B000 | 64Bytes |
| L4LS_TA_DCAN0 | 0x481C E000 | 64Bytes |
| L4LS_TA_DCAN1 | 0x481D 2000 | 64Bytes |
| L4LS_TA_ATL | 0x481D 5000 | 64Bytes |
| L4LS_TA_FDIF | 0x481D 7000 | 64Bytes |
| L4LS_TA_MMCHS1 | 0x481E 8000 | 64Bytes |

**Table 1-317. L4HS Instance Summary**

| Module Name | L3 Base Address | Size |
|---|---|---|
| L4HS_AP | 0x4A00 0000 | 1KBytes |
| L4HS_LA | 0x4A00 0800 | 512Bytes |
| L4HS_IA_IP0 | 0x4A00 1000 | 128Bytes |
| L4HS_IA_IP1 | 0x4A00 1400 | 128Bytes |
| L4HS_TA_SW3PG | 0x4A10 8000 | 64Bytes |
| L4HS_TA_SATA | 0x4A15 0000 | 64Bytes |
| L4HS_TA_MLB | 0x4A1A 0000 | 64Bytes |
| L4HS_TA_MCASP3CFG | 0x4A1A 4000 | 64Bytes |
| L4HS_TA_MCASP3DAT | 0x4A1A 6000 | 64Bytes |
| L4HS_TA_MCASP4CFG | 0x4A1A A000 | 64Bytes |
| L4HS_TA_MCASP4DAT | 0x4A1A C000 | 64Bytes |
| L4HS_TA_MCASP5CFG | 0x4A1B 0000 | 64Bytes |
| L4HS_TA_MCASP5DAT | 0x4A1B 2000 | 64Bytes |
| L4HS_TA_VCP2 | 0x4A1B 4000 | 64Bytes |

#### *1.12.3.5.2 L4 Initiator Agent (L4 IA)*

##### *1.12.3.5.2.1 L4 Initiator Agent (L4 IA) Register Summary*

Table 1-318 summarizes the L4 IA register mapping.

**Table 1-318. IA Registers Mapping Summary**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_IA_0 L3 Physical Address | L4HS_IA_0 L3 Physical Address | |
|---|---|---|---|---|---|---|
| L4_IA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4800 1000 | 0x4A00 1000 | |
| L4_IA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4800 1004 | 0x4A00 1004 | |
| L4_IA_CORE_L | R | 32 | 0x0000 0018 | 0x4800 1018 | 0x4A00 1018 | |
| L4_IA_CORE_H | R | 32 | 0x0000 001C | 0x4800 0101C | 0x4A00 0101C | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-318. IA Registers Mapping Summary (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_IA_0 L3 Physical Address | L4HS_IA_0 L3 Physical Address | |
|---|---|---|---|---|---|---|
| L4_IA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4800 1020 | 0x4A00 1020 | |
| L4_IA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4800 1024 | 0x4A00 1024 | |
| L4_IA_AGENT_STATUS_L | RW | 32 | 0x0000 0028 | 0x4800 1028 | 0x4A00 1028 | |
| L4_IA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4800 102C | 0x4A00 102C | |
| L4_IA_ERROR_LOG_L | RW | 32 | 0x0000 0058 | 0x4800 1058 | 0x4A00 1058 | |
| L4_IA_ERROR_LOG_H | R | 32 | 0x0000 005C | 0x4800 105C | 0x4A00 105C | |
| L4_IA_ERROR_LOG_ADDR_L | R | 32 | 0x0000 0060 | 0x4800 1060 | 0x4A00 1060 | |
| L4_IA_ERROR_LOG_ADDR_H | R | 32 | 0x0000 0064 | 0x4800 1064 | 0x4A00 1064 | |

#### *1.12.3.5.2.2 L4 Initiator Agent (L4 IA) Register Description*

through describe the L4 IA registers.

**Table 1-319. L4_IA_COMPONENT_L**

| | |
|---|---|
| **Address Offset** | 0x0000 0000 |
| **Physical Address** | 0x4800 1000<br>0x4A00 1000 |
| **Description** | COMPONENT register identifies the component to which this register block belongs. The register contains a component code and revision, which are used to identify the hardware of the component. The COMPONENT register is read-only. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CODE | | | | | | | | | | | | | | | | REV | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | CODE | Interconnect code | R | See (1). |
| 15:0 | REV | Component revision code | R | See(1). |

(1) TI Internal Data


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-320. L4_IA_COMPONENT_H

| | |
|---|---|
| **Address Offset** | 0x0000 0004 |
| **Physical Address** | 0x4800 1004<br>0x4A00 1004 |
| **Description** | COMPONENT register identifies the component to which this register block belongs. The register contains a component code and revision, which are used to identify the hardware of the component. The COMPONENT register is read-only. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | Read returns 0. | R | 0x0000 |

### Table 1-321. L4_IA_CORE_L

| | |
|---|---|
| **Address Offset** | 0x0000 0018 |
| **Physical Address** | 0x4800 1018<br>0x4A00 1018 |
| **Description** | Provide information about the core initiator |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CORE_CODE | | | | | | | | | | | | | | | | CORE_REV | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | CORE_CODE | Interconnect core code | R | See (1). |
| 15:0 | CORE_REV | Component revision code code | R | See(1). |

(1) TI Internal Data

### Table 1-322. L4_IA_CORE_H

| | |
|---|---|
| **Address Offset** | 0x0000 001C |
| **Physical Address** | 0x4800 101C<br>0x4A00 101C |
| **Description** | Provide information about the core initiator |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | VENDOR_CODE | | | | | | | | | | | | | | | |

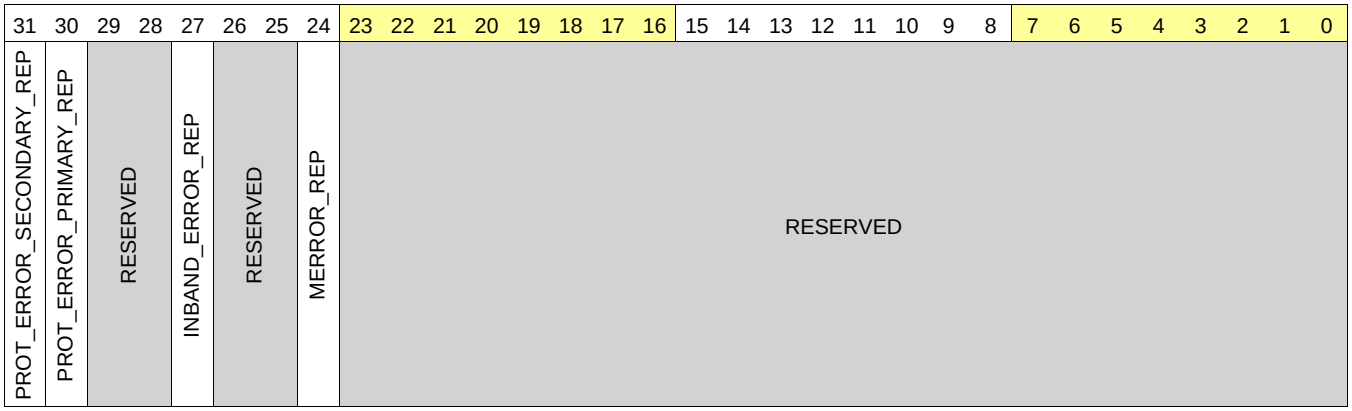
| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | RESERVED | Reserved | R | 0x0000 |
| 15:0 | VENDOR_CODE | Vendor revision core code | R | See (1). |

(1) TI Internal Data


Copyright © 2011–2016, Texas Instruments Incorporated

## Table 1-323. L4_IA_AGENT_CONTROL_L

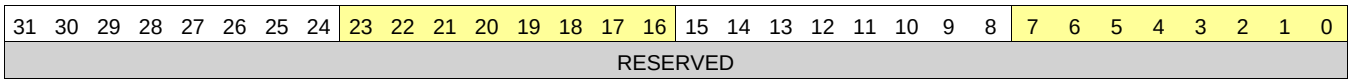
| | |
|---|---|
| **Address Offset** | 0x0000 0020 |
| **Physical Address** | 0x4800 1020<br>0x4A00 1020 |
| **Description** | Enable error reporting on an initiator interface.The error reporting mechanism is enabled when the INBAND_ERROR_REP bit field is set to 1. The out-of-band OCP MError reporting mechanism is enabled when the MERROR_REP bit field is set to 1. |
| **Type** | RW |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31 | PROT_ERROR_SECONDARY_REP | Out-of-band reporting of protection mechanism secondary errors | RW | 1 |
| 30 | PROT_ERROR_PRIMARY_REP | Out-of-band reporting of protection mechanism primary errors | RW | 1 |
| 29:28 | RESERVED | Read returns 0. | R | 0x0 |
| 27 | INBAND_ERROR_REP | Setting this field to 1 reports on in-band errors using the INBAND_ERROR log bit of IA.AGENT_STATUS register. | RW | 1 |
| 26:25 | RESERVED | Read returns 0. | R | 0x0 |
| 24 | MERROR_REP | OCP MError reporting control | R | 0 |
| 23:0 | RESERVED | Read returns 0. | R | 0x0 |

## Table 1-324. L4_IA_AGENT_CONTROL_H

| | |
|---|---|
| **Address Offset** | 0x0000 0024 |
| **Physical Address** | 0x4800 1024<br>0x4A00 1020 |
| **Description** | Enable error reporting on an initiator interface. |
| **Type** | R |

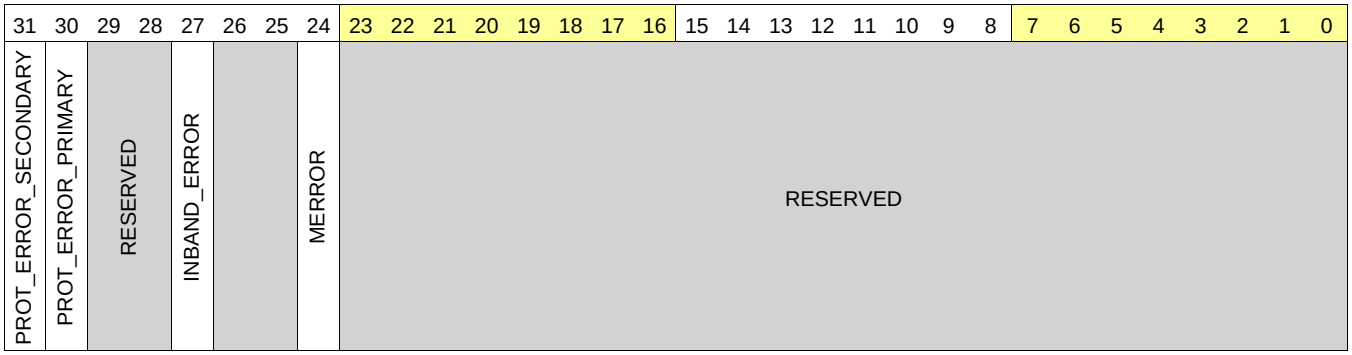



| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | Read returns 0. | R | 0x0000 0000 |


Copyright © 2011–2016, Texas Instruments Incorporated

## Table 1-325. L4_IA_AGENT_STATUS_L

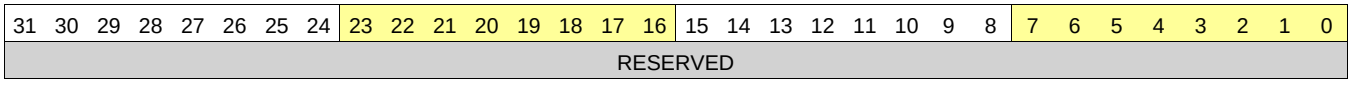
| | |
|---|---|
| **Address Offset** | 0x0000 0028 |
| **Physical Address** | 0x4800 1028<br>0x4A00 1028 |
| **Description** | Stores status information for an initiator. The INBAND_ERROR and MERROR fields are read/write and are implemented as log bits. |
| **Type** | RW |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 63:32 | RESERVED | Read returns 0s | R | 0x0000 0000 |
| 31 | PROT_ERROR_SECONDARY | Protection error status. MErrorSteer indicates a secondary error. | RW<br>W1toClr | 0 |
| 30 | PROT_ERROR_PRIMARY | Protection error status. MErrorSteer indicates a primary error (or is not enabled). | RW<br>W1toClr | 0 |
| 29:28 | RESERVED | Read returns 0s | R | 0x0 |
| 27 | INBAND_ERROR | Error status | RW<br>W1toClr | 0 |
| 26:25 | RESERVED | Read returns 0s | R | 0x0 |
| 24 | MERROR | Value of the OCP MError signal | R | 0 |
| 23:0 | RESERVED | Read returns 0s | R | 0x00 0000 |

## Table 1-326. L4_IA_AGENT_STATUS_H

| | |
|---|---|
| **Address Offset** | 0x0000 002C |
| **Physical Address** | 0x4800 102C<br>0x4A00 102C |
| **Description** | Stores status information for an initiator. |
| **Type** | R |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | Read returns 0. | R | 0x0000 0000 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-327. L4_IA_ERROR_LOG_L

| | |
|---|---|
| **Address Offset** | 0x0000 0058 |
| **Physical Address** | 0x4800 1058<br>0x4A00 1058 |
| **Description** | Log information about error conditions. The CODE field logs any protection violation or address hole errors detected by the initiator subsystem while decoding a request. |
| **Type** | RW |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MULTI | SECONDARY | RESERVED | | | | CODE | | RESERVED | | | | | | | | CONNID | | | | | | | | RESERVED | | | | | | CMD | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31 | MULTI | Multiple errors detected | RW<br>W1toClr | 0 |
| 30 | SECONDARY | Indicates whether protection violation was a primary or secondary error | RW<br>W1toClr | 0 |
| 29:26 | RESERVED | Read returns 0. | R | 0x0 |
| 25:24 | CODE | The error code of an initiator request.<br>0x00: No errors<br>0x01: Reserved<br>0x10: Address hole<br>0x11: Protection violation | RW<br>W1toClr | 0x0 |
| 23:14 | RESERVED | Read returns 0. | R | 0x000 |
| 13:8 | CONNID | ConnID of request causing the error, refer to Table 1-298 | R | 0x00 |
| 7:3 | RESERVED | Read returns 0. | R | 0x00 |
| 2:0 | CMD | Command that caused error | R | 0x0 |

### Table 1-328. L4_IA_ERROR_LOG_H

| | |
|---|---|
| **Address Offset** | 0x0000 005C |
| **Physical Address** | 0x4800 105C<br>0x4800 105C |
| **Description** | Log information about error conditions. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | REQ_INFO | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | RESERVED | Read returns 0. | R | 0x0000 |
| 15:0 | REQ_INFO | MReqInfo bits of request that caused the error<br>REQ_INFO[0] = supervisor,<br>REQ_INFO[1] = Debug | R | 0x0000 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-329. L4_IA_ERROR_LOG_ADDR_L**

| | |
|---|---|
| **Address Offset** | 0x0000 0060 |
| **Physical Address** | 0x4800 1060<br>0x4A00 1060 |
| **Description** | Extended error log (address information) |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | ADDR | Address of request that caused the error. N is the number MAddr bits. | R | 0x0000 0000 |

**Table 1-330. L4_IA_ERROR_LOG_ADDR_H**

| | |
|---|---|
| **Address Offset** | 0x0000 0064 |
| **Physical Address** | 0x4800 1064<br>0x4A00 1064 |
| **Description** | Extended error log (address information) |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | Read returns 0. | R | 0x0000 0000 |

### *1.12.3.5.3 L4 Target Agent (L4 TA)*

#### *1.12.3.5.3.1 L4 Target Agent (L4 TA) Register Summary*

Table 1-331 through Table 1-347 summarizes the L4 TA mapping of the CFG_TA, PER_TA, and WKUP_TA registers.

**Table 1-331. L4LS_TA Register Mapping Summary 1**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_MMU L3 Physical Address | L4LS_TA_UART0 L3 Physical Address | L4LS_TA_UART1 L3 Physical Address | L4LS_TA_UART2 L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4801 1000 | 0x4802 1000 | 0x4802 3000 | 0x4802 5000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4801 1004 | 0x4802 1004 | 0x4802 3004 | 0x4802 5004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4801 1018 | 0x4802 1018 | 0x4802 3018 | 0x4802 5018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4801 101C | 0x4802 101C | 0x4802 301C | 0x4802 501C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4801 1020 | 0x4802 1020 | 0x4802 3020 | 0x4802 5020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4801 1024 | 0x4802 1024 | 0x4802 3024 | 0x4802 5024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4801 1028 | 0x4802 1028 | 0x4802 3028 | 0x4802 5028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4801 102C | 0x4802 102C | 0x4802 302C | 0x4802 502C |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-332. L4LS_TA Register Mapping Summary 2**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_I2C0 L3 Physical Address | L4LS_TA_I2C1 L3 Physical Address | L4LS_TA_TIMER 1 L3 Physical Address |
|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4802 9000 | 0x4802 B000 | 0x4802 F000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4802 9004 | 0x4802 B004 | 0x4802 F004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4802 9018 | 0x4802 B018 | 0x4802 F018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4802 901C | 0x4802 B01C | 0x4802 F01C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4802 9020 | 0x4802 B020 | 0x4802 F020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4802 9024 | 0x4802 B024 | 0x4802 F024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4802 9028 | 0x4802 B028 | 0x4802 F028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4802 902C | 0x4802 B02C | 0x4802 F02C |

**Table 1-333. L4LS_TA Register Mapping Summary 3**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_SPI0 L3 Physical Address | L4LS_TA_GPIO0 L3 Physical Address | L4LS_TA_MCASP0 L3 Physical Address | L4LS_TA_MCASP1 L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4803 1000 | 0x4803 3000 | 0x4803 A000 | 0x4803 E000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4803 1004 | 0x4803 3004 | 0x4803 A004 | 0x4803 E004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4803 1018 | 0x4803 3018 | 0x4803 A018 | 0x4803 E018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4803 101C | 0x4803 301C | 0x4803 A01C | 0x4803 E01C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4803 1020 | 0x4803 3020 | 0x4803 A020 | 0x4803 E020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4803 1024 | 0x4803 3024 | 0x4803 A024 | 0x4803 E024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4803 1028 | 0x4803 3028 | 0x4803 A028 | 0x4803 E028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4803 102C | 0x4803 302C | 0x4803 A02C | 0x4803 E02C |

**Table 1-334. L4LS_TA Register Mapping Summary 4**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_TIMER2 L3 Physical Address | L4LS_TA_TIMER3 L3 Physical Address | L4LS_TA_TIMER4 L3 Physical Address | L4LS_TA_TIMER5 L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4804 1000 | 0x4804 3000 | 0x4804 5000 | 0x4804 7000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4804 1004 | 0x4804 3004 | 0x4804 5004 | 0x4804 7004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4804 1018 | 0x4804 3018 | 0x4804 5018 | 0x4804 7018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4804 101C | 0x4804 301C | 0x4804 501C | 0x4804 701C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4804 1020 | 0x4804 3020 | 0x4804 5020 | 0x4804 7020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4804 1024 | 0x4804 3024 | 0x4804 5024 | 0x4804 7024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4804 1028 | 0x4804 3028 | 0x4804 5028 | 0x4804 7028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4804 102C | 0x4804 302C | 0x4804 502C | 0x4804 702C |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-335. L4LS_TA Register Mapping Summary 5**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_TIMER6 L3 Physical Address | L4LS_TA_TIMER7 L3 Physical Address | L4LS_TA_GPIO1 L3 Physical Address | L4LS_TA_MCASP2 L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4804 9000 | 0x4804 B000 | 0x4804 D000 | 0x4805 2000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4804 9004 | 0x4804 B004 | 0x4804 D004 | 0x4805 2004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4804 9018 | 0x4804 B018 | 0x4804 D018 | 0x4805 2018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4804 901C | 0x4804 B01C | 0x4804 D01C | 0x4805 201C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4804 9020 | 0x4804 B020 | 0x4804 D020 | 0x4805 2020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4804 9024 | 0x4804 B024 | 0x4804 D024 | 0x4805 2024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4804 9028 | 0x4804 B028 | 0x4804 D028 | 0x4805 2028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4804 902C | 0x4804 B02C | 0x4804 D02C | 0x4805 202C |

**Table 1-336. L4LS_TA Register Mapping Summary 6**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_MMCHS0 L3 Physical Address | L4LS_TA_ELM L3 Physical Address | L4LS_TA_RTC L3 Physical Address | L4LS_TA_Mailbox L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4807 0000 | 0x4809 0000 | 0x480C 1000 | 0x480C 9000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4807 0004 | 0x4809 0004 | 0x480C 1004 | 0x480C 9004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4807 0018 | 0x4809 0018 | 0x480C 1018 | 0x480C 9018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4807 001C | 0x4809 001C | 0x480C 101C | 0x480C 901C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4807 0020 | 0x4809 0020 | 0x480C 1020 | 0x480C 9020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4807 0024 | 0x4809 0024 | 0x480C 1024 | 0x480C 9024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4807 0028 | 0x4809 0028 | 0x480C 1028 | 0x480C 9028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4807 002C | 0x4809 002C | 0x480C 102C | 0x480C 902C |

**Table 1-337. L4LS_TA Register Mapping Summary 7**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_Spinlcok L3 Physical Address | L4LS_TA_HDVPSS L3 Physical Address | L4LS_TA_HDMI L3 Physical Address | L4LS_TA_CONTROL L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x480C B000 | 0x4812 0000 | 0x4812 3000 | 0x4816 0000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x480C B004 | 0x4812 0004 | 0x4812 3004 | 0x4816 0004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x480C B018 | 0x4812 0018 | 0x4812 3018 | 0x4816 0018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x480C B01C | 0x4812 001C | 0x4812 301C | 0x4816 001C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x480C B020 | 0x4812 0020 | 0x4812 3020 | 0x4816 0020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x480C B024 | 0x4812 0024 | 0x4812 3024 | 0x4816 0024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x480C B028 | 0x4812 0028 | 0x4812 3028 | 0x4816 0028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x480C B02C | 0x4812 002C | 0x4812 302C | 0x4816 002C |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-338. L4LS_TA Register Mapping Summary 8**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_PRCM L3 Physical Address | L4LS_TA_SR0 L3 Physical Address | L4LS_TA_SR1 L3 Physical Address | L4LS_TA_OCPWP L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4818 3000 | 0x4818 9000 | 0x4818 B000 | 0x4818 D000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4818 3004 | 0x4818 9004 | 0x4818 B004 | 0x4818 D004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4818 3018 | 0x4818 9018 | 0x4818 B018 | 0x4818 D018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4818 301C | 0x4818 901C | 0x4818 B01C | 0x4818 D01C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4818 3020 | 0x4818 9020 | 0x4818 B020 | 0x4818 D020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4818 3024 | 0x4818 9024 | 0x4818 B024 | 0x4818 D024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4818 3028 | 0x4818 9028 | 0x4818 B028 | 0x4818 D028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4818 302C | 0x4818 902C | 0x4818 B02C | 0x4818 D02C |

**Table 1-339. L4LS_TA Register Mapping Summary 9**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_SR3 L3 Physical Address | L4LS_TA_SR3 L3 Physical Address | L4LS_TA_I2C2 L3 Physical Address | L4LS_TA_I2C3 L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4819 1000 | 0x4819 3000 | 0x4819 D000 | 0x4819 F000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4819 1004 | 0x4819 3004 | 0x4819 D004 | 0x4819 F004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4819 1018 | 0x4819 3018 | 0x4819 D018 | 0x4819 F018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4819 101C | 0x4819 301C | 0x4819 D01C | 0x4819 F01C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4819 1020 | 0x4819 3020 | 0x4819 D020 | 0x4819 F020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4819 1024 | 0x4819 3024 | 0x4819 D024 | 0x4819 F024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4819 1028 | 0x4819 3028 | 0x4819 D028 | 0x4819 F028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4819 102C | 0x4819 302C | 0x4819 D02C | 0x4819 F02C |

**Table 1-340. L4LS_TA Register Mapping Summary 10**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_SPI1 L3 Physical Address | L4LS_TA_SPI2 L3 Physical Address | L4LS_TA_SPI3 L3 Physical Address | L4LS_TA_UART3 L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x481A 1000 | 0x481A 3000 | 0x481A 5000 | 0x481A 7000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x481A 1004 | 0x481A 3004 | 0x481A 5004 | 0x481A 7004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x481A 1018 | 0x481A 3018 | 0x481A 5018 | 0x481A 7018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x481A 101C | 0x481A 301C | 0x481A 501C | 0x481A 701C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x481A 1020 | 0x481A 3020 | 0x481A 5020 | 0x481A 7020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x481A 1024 | 0x481A 3024 | 0x481A 5024 | 0x481A 7024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x481A 1028 | 0x481A 3028 | 0x481A 5028 | 0x481A 7028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x481A 102C | 0x481A 302C | 0x481A 502C | 0x481A 702C |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-341. L4LS_TA Register Mapping Summary 11**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_UART4 L3 Physical Address | L4LS_TA_UART5 L3 Physical Address | L4LS_TA_GPIO2 L3 Physical Address | L4LS_TA_GPIO3 L3 Physcial Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x481A 9000 | 0x481A B000 | 0x481A D000 | 0x481A F000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x481A 9004 | 0x481A B004 | 0x481A D004 | 0x481A F004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x481A 9018 | 0x481A B018 | 0x481A D018 | 0x481A F018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x481A 901C | 0x481A B01C | 0x481A D01C | 0x481A F01C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x481A 9020 | 0x481A B020 | 0x481A D020 | 0x481A F020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x481A 9024 | 0x481A B024 | 0x481A D024 | 0x481A F024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x481A 9028 | 0x481A B028 | 0x481A D028 | 0x481A F028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x481A 902C | 0x481A B02C | 0x481A D02C | 0x481A F02C |

**Table 1-342. L4LS_TA Register Mapping Summary 12**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_TIMER8 L3 Physical Address | L4LS_TA_SYNCTIMER32K L3 Physical Address | L4LS_TA_PLLSS L3 Physical Address | L4LS_TA_WDT0 L3 Physical Address |
|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x481C 2000 | 0x481C 4000 | 0x481C 6000 | 0x481C 8000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x481C 2004 | 0x481C 4004 | 0x481C 6004 | 0x481C 8004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x481C 2018 | 0x481C 4018 | 0x481C 6018 | 0x481C 8018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x481C 201C | 0x481C 401C | 0x481C 601C | 0x481C 801C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x481C 2020 | 0x481C 4020 | 0x481C 6020 | 0x481C 8020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x481C 2024 | 0x481C 4024 | 0x481C 6024 | 0x481C 8024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x481C 2028 | 0x481C 4028 | 0x481C 6028 | 0x481C 8028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x481C 202C | 0x481C 402C | 0x481C 602C | 0x481C 802C |

**Table 1-343. L4LS_TA Register Mapping Summary 13**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_DCAN0 L3 Physical Address | L4LS_TA_DCAN1 L3 Physical Address | L4LS_TA_ATL L3 Physical Address | L4LS_TA_FDIF L3 Physical Address | L4LS_TA_MMCHS1 L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x481C E000 | 0x481D 2000 | 0x481D 5000 | 0x481D 7000 | 0x481E 8000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x481C E004 | 0x481D 2004 | 0x481D 5004 | 0x481D 7004 | 0x481E 8004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x481C E018 | 0x481D 2018 | 0x481D 5018 | 0x481D 7018 | 0x481E 8018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x481C E01C | 0x481D 201C | 0x481D 501C | 0x481D 701C | 0x481E 801C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x481C E020 | 0x481D 2020 | 0x481D 5020 | 0x481D 7020 | 0x481E 8020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x481C E024 | 0x481D 2024 | 0x481D 5024 | 0x481D 7024 | 0x481E 8024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x481C E028 | 0x481D 2028 | 0x481D 5028 | 0x481D 7028 | 0x481E 8028 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-343. L4LS_TA Register Mapping Summary 13 (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_TA_DCAN0 L3 Physical Address | L4LS_TA_DCAN1 L3 Physical Address | L4LS_TA_ATL L3 Physical Address | L4LS_TA_FDIF L3 Physical Address | L4LS_TA_MMCHS1 L3 Physical Address |
|---|---|---|---|---|---|---|---|---|
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x481C E02C | 0x481D 202C | 0x481D 502C | 0x481D 702C | 0x481E 802C |

**Table 1-344. L4HS_TA Register Mapping Summary 1**

| Register Name | Type | Register Width (Bits) | Address Offset | L4HS_TA_SW3PG L3 Physical Address | L4HS_TA_SATA L3 Physical Address | |
|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4A10 8000 | 0x4A15 0000 | |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4A10 8004 | 0x4A15 0004 | |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4A10 8018 | 0x4A15 0018 | |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4A10 801C | 0x4A15 001C | |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4A10 8020 | 0x4A15 0020 | |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4A10 8024 | 0x4A15 0024 | |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4A10 8028 | 0x4A15 0028 | |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4A10 802C | 0x4A10 002C | |

**Table 1-345. L4HS_TA Register Mapping Summary 2**

| Register Name | Type | Register Width (Bits) | Address Offset | L4HS_TA_MLB L3 Physical Address | L4HS_TA_MCASP3CFG L3 Physical Address | L4HS_TA_MCASP3DAT L3 Physical Address |
|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4A1A 0000 | 0x4A1A 4000 | 0x4A1A 6000 |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4A1A 0004 | 0x4A1A 4004 | 0x4A1A 6004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4A1A 0018 | 0x4A1A 4018 | 0x4A1A 6018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4A1A 001C | 0x4A1A 401C | 0x4A1A 601C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4A1A 0020 | 0x4A1A 4020 | 0x4A1A 6020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4A1A 0024 | 0x4A1A 4024 | 0x4A1A 6024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4A1A 0028 | 0x4A1A 4028 | 0x4A1A 6028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4A1A 002C | 0x4A1A 402C | 0x4A1A 602C |

**Table 1-346. L4HS_TA Register Mapping Summary 3**

| Register Name | Type | Register Width (Bits) | Address Offset | L4HS_TA_MCASP4CFG L3 Physical Address | L4HS_TA_MCASP4DAT L3 Physical Address | L4HS_TA_MCASP5CFG L3 Physical Address |
|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4A1A A000 | 0x4A1A C000 | 0x4A1B 0000 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-346. L4HS_TA Register Mapping Summary 3 (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset | L4HS_TA_MCASP4CFG L3 Physical Address | L4HS_TA_MCASP4DAT L3 Physical Address | L4HS_TA_MCASP5CFG L3 Physical Address |
|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4A1A A004 | 0x4A1A C004 | 0x4A1B 0004 |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4A1A A018 | 0x4A1A C018 | 0x4A1B 0018 |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4A1A A01C | 0x4A1A C01C | 0x4A1B 001C |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4A1A A020 | 0x4A1A C020 | 0x4A1B 0020 |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4A1A A024 | 0x4A1A C024 | 0x4A1B 0024 |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4A1A A028 | 0x4A1A C028 | 0x4A1B 0028 |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4A1A A02C | 0x4A1A C02C | 0x4A1B 002C |

**Table 1-347. L4HS_TA Register Mapping Summary 4**

| Register Name | Type | Register Width (Bits) | Address Offset | L4HS_TA_MCASP5DAT L3 Physical Address | L4HS_TA_VCP L3 Physical Address | |
|---|---|---|---|---|---|---|
| L4_TA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4A1B 2000 | 0x4A1B 4000 | |
| L4_TA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4A1B 2004 | 0x4A1B 4004 | |
| L4_TA_CORE_L | R | 32 | 0x0000 0018 | 0x4A1B 2018 | 0x4A1B 4018 | |
| L4_TA_CORE_H | R | 32 | 0x0000 001C | 0x4A1B 201C | 0x4A1B 401C | |
| L4_TA_AGENT_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4A1B 2020 | 0x4A1B 4020 | |
| L4_TA_AGENT_CONTROL_H | R | 32 | 0x0000 0024 | 0x4A1B 2024 | 0x4A1B 4024 | |
| L4_TA_AGENT_STATUS_L | R | 32 | 0x0000 0028 | 0x4A1B 2028 | 0x4A1B 4028 | |
| L4_TA_AGENT_STATUS_H | R | 32 | 0x0000 002C | 0x4A1B 202C | 0x4A1B 402C | |

#### *1.12.3.5.3.2 L4 Target Agent (L4 TA) Register Description*

Table 1-348 through Table 1-355 describe the L4 TA registers.

**Table 1-348. L4_TA_COMPONENT_H**

| | |
|---|---|
| **Address Offset** | 0x0000 0000 |
| **Physical Address** | See Table 1-331 to Table 1-347. |
| **Description** | Contains a component code and revision. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | Read returns 0 | R | 0x0000 000 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-349. L4_TA_COMPONENT_L

| | |
|---|---|
| **Address Offset** | 0x0000 0004 |
| **Physical Address** | See Table 1-331 to Table 1-347. |
| **Description** | Contains a component code and revision. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CODE | | | | | | | | | | | | | | | | REV | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | CODE | Interconnect code. | R | See (1). |
| 15:0 | REV | Component revision code. | R | See (1). |

(1) TI Internal Data

### Table 1-350. L4_TA_CORE_L

| | |
|---|---|
| **Address Offset** | 0x0000 0018 |
| **Physical Address** | See Table 1-331 to Table 1-347. |
| **Description** | Contains a component code and revision. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CORE_CODE | | | | | | | | | | | | | | | | CORE_REV | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | CORE_CODE | Interconnect core code | R | See (1). |
| 15:0 | CORE_REV | Component revision code code | R | See (1). |

(1) TI Internal Data

### Table 1-351. L4_TA_CORE_H

| | |
|---|---|
| **Address Offset** | 0x0000 001C |
| **Physical Address** | See Table 1-331 to Table 1-347. |
| **Description** | Contains a component code and revision. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | VENDOR_CODE | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | RESERVED | Reserved | R | 0x0000 |
| 15:0 | VENDOR_CODE | Vendor revision core code | R | See (1). |

(1) TI Internal Data


Copyright © 2011–2016, Texas Instruments Incorporated

## Table 1-352. L4_TA_AGENT_CONTROL_L

| | |
|---|---|
| **Address Offset** | 0x0000 0020 |
| **Physical Address** | See Table 1-331 to Table 1-347. |
| **Description** | Enable error reporting |
| **Type** | RW |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | SERROR_REP | RESERVED | | | | | | | | | | | | | REQ_TIMEOUT | | | RESERVED | | | | | | | OCP_RESET |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:25 | RESERVED | Read returns 0. | R | 0x00 |
| 24 | SERROR_REP | Enable logging of error | R | 0x0 |
| 23:11 | RESERVED | Read returns 0. | UNDEFINED_TYPE_STRING | |
| 10:8 | REQ_TIMEOUT | Time-out Bound. Values are:<br>0 - No time-out<br>1 - 1x base cycles.<br>2 - 4x base cycles.<br>3 - 16x base cycles.<br>4 - 64x base cycles. | RW | 0x2 |
| 7:1 | RESERVED | Read returns 0. | R | 0x00 |
| 0 | OCP_RESET | The OCP_RESET field controls the OCP reset signal to the attached core. Setting this bit clears any pending transfers and resets the OCP interface. The bit must be cleared to deassert the OCP reset signal. When the software reset feature is available on a target agent, the target agent OCP must also have a reset signal directed to the target core. | RW | 0 |

## Table 1-353. L4_TA_AGENT_CONTROL_H

| | |
|---|---|
| **Address Offset** | 0x0000 0024 |
| **Physical Address** | See Table 1-331 to Table 1-347. |
| **Description** | Enable clock power management |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | EXT_CLOCK | RESERVED | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:9 | RESERVED | Read returns 0. | R | 0x000000 |
| 8 | EXT_CLOCK | When set to 1, the ext_clk_off_i signal on a target agent indicates when the target agent should shut off. | R | 0 |
| 7:0 | RESERVED | Read returns 0. | R | 0x00 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-354. L4_TA_AGENT_STATUS_L**

| | |
|---|---|
| **Address Offset** | 0x0000 0028 |
| **Physical Address** | See Table 1-331 to Table 1-347. |
| **Description** | Error reporting |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | SERROR | RESERVED | | | | | | | | | | | | | | | REQ_TIMEOUT | RESERVED | | | | | | | OCP_RESET |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:25 | RESERVED | Read returns 0. | R | 0x00 |
| 24 | SERROR | Value of OCP SError signal | R | 0x0 |
| 23:9 | RESERVED | Read returns 0. | R | 0x0000 |
| 8 | REQ_TIMEOUT | Time-out status:<br>0x0: No request time-out<br>0x1: A request time-out has occurred | R<br>1toCLR | 0 |
| 7:1 | RESERVED | Read returns 0. | R | 0x00 |
| 0 | OCP_RESET | | R | 0x0 |

**Table 1-355. L4_TA_AGENT_STATUS_H**

| | |
|---|---|
| **Address Offset** | 0x0000 002C |
| **Physical Address** | See Table 1-331 to Table 1-347. |
| **Description** | Error reporting |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | Read returns 0 | R | 0x0000 000 |

### 1.12.3.5.4 L4 Link Agent (L4 LA)

#### 1.12.3.5.4.1 L4 Link Agent (L4 LA) Register Summary

Table 1-356 summarizes the L4 LA register mapping.

**Table 1-356. LA Register Mapping Summary**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_LA L3 Physical Address | L4HS_LA L3 Physical Address |
|---|---|---|---|---|---|
| L4_LA_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4800 0800 | 0x4A00 0800 |
| L4_LA_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4800 0804 | 0x4A00 0804 |
| L4_LA_NETWORK_L | R | 32 | 0x0000 0010 | 0x4800 0810 | 0x4A00 0810 |
| L4_LA_NETWORK_H | R | 32 | 0x0000 0014 | 0x4800 0814 | 0x4A00 0814 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-356. LA Register Mapping Summary (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_LA L3 Physical Address | L4HS_LA L3 Physical Address |
|---|---|---|---|---|---|
| L4_LA_INITIATOR_INFO_L | R | 32 | 0x0000 0018 | 0x4800 0818 | 0x4A00 0818 |
| L4_LA_INITIATOR_INFO_H | R | 32 | 0x0000 001C | 0x4800 081C | 0x4A00 081C |
| L4_LA_NETWORK_CONTROL_L | RW | 32 | 0x0000 0020 | 0x4800 0820 | 0x4A00 0820 |
| L4_LA_NETWORK_CONTROL_H | RW | 32 | 0x0000 0024 | 0x4800 0824 | 0x4A00 0824 |
| L4_LA_FLAG_MASK_i_L [(1)] | RW | 32 | 0x0000 0100 + (0x20*i) | 0x4800 0900 + (0x20*i) | 0x4A00 0900 + (0x20*i) |
| L4_LA_FLAG_MASK_i_H [(1)] | RW | 32 | 0x0000 0104 + (0x20*i) | 0x4800 0904 + (0x20*i) | 0x4A00 0904 + (0x20*i) |
| L4_LA_FLAG_STATUS_i _L[(1)] | R | 32 | 0x0000 0110 + (0x20*I) | 0x4800 0910 + (0x20*i) | 0x4A00 0910 + (0x20*i) |
| L4_LA_FLAG_STATUS_i_H | R | 32 | 0x0000 0114 + (0x20*I) | 0x4800 0914 + (0x20*i) | 0x4A00 0914 + (0x20*i) |

(1) i = 0 to 1 for L4LS
i = 0 to 1 for L4HS

### *1.12.3.5.4.2 L4 Link Agent (L4 LA) Register Description*

Table 1-357 through Table 1-365 describe the L4 LA registers.

**Table 1-357. L4_LA_COMPONENT_L**

| | |
|---|---|
| **Address Offset** | 0x0000 0000 |
| **Physical Address** | See Table 1-356 |
| **Description** | Contain a component code and revision, which are used to identify the hardware of the component. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CODE | | | | | | | | | | | | | | | | REV | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | CODE | Interconnect code. | R | See [(1)]. |
| 15:0 | REV | Component revision code. | R | See [(1)]. |

(1) TI Internal Data

**Table 1-358. L4_LA_COMPONENT_H**

| | |
|---|---|
| **Address Offset** | 0x0000 0004 |
| **Physical Address** | See Table 1-356 |
| **Description** | Contain a component code and revision, which are used to identify the hardware of the component. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | Read returns 0. | R | 0x0000 0000 |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-359. L4_LA_NETWORK_L

| | |
|---|---|
| **Address Offset** | 0x0000 0010 |
| **Physical Address** | See Table 1-356 |
| **Description** | Identify the interconnect |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | Read returns 0 | R | 0x0000 0000 |

### Table 1-360. L4_LA_NETWORK_H

| | |
|---|---|
| **Address Offset** | 0x0000 0014 |
| **Physical Address** | See Table 1-356 |
| **Description** | Identify the interconnect |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | ID | The ID field uniquely identifies this interconnect. | R | 0x00000000 |

### Table 1-361. L4_LA_INITIATOR_INFO_L

| | |
|---|---|
| **Address Offset** | 0x0000 0018 |
| **Physical Address** | See Table 1-356 |
| **Description** | Contain initiator subsystem information. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | PROT_GROUPS | | | | NUMBER_REGIONS | | | | | | | | RESERVED | | | | | | | | | | | | | | SEGMENTS | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:28 | RESERVED | Read returns 0. | R | 0x0 |
| 27:24 | PROT_GROUPS | Number of protection group of in the current L4<br>0x0: No protection group<br>0x1: 1 protection group<br>0x2: 2 protection groups<br>....<br>0x8: 8 protection groups<br>0x9 to 0xF: Reserved | R | see Table 1-362 |


Copyright © 2011–2016, Texas Instruments Incorporated

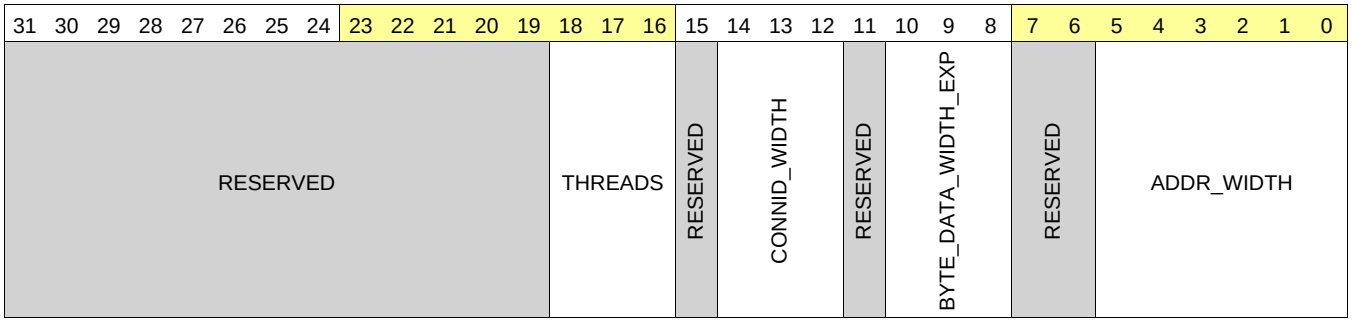
| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 23:16 | NUMBER_REGIONS | Number of regions in the current L4<br>0x0: Reserved<br>0x1: 1 region<br>0x2: 2 regions<br>....<br>Max regions +1 to 0xFF: Reserved, maximum regions is listed in Table 1-362 | R | see Table 1-362 |
| 15:4 | RESERVED | Read returns 0. | R | 0x000 |
| 3:0 | SEGMENTS | Number of segments in the current L4<br>0x0: Reserved<br>0x1: 1 segment<br>0x2: 2 segments<br>....<br>0x8: 8 segments | R | see Table 1-362 |

**Table 1-362. Reset value for L4_LA_INITIATOR_INFO_L**

| Field Name | L4LS | L4HS |
|---|---|---|
| PROT_GROUPS | 0x8 | 0x8 |
| NUMBER_REGIONS | 0x80 | 0x1D |
| SEGMENTS | 0x3 | 0x2 |

**Table 1-363. L4_LA_INITIATOR_INFO_H**

| | |
|---|---|
| **Address Offset** | 0x0000 001C |
| **Physical Address** | See Table 1-356 |
| **Description** | Contain initiator subsystem information. |
| **Type** | R |



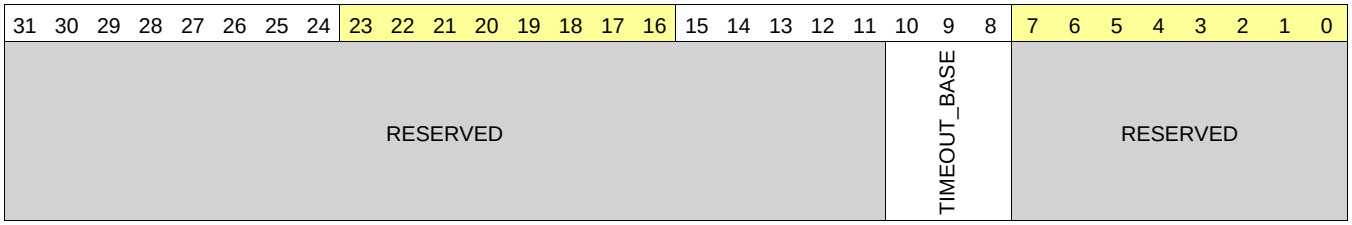

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:19 | RESERVED | Read returns 0. | R | 0x0000 |
| 18:16 | THREADS | The THREADS field specifies the number of initiator threads connected to the interconnect. The field contains read-only configuration information for the initiator subsystem. | R | 0x2 |
| 15 | RESERVED | Read returns 0. | R | 0 |
| 14:12 | CONNID_WIDTH | The initiator subsystem ConnID width. The CONNID_WIDTH field contains read-only configuration information for the initiator subsystem. | R | 0x4 |
| 11 | RESERVED | Read returns 0. | R | 0 |
| 10:8 | BYTE_DATA_WIDTH_EXP | This field specifies the initiator subsystem data width.The BYTE_DATA_WIDTH_EXP field contains read-only configuration information for the initiator subsystem.<br>0x1: 16-bit data width is specified<br>0x2: 32-bit data width is specified | R | 0x2 |
| 7:6 | RESERVED | Read returns 0. | R | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 5:0 | ADDR_WIDTH | This field specifies the initiator subsystem address width. The ADDR_WIDTH field contains read-only configuration information for the initiator subsystem. | R | 0x18 |

### Table 1-364. L4_LA_NETWORK_CONTROL_L

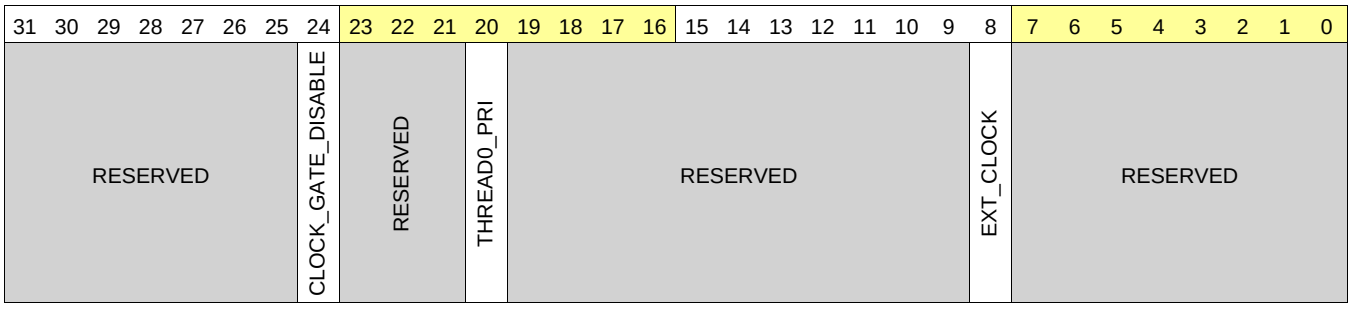
| | |
|---|---|
| **Address Offset** | 0x0000 0020 |
| **Physical Address** | See Table 1-356 |
| **Description** | Control interconnect minimum timeout values. |
| **Type** | RW |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:11 | RESERVED | Read returns 0. | R | 0x000000 |
| 10:8 | TIMEOUT_BASE | The TIMEOUT_BASE field indicates the time-out period (that is, base cycles) for the highest frequency time-base signal sent from the L4 initiator subsystem to all target agents that have time-out enabled. Values for the field are:<br>0 - Time-out disabled<br>1 - L4 interconnect clock cycles divided by 64<br>2 - L4 interconnect clock cycles divided by 256<br>3 - L4 interconnect clock cycles divided by 1024<br>4 - L4 interconnect clock cycles divided by 4096 | RW | 0x4 |
| 7:0 | RESERVED | Read returns 0. | R | 0x00 |

### Table 1-365. L4_LA_NETWORK_CONTROL_H

| | |
|---|---|
| **Address Offset** | 0x0000 0024 |
| **Physical Address** | See Table 1-356 |
| **Description** | Control interconnect global power control |
| **Type** | RW |




| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:25 | RESERVED | Read returns 0. | R | 0x00 |
| 24 | CLOCK_GATE_DISABLE | When set to 1 this field disables all clock gating. | RW | 0 |
| 23:21 | RESERVED | Read returns 0. | R | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 20 | THREAD0_PRI | Sets thread priority. If the field is set to 0, the default, all initiator threads are treated the same. Setting the THREAD0_PRI field to 1 assigns a higher arbitration priority to thread 0 of the first initiator OCP interface. To avoid starvation, arbitration is imposed by the initiator subsystem. When multiple requests from different initiator threads are dispatched to targets simultaneously, the oldest request is dispatched first. If thread 0 is assigned a higher priority, a request on thread 0 always wins arbitration. Assigning thread 0 of the first initiator OCP the highest priority on a request or response can result in the starvation of other threads. | R | 0 |
| 19:9 | RESERVED | Read returns 0. | R | 0x000 |
| 8 | EXT_CLOCK | When set to 1, the ext_clk_off_i signal on the initiator subsystem instructs the entire L4 to shut off. | R | 1 |
| 7:0 | RESERVED | Read returns 0. | R | 0x00 |

### Table 1-366. L4_LA_FLAG_MASK_i_L

| | |
|---|---|
| **Address Offset** | 0x0000 0100 + (0x20*i) |
| **Physical Address** | See Table 1-356 |
| **Description** | Mask of composite sideband flag(0) |
| **Type** | RW |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | MASK | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | RESERVED | | R | 0x0000 0000 0000 0000 |
| 1:0 | MASK | Number of input sideband signals | RW | 0x3 |

### Table 1-367. L4_LA_FLAG_MASK_i_H

| | |
|---|---|
| **Address Offset** | 0x0000 0104 + (0x20*i) |
| **Physical Address** | See Table 1-356 |
| **Description** | Mask of composite sideband flag(0) |
| **Type** | RW |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | RESERVED | R | 0x0000 0000 0000 0000 |

### Table 1-368. L4_LA_FLAG_STATUS_i_L

| | | | |
|---|---|---|---|
| **Address Offset** | 0x0000 0110 + (0x20*i) | | |
| **Physical Address** | See Table 1-356 | **Instance** | |
| **Description** | Status of composite sideband flag(0) | | |
| **Type** | R | | |


Copyright © 2011–2016, Texas Instruments Incorporated

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | STATUS | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:2 | RESERVED | | R | 0x0000 0000 0000 0000 |
| 1:0 | STATUS | Status of input sideband signals | R | 0x0 |

**Table 1-369. L4_LA_FLAG_STATUS_i_H**

| | |
|---|---|
| **Address Offset** | 0x0000 0114 + (0x20*i) |
| **Physical Address** | **Instance** |
| **Description** | Status of composite sideband flag(0) |
| **Type** | RW |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | RESERVED | R | 0x0000 0000 0000 0000 |

### *1.12.3.5.5 L4 Address Protection (L4 AP)*

#### *1.12.3.5.5.1 L4 Address Protection (L4 AP) Register Summary*

Table 1-370 summarizes the L4 AP register mapping.

**Table 1-370. L4 AP Register Summary**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_AP Physical Address | L4HS_AP Physical Address |
|---|---|---|---|---|---|
| L4_AP_COMPONENT_L | R | 32 | 0x0000 0000 | 0x4800 0 000 | 0x4A00 0 000 |
| L4_AP_COMPONENT_H | R | 32 | 0x0000 0004 | 0x4800 0004 | 0x4A00 0004 |
| L4_AP_SEGMENT_i_L (1) | RW | 32 | 0x0000 0100 + (0x08*i) | 0x4800 0100 + (0x08*i) | 0x4A00 0100 + (0x08*i) |
| L4_AP_SEGMENT_i_H (1) | RW | 32 | 0x0000 0104 + (0x08*i) | 0x4800 0104 + (0x08*i) | 0x4A00 0104 + (0x08*i) |
| L4_AP_PROT_GROUP_MEMBERS_k_L (2) | R | 32 | 0x0000 0200 + (0x08*k) | 0x4800 0200 + (0x08*k) | 0x4A00 0200 + (0x08*k) |
| L4_AP_PROT_GROUP_MEMBERS_k_H (2) | R | 32 | 0x0000 0204 + (0x08*k) | 0x4800 0204 + (0x08*k) | 0x4A00 0204 + (0x08*k) |
| L4_AP_PROT_GROUP_ROLES_k_L (2) | R | 32 | 0x0000 0280 + (0x08*k) | 0x4800 0280 + (0x08*k) | 0x4A00 0280 + (0x08*k) |
| L4_AP_PROT_GROUP_ROLES_k_H (2) | R | 32 | 0x0000 0284 + (0x08*k) | 0x4800 0284 + (0x08*k) | 0x4A00 0284 + (0x08*k) |
| L4_AP_REGION_l_L (3) | RW | 32 | 0x0000 0300 + (0x08*l) | 0x4800 0300 + (0x08*l) | 0x4A00 0300 + (0x08*l) |

(1) i = 0 to 2 for L4LS_AP
i = 0 to 1 for L4HS_AP
(2) k = 0 to 7 for L4LS_AP
k = 0 to 7 for L4HS_AP
(3) l = 0 to 127 for L4LS_AP
l = 0 to 28 for L4HS_AP


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-370. L4 AP Register Summary (continued)**

| Register Name | Type | Register Width (Bits) | Address Offset | L4LS_AP Physical Address | L4HS_AP Physical Address |
|---|---|---|---|---|---|
| L4_AP_REGION_I_H [3] | RW | 32 | 0x0000 0304 + (0x08*I) | 0x4800 0304 + (0x08*I) | 0x4A00 0304 + (0x08*I) |

### 1.12.3.5.5.2 L4 Address Protection (L4 AP) Register Description

Table 1-371 through Table 1-383 describe the L4 AP registers.

**Table 1-371. L4_AP_COMPONENT_L**

| | |
|---|---|
| **Address Offset** | 0x000 |
| **Physical address** | 0x4800 0000<br>0x4A00 0000 |
| **Description** | Contains a component code and revision, which are used to identify the hardware of the component. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CODE | | | | | | | | | | | | | | | | REV | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | CODE | Interconnect code | R | See [1]. |
| 15:0 | REV | Component revision code | R | See [1]. |

[1] TI Internal Data

**Table 1-372. L4_AP_COMPONENT_H**

| | |
|---|---|
| **Address Offset** | 0x004 |
| **Physical address** | 0x4800 0004<br>0x4A00 0004 |
| **Description** | Contains a component code and revision, which are used to identify the hardware of the component. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | RESERVED | Read returns 0 | R | 0x0000 0000 |

**Table 1-373. L4_AP_SEGMENT_i_L**

| | |
|---|---|
| **Address Offset** | 0x100 + (0x08*i) |
| **Physical address** | 0x4800 0100 + (0x08*i))<br>0x4A00 0100 + (0x08*i) |
| **Description** | Define the base address of each segments |
| **Type** | RW |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BASE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | BASE | The base address of the segment (with 0s from bit 0 to bit SIZE-1). | R | see Table 1-375 |

## Table 1-374. L4_AP_SEGMENT_i_H

| | |
|---|---|
| **Address Offset** | 0x104 + (0x08*i) |
| **Physical address** | 0x4800 0104 + (0x08*i)<br>0x4A00 0104 + (0x08*i) |
| **Description** | Define the size of each segments |
| **Type** | RW |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | SIZE | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:6 | RESERVED | Read returns 0. | R | 0x0000000 |
| 5:0 | SIZE | Segment size is a power of 2, where $2^{SIZE}$ is the byte size of a segment (all segment registers use the same size). | R | 0x14 |

## Table 1-375. Reset Value for L4_AP_SEGMENT_i

| i | L4LS | | L4HS | |
|---|---|---|---|---|
| | BASE | SIZE | BASE | SIZE |
| 0 | 0x0000 0000 | 0x14 | 0x0000 0000 | 0x14 |
| 1 | 0x0010 0000 | 0x14 | 0x0010 0000 | 0x14 |
| 2 | 0x0020 0000 | 0x14 | - | - |

## Table 1-376. L4_AP_PROT_GROUP_MEMBERS_k_L

| | | | |
|---|---|---|---|
| **Address Offset** | 0x200 + (0x08*k) | **Index** | |
| **Physical address** | 0x4800 0200 + (0x08*k)<br>0x4A00 0200 + (0x08*k) | | |
| **Description** | Define ConnID bit vectors for a protection group. | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | CONNID_BIT_VECTOR | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:16 | Reserved | Reserved | R | 0x000000000000 |
| 15:0 | CONNID_BIT_VECTOR | Specifies protection group members N is 2**W, where W is the connID width | RW | 0xFFFF |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-377. L4_AP_PROT_GROUP_MEMBERS_k_H

| | |
|---|---|
| **Address Offset** | 0x204 + (0x08*k) |
| **Physical address** | 0x4800 0204 + (0x08*k)<br>0x4A00 0204 + (0x08*k) |
| **Description** | Define ConnID bit vectors for a protection group. |
| **Type** | R |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | Reserved | Read returns 0's | R | 0x0000 0000 |

### Table 1-378. L4_AP_PROT_GROUP_ROLES_k_L

| | | | |
|---|---|---|---|
| **Address Offset** | 0x200 + (0x08*k) | **Index** | k = 0 to 7 for L4LS_AP<br>k = 0 to 7 for L4HS_AP |
| **Physical address** | 0x4800 0280 + (0x08*k)<br>0x4A00 0280 + (0x08*k) | | |
| **Description** | Define MReqInfo bit vectors for a protection group. | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ENABLE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | ENABLE | Enabled bits N is 2**W, where W (W is less than or equal to 6) is the number of MReqInfo bits configured for role checking | R | 0xFFFF FFFF |

### Table 1-379. L4_AP_PROT_GROUP_ROLES_k_H

| | | | |
|---|---|---|---|
| **Address Offset** | 0x204 + (0x08*k) | **Index** | k = 0 to 7 for L4LS_AP<br>k = 0 to 7 for L4HS_AP |
| **Physical address** | 0x4800 0284 + (0x08*k)<br>0x4A00 0284 + (0x08*k) | | |
| **Description** | Define ConnID bit vectors for a protection group. | | |
| **Type** | R | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ENABLE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:0 | ENABLE | Enabled bits N is 2**W, where W (W is less than or equal to 6) is the number of MReqInfo bits configured for role checking | R | 0xFFFF FFFF |


Copyright © 2011–2016, Texas Instruments Incorporated

### Table 1-380. L4_AP_REGION_I_L

| | |
|---|---|
| **Address Offset** | 0x300 + (0x08*I) |
| **Physical address** | 0x4800 0300 + (0x08*I)<br>0x4A00 0300 + (0x08*I) |
| **Description** | Define the base address of the region in respect to the segment it belongs to. |
| **Type** | RW |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | BASE | | | | | | | | | | | | | | | | | | | | | | | |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:20 | Reserved | Read returns 0. | R | 0x00 |
| 19:0 | BASE | Sets the base address of the region relative to its segment base. | R | See Table 1-382 to Table 1-383 |

### Table 1-381. L4_AP_REGION_I_H

| | |
|---|---|
| **Address Offset** | 0x304 + (0x08*I) |
| **Physical address** | 0x4800 0304 + (0x08*I)<br>0x4A00 0304 + (0x08*I) |
| **Description** | Define the size, protection group and segment ID of the region |
| **Type** | RW |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MADDRSPACE | | | | RESERVED | SEGMENT_ID | | | RESERVED | PROT_GROUP_ID | | | RESERVED | BYTE_DATA_WIDTH_EXP | | RESERVED | PHY_TARGET_ID | | | | | | | | RESERVED | SIZE | | | | | | ENABLE |

| Bits | Field Name | Description | Type | Reset |
|---|---|---|---|---|
| 31:28 | MADDRSPACE | Target OCP MAddrSpace value | R | See Table 1-382 and Table 1-383. |
| 27 | RESERVED | Read returns 0. | R | 0x0 |
| 26:24 | SEGMENT_ID | The segment associated to the region. | R | See Table 1-382 and Table 1-383. |
| 23 | RESERVED | Read returns 0. | R | 0 |
| 22:20 | PROT_GROUP_ID | The protection group associated to the region. | RW | See Table 1-382 and Table 1-383. |
| 19 | RESERVED | Read returns 0. | R | 0 |
| 18:17 | BYTE_DATA_WIDTH_EXP | The target data byte width is $2^{(BYTE_DATA_WIDTH_EXP)}$ bytes. | R | See Table 1-382 and Table 1-383. |
| 16:15 | Reserved | Read returns 0. | R | 0x0 |
| 14:8 | PHY_TARGET_ID | Physical target ID | R | See Table 1-382 and Table 1-383. |
| 7 | RESERVED | Read returns 0. | R | 0 |
| 6:1 | SIZE | Define the size of the region in bytes. $2^{SIZE}$ equals the region. | R | See Table 1-382 and Table 1-383. |
| 0 | ENABLE | 0x0: Disable the region, no access allows<br>0x1: Enable the region, with access as define in registers | R | 0x1 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-382. L4_AP_REGION_l Reset Value for L4HS**

| Region | BASE | ENABLE | SIZE | PHY_TARGET_ID | BYTE_DATA_WIDTH_EXP | PROT_GROUP_ID | SEGMENT_ID | MADDRSPACE |
|---|---|---|---|---|---|---|---|---|
| 0 | 0x0 0000 | 0x1 | 0x0B | 0x00 | 0x2 | 0x0 | 0x0 | 0x0 |
| 1 | 0x0 1000 | 0x1 | 0x0A | 0x00 | 0x2 | 0x7 | 0x0 | 0x1 |
| 2 | 0x0 0800 | 0x1 | 0x0B | 0x01 | 0x2 | 0x7 | 0x0 | 0x5 |
| 3 | 0x8 0000 | 0x1 | 0x12 | 0x04 | 0x2 | 0x7 | 0x0 | 0x0 |
| 4 | 0xC 0000 | 0x1 | 0x0C | 0x05 | 0x2 | 0x7 | 0x0 | 0x0 |
| 5 | 0x0 0000 | 0x1 | 0x0E | 0x06 | 0x2 | 0x7 | 0x1 | 0x1 |
| 6 | 0x0 8000 | 0x1 | 0x0C | 0x07 | 0x2 | 0x7 | 0x1 | 0x0 |
| 7 | 0x4 0000 | 0x1 | 0x10 | 0x10 | 0x2 | 0x7 | 0x1 | 0x0 |
| 8 | 0x5 0000 | 0x1 | 0x0C | 0x11 | 0x2 | 0x7 | 0x1 | 0x0 |
| 9 | 0x8 0000 | 0x1 | 0x11 | 0x18 | 0x2 | 0x7 | 0x1 | 0x0 |
| 10 | 0xA 0000 | 0x1 | 0x0C | 0x19 | 0x2 | 0x7 | 0x1 | 0x0 |
| 11 | 0x0 1400 | 0x1 | 0x0A | 0x00 | 0x2 | 0x7 | 0x0 | 0x2 |
| 12 | 0xA 4000 | 0x1 | 0x0C | 0x03 | 0x2 | 0x7 | 0x1 | 0x0 |
| 13 | 0xA 5000 | 0x1 | 0x0C | 0x0A | 0x2 | 0x7 | 0x1 | 0x0 |
| 14 | 0xA 6000 | 0x1 | 0x0C | 0x0B | 0x2 | 0x7 | 0x1 | 0x0 |
| 15 | 0xA A000 | 0x1 | 0x0C | 0x13 | 0x2 | 0x7 | 0x1 | 0x0 |
| 16 | 0xA B000 | 0x1 | 0x0C | 0x1A | 0x2 | 0x7 | 0x1 | 0x0 |
| 17 | 0xA C000 | 0x1 | 0x0C | 0x1B | 0x2 | 0x7 | 0x1 | 0x0 |
| 18 | 0xB 0000 | 0x1 | 0x0C | 0x23 | 0x2 | 0x7 | 0x1 | 0x0 |
| 19 | 0xB 1000 | 0x1 | 0x0C | 0x2A | 0x2 | 0x7 | 0x1 | 0x0 |
| 20 | 0xB 2000 | 0x1 | 0x0C | 0x2B | 0x2 | 0x7 | 0x1 | 0x0 |
| 21 | 0xB 3000 | 0x1 | 0x0C | 0x08 | 0x2 | 0x7 | 0x1 | 0x0 |
| 22 | 0xB 4000 | 0x1 | 0x0C | 0x09 | 0x2 | 0x7 | 0x1 | 0x0 |
| 23 | 0xB 5000 | 0x1 | 0x0C | 0x32 | 0x2 | 0x7 | 0x1 | 0x0 |
| 24 | 0xB 6000 | 0x1 | 0x0C | 0x33 | 0x2 | 0x7 | 0x1 | 0x0 |
| 25 | 0xA 2000 | 0x1 | 0x0D | 0x02 | 0x2 | 0x7 | 0x1 | 0x1 |
| 26 | 0xA 8000 | 0x1 | 0x0D | 0x12 | 0x2 | 0x7 | 0x1 | 0x1 |
| 27 | 0xA E000 | 0x1 | 0x0D | 0x22 | 0x2 | 0x7 | 0x1 | 0x1 |
| 28 | 0x0 4000 | 0x1 | 0x0E | 0x06 | 0x2 | 0x7 | 0x1 | 0x0 |

**Table 1-383. L4_AP_REGION_l Reset Value for L4 LS**

| Region | BASE | ENABLE | SIZE | PHY_TARGET_ID | BYTE_DATA_WIDTH_EXP | PROT_GROUP_ID | SEGMENT_ID | MADDRSPACE |
|---|---|---|---|---|---|---|---|---|
| 0 | 0x0 | 0x1 | 0x0B | 0x0 | 0x2 | 0x0 | 0x0 | 0x0 |
| 1 | 0x0 1000 | 0x1 | 0x0A | 0x0 | 0x2 | 0x7 | 0x0 | 0x1 |
| 2 | 0x0 0800 | 0x1 | 0x0B | 0x01 | 0x2 | 0x7 | 0x0 | 0x5 |
| 3 | 0x2 0000 | 0x1 | 0x0C | 0x12 | 0x2 | 0x7 | 0x0 | 0x0 |
| 4 | 0x2 2000 | 0x1 | 0x0C | 0x1A | 0x2 | 0x7 | 0x0 | 0x0 |
| 5 | 0x2 3000 | 0x1 | 0x0C | 0x1B | 0x2 | 0x7 | 0x0 | 0x0 |
| 6 | 0x2 4000 | 0x1 | 0x0C | 0x22 | 0x2 | 0x7 | 0x0 | 0x0 |
| 7 | 0x2 5000 | 0x1 | 0x0C | 0x23 | 0x2 | 0x7 | 0x0 | 0x0 |
| 8 | 0x2 8000 | 0x1 | 0x0C | 0x4A | 0x2 | 0x7 | 0x0 | 0x0 |
| 9 | 0x2 9000 | 0x1 | 0x0C | 0x4B | 0x2 | 0x7 | 0x0 | 0x0 |
| 10 | 0x2 A000 | 0x1 | 0x0C | 0x6C | 0x2 | 0x7 | 0x0 | 0x0 |
| 11 | 0x2 B000 | 0x1 | 0x0C | 0x6D | 0x2 | 0x7 | 0x0 | 0x0 |
| 12 | 0x2 C000 | 0x1 | 0x0C | 0x64 | 0x2 | 0x1 | 0x0 | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-383. L4_AP_REGION_l Reset Value for L4 LS (continued)**

| Region | BASE | ENABLE | SIZE | PHY_TARGET_ID | BYTE_DATA_WIDTH_EXP | PROT_GROUP_ID | SEGMENT_ID | MADDRSPACE |
|---|---|---|---|---|---|---|---|---|
| 13 | 0x2 D000 | 0x1 | 0x0C | 0x65 | 0x2 | 0x7 | 0x0 | 0x0 |
| 14 | 0x2 1000 | 0x1 | 0x0C | 0x13 | 0x2 | 0x7 | 0x0 | 0x0 |
| 15 | 0x2 E000 | 0x1 | 0x0C | 0x5C | 0x2 | 0x7 | 0x0 | 0x0 |
| 16 | 0x2 F000 | 0x1 | 0x0C | 0x5D | 0x2 | 0x7 | 0x0 | 0x0 |
| 17 | 0x3 0000 | 0x1 | 0x0C | 0x20 | 0x2 | 0x7 | 0x0 | 0x0 |
| 18 | 0x3 1000 | 0x1 | 0x0C | 0x21 | 0x2 | 0x7 | 0x0 | 0x0 |
| 19 | 0x3 2000 | 0x1 | 0x0C | 0x4E | 0x2 | 0x7 | 0x0 | 0x0 |
| 20 | 0x3 3000 | 0x1 | 0x0C | 0x4F | 0x2 | 0x7 | 0x0 | 0x0 |
| 21 | 0xC 2000 | 0x1 | 0x0C | 0x06 | 0x2 | 0x1 | 0x0 | 0x0 |
| 22 | 0xC 3000 | 0x1 | 0x0C | 0x07 | 0x2 | 0x7 | 0x0 | 0x0 |
| 23 | 0x6 0000 | 0x1 | 0x10 | 0x6E | 0x2 | 0x7 | 0x0 | 0x0 |
| 24 | 0x7 0000 | 0x1 | 0x0C | 0x6F | 0x2 | 0x7 | 0x0 | 0x0 |
| 25 | 0xC 0000 | 0x1 | 0x0C | 0x40 | 0x2 | 0x7 | 0x0 | 0x0 |
| 26 | 0xC 1000 | 0x1 | 0x0C | 0x41 | 0x2 | 0x7 | 0x0 | 0x0 |
| 27 | 0x8 0000 | 0x1 | 0x0D | 0x38 | 0x2 | 0x7 | 0x1 | 0x0 |
| 28 | 0x8 3000 | 0x1 | 0x0C | 0x39 | 0x2 | 0x7 | 0x1 | 0x0 |
| 29 | 0x3 C000 | 0x1 | 0x0D | 0x10 | 0x2 | 0x7 | 0x0 | 0x0 |
| 30 | 0x3 E000 | 0x1 | 0x0C | 0x11 | 0x2 | 0x7 | 0x0 | 0x0 |
| 31 | 0x5 0000 | 0x1 | 0x0D | 0x18 | 0x2 | 0x7 | 0x0 | 0x0 |
| 32 | 0x5 2000 | 0x1 | 0x0C | 0x19 | 0x2 | 0x7 | 0x0 | 0x0 |
| 33 | 0x3 8000 | 0x1 | 0x0D | 0x08 | 0x2 | 0x7 | 0x0 | 0x0 |
| 34 | 0x3 A000 | 0x1 | 0x0C | 0x09 | 0x2 | 0x7 | 0x0 | 0x0 |
| 35 | 0x8 E000 | 0x1 | 0x0C | 0x26 | 0x2 | 0x7 | 0x1 | 0x0 |
| 36 | 0x8 F000 | 0x1 | 0x0C | 0x27 | 0x2 | 0x7 | 0x1 | 0x0 |
| 37 | 0x8 C000 | 0x1 | 0x0C | 0x1C | 0x2 | 0x7 | 0x1 | 0x0 |
| 38 | 0x8 D000 | 0x1 | 0x0C | 0x1D | 0x2 | 0x7 | 0x1 | 0x0 |
| 39 | 0x0 1400 | 0x1 | 0x0A | 0x00 | 0x2 | 0x7 | 0x0 | 0x2 |
| 40 | 0x0 8000 | 0x1 | 0x0C | 0x24 | 0x2 | 0x2 | 0x0 | 0x0 |
| 41 | 0x0 9000 | 0x1 | 0x0C | 0x25 | 0x2 | 0x7 | 0x0 | 0x0 |
| 42 | 0x1 0000 | 0x1 | 0x0C | 0x6A | 0x2 | 0x7 | 0x0 | 0x0 |
| 43 | 0x1 1000 | 0x1 | 0x0C | 0x6B | 0x2 | 0x7 | 0x0 | 0x0 |
| 44 | 0x3 4000 | 0x1 | 0x0C | 0x28 | 0x2 | 0x1 | 0x0 | 0x0 |
| 45 | 0x3 5000 | 0x1 | 0x0C | 0x29 | 0x2 | 0x7 | 0x0 | 0x0 |
| 46 | 0x4 0000 | 0x1 | 0x0C | 0x2E | 0x2 | 0x7 | 0x0 | 0x0 |
| 47 | 0x4 1000 | 0x1 | 0x0C | 0x2F | 0x2 | 0x7 | 0x0 | 0x0 |
| 48 | 0x4 2000 | 0x1 | 0x0C | 0x36 | 0x2 | 0x7 | 0x0 | 0x0 |
| 49 | 0x4 3000 | 0x1 | 0x0C | 0x37 | 0x2 | 0x7 | 0x0 | 0x0 |
| 50 | 0x4 4000 | 0x1 | 0x0C | 0x3E | 0x2 | 0x7 | 0x0 | 0x0 |
| 51 | 0x4 5000 | 0x1 | 0x0C | 0x3F | 0x2 | 0x7 | 0x0 | 0x0 |
| 52 | 0x4 6000 | 0x1 | 0x0C | 0x46 | 0x2 | 0x7 | 0x0 | 0x0 |
| 53 | 0x4 7000 | 0x1 | 0x0C | 0x47 | 0x2 | 0x7 | 0x0 | 0x0 |
| 54 | 0x4 8000 | 0x1 | 0x0C | 0x70 | 0x2 | 0x7 | 0x0 | 0x0 |
| 55 | 0x4 9000 | 0x1 | 0x0C | 0x71 | 0x2 | 0x7 | 0x0 | 0x0 |
| 56 | 0x4 A000 | 0x1 | 0x0C | 0x78 | 0x2 | 0x7 | 0x0 | 0x0 |
| 57 | 0x4 B000 | 0x1 | 0x0C | 0x79 | 0x2 | 0x7 | 0x0 | 0x0 |
| 58 | 0x4 C000 | 0x1 | 0x0C | 0x56 | 0x2 | 0x7 | 0x0 | 0x0 |
| 59 | 0x4 D000 | 0x1 | 0x0C | 0x57 | 0x2 | 0x7 | 0x0 | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-383. L4_AP_REGION_l Reset Value for L4 LS (continued)**

| Region | BASE | ENABLE | SIZE | PHY_TARGET_ID | BYTE_DATA_WIDTH_EXP | PROT_GROUP_ID | SEGMENT_ID | MADDRSPACE |
|---|---|---|---|---|---|---|---|---|
| 60 | 0x8 0000 | 0x1 | 0x10 | 0x72 | 0x2 | 0x7 | 0x0 | 0x1 |
| 61 | 0x9 0000 | 0x1 | 0x0C | 0x73 | 0x2 | 0x7 | 0x0 | 0x0 |
| 62 | 0xA 0000 | 0x1 | 0x10 | 0x58 | 0x2 | 0x7 | 0x0 | 0x0 |
| 63 | 0xB 0000 | 0x1 | 0x0C | 0x59 | 0x2 | 0x7 | 0x0 | 0x0 |
| 64 | 0xC 8000 | 0x1 | 0x0C | 0x14 | 0x2 | 0x7 | 0x0 | 0x0 |
| 65 | 0xC 9000 | 0x1 | 0x0C | 0x15 | 0x2 | 0x7 | 0x0 | 0x0 |
| 66 | 0xC A000 | 0x1 | 0x0C | 0x48 | 0x2 | 0x7 | 0x0 | 0x0 |
| 67 | 0xC B000 | 0x1 | 0x0C | 0x49 | 0x2 | 0x7 | 0x0 | 0x0 |
| 68 | 0x8 2000 | 0x1 | 0x0C | 0x38 | 0x2 | 0x7 | 0x1 | 0x1 |
| 69 | 0x8 8000 | 0x1 | 0x0C | 0x3C | 0x2 | 0x7 | 0x1 | 0x0 |
| 70 | 0x8 9000 | 0x1 | 0x0C | 0x3D | 0x2 | 0x7 | 0x1 | 0x0 |
| 71 | 0x8 A000 | 0x1 | 0x0C | 0x44 | 0x2 | 0x7 | 0x1 | 0x0 |
| 72 | 0x8 B000 | 0x1 | 0x0C | 0x45 | 0x2 | 0x7 | 0x1 | 0x0 |
| 73 | 0x0 0000 | 0x1 | 0x11 | 0x04 | 0x2 | 0x7 | 0x1 | 0x0 |
| 74 | 0x2 0000 | 0x1 | 0x0C | 0x05 | 0x2 | 0x7 | 0x1 | 0x0 |
| 75 | 0x2 2000 | 0x1 | 0x0C | 0x0C | 0x2 | 0x7 | 0x1 | 0x0 |
| 76 | 0x2 3000 | 0x1 | 0x0C | 0x0D | 0x2 | 0x7 | 0x1 | 0x0 |
| 77 | 0x9 0000 | 0x1 | 0x0C | 0x4C | 0x2 | 0x7 | 0x1 | 0x0 |
| 78 | 0x9 1000 | 0x1 | 0x0C | 0x4D | 0x2 | 0x7 | 0x1 | 0x0 |
| 79 | 0x9 2000 | 0x1 | 0x0C | 0x54 | 0x2 | 0x7 | 0x1 | 0x0 |
| 80 | 0x9 3000 | 0x1 | 0x0C | 0x55 | 0x2 | 0x7 | 0x1 | 0x0 |
| 81 | 0x9 C000 | 0x1 | 0x0C | 0x52 | 0x2 | 0x7 | 0x1 | 0x0 |
| 82 | 0x9 D000 | 0x1 | 0x0C | 0x53 | 0x2 | 0x7 | 0x1 | 0x0 |
| 83 | 0x9 E000 | 0x1 | 0x0C | 0x5A | 0x2 | 0x7 | 0x1 | 0x0 |
| 84 | 0x9 F000 | 0x1 | 0x0C | 0x5B | 0x2 | 0x7 | 0x1 | 0x0 |
| 85 | 0xA 0000 | 0x1 | 0x0C | 0x42 | 0x2 | 0x7 | 0x1 | 0x0 |
| 86 | 0xA 4000 | 0x1 | 0x0C | 0x16 | 0x2 | 0x7 | 0x1 | 0x0 |
| 87 | 0xA 5000 | 0x1 | 0x0C | 0x17 | 0x2 | 0x7 | 0x1 | 0x0 |
| 88 | 0xA C000 | 0x1 | 0x0C | 0x5E | 0x2 | 0x7 | 0x1 | 0x0 |
| 89 | 0xA D000 | 0x1 | 0x0C | 0x5F | 0x2 | 0x7 | 0x1 | 0x0 |
| 90 | 0xA E000 | 0x1 | 0x0C | 0x66 | 0x2 | 0x7 | 0x1 | 0x0 |
| 91 | 0xA F000 | 0x1 | 0x0C | 0x67 | 0x2 | 0x7 | 0x1 | 0x0 |
| 92 | 0xD 4000 | 0x1 | 0x0C | 0x50 | 0x2 | 0x7 | 0x1 | 0x0 |
| 93 | 0xD 5000 | 0x1 | 0x0C | 0x51 | 0x2 | 0x7 | 0x1 | 0x0 |
| 94 | 0xA 6000 | 0x1 | 0x0C | 0x2A | 0x2 | 0x7 | 0x1 | 0x0 |
| 95 | 0xA 7000 | 0x1 | 0x0C | 0x2B | 0x2 | 0x7 | 0x1 | 0x0 |
| 96 | 0xA 8000 | 0x1 | 0x0C | 0x32 | 0x2 | 0x7 | 0x1 | 0x0 |
| 97 | 0xA 9000 | 0x1 | 0x0C | 0x33 | 0x2 | 0x7 | 0x1 | 0x0 |
| 98 | 0xA A000 | 0x1 | 0x0C | 0x3A | 0x2 | 0x7 | 0x1 | 0x0 |
| 99 | 0xA B000 | 0x1 | 0x0C | 0x3B | 0x2 | 0x7 | 0x1 | 0x0 |
| 100 | 0x3 6000 | 0x1 | 0x0C | 0x30 | 0x2 | 0x1 | 0x0 | 0x0 |
| 101 | 0x3 7000 | 0x1 | 0x0C | 0x31 | 0x2 | 0x7 | 0x0 | 0x0 |
| 102 | 0x4 0000 | 0x1 | 0x11 | 0x2C | 0x2 | 0x7 | 0x1 | 0x0 |
| 103 | 0x6 0000 | 0x1 | 0x0C | 0x2D | 0x2 | 0x7 | 0x1 | 0x0 |
| 104 | 0xA 1000 | 0x1 | 0x0C | 0x43 | 0x2 | 0x7 | 0x1 | 0x0 |
| 105 | 0xA 2000 | 0x1 | 0x0C | 0x0E | 0x2 | 0x7 | 0x1 | 0x1 |
| 106 | 0xA 3000 | 0x1 | 0x0C | 0x0F | 0x2 | 0x7 | 0x1 | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 1-383. L4_AP_REGION_l Reset Value for L4 LS (continued)**

| Region | BASE | ENABLE | SIZE | PHY_TARGET_ID | BYTE_DATA_WIDTH_EXP | PROT_GROUP_ID | SEGMENT_ID | MADDRSPACE |
|---|---|---|---|---|---|---|---|---|
| 107 | 0xB 0000 | 0x1 | 0x10 | 0x60 | 0x2 | 0x7 | 0x1 | 0x0 |
| 108 | 0xC 0000 | 0x1 | 0x0C | 0x61 | 0x2 | 0x7 | 0x1 | 0x0 |
| 109 | 0xC A000 | 0x1 | 0x0C | 0x1E | 0x2 | 0x7 | 0x1 | 0x0 |
| 110 | 0xC B000 | 0x1 | 0x0C | 0x1F | 0x2 | 0x7 | 0x1 | 0x0 |
| 111 | 0xC C000 | 0x1 | 0x0D | 0x02 | 0x2 | 0x7 | 0x1 | 0x0 |
| 112 | 0xC E000 | 0x1 | 0x0D | 0x03 | 0x2 | 0x7 | 0x1 | 0x0 |
| 113 | 0xD 0000 | 0x1 | 0x0D | 0x0A | 0x2 | 0x7 | 0x1 | 0x0 |
| 114 | 0xD 2000 | 0x1 | 0x0D | 0x0B | 0x2 | 0x7 | 0x1 | 0x0 |
| 115 | 0xD 6000 | 0x1 | 0x0C | 0x34 | 0x2 | 0x7 | 0x1 | 0x0 |
| 116 | 0xD 7000 | 0x1 | 0x0C | 0x35 | 0x2 | 0x7 | 0x1 | 0x0 |
| 117 | 0xD 8000 | 0x1 | 0x0F | 0x76 | 0x2 | 0x7 | 0x1 | 0x0 |
| 118 | 0xE 8000 | 0x1 | 0x0C | 0x77 | 0x2 | 0x7 | 0x1 | 0x0 |
| 119 | 0xE 0000 | 0x1 | 0x0F | 0x76 | 0x2 | 0x7 | 0x1 | 0x1 |
| 120 | 0xC 1000 | 0x1 | 0x0C | 0x68 | 0x2 | 0x7 | 0x1 | 0x0 |
| 121 | 0xC 2000 | 0x1 | 0x0C | 0x69 | 0x2 | 0x7 | 0x1 | 0x0 |
| 122 | 0xC 3000 | 0x1 | 0x0C | 0x62 | 0x2 | 0x7 | 0x1 | 0x0 |
| 123 | 0xC 4000 | 0x1 | 0x0C | 0x62 | 0x2 | 0x7 | 0x1 | 0x1 |
| 124 | 0xC 5000 | 0x1 | 0x0C | 0x74 | 0x2 | 0x7 | 0x1 | 0x0 |
| 125 | 0xC 6000 | 0x1 | 0x0C | 0x75 | 0x2 | 0x7 | 0x1 | 0x0 |
| 126 | 0xC 7000 | 0x1 | 0x0C | 0x7E | 0x2 | 0x7 | 0x1 | 0x0 |
| 127 | 0xC 8000 | 0x1 | 0x0C | 0x7F | 0x2 | 0x7 | 0x1 | 0x0 |


Copyright © 2011–2016, Texas Instruments Incorporated

# Power, Reset, and Clock Management (PRCM) Module

This chapter discusses the function of the power, reset, and clock management module.

**Topic** **Page**


2.1 Introduction ..... 463
2.2 Power Reset Clock Management Structure ..... 472
2.3 Clock Generation and Management ..... 474
2.4 Oscillator ..... 490
2.5 DPLLS ..... 491
2.6 DPLLLJ ..... 493
2.7 Reset Management ..... 499
2.8 Voltage and Power Domains ..... 512
2.9 Voltage Management Functional Description ..... 517
2.10 PRCM Registers ..... 518



Copyright © 2011–2016, Texas Instruments Incorporated

## 2.1 Introduction

The device power-management architecture ensures maximum performance and operation time for user satisfaction (audio/video support) while offering versatile power-management techniques for maximum design flexibility, depending on application requirements. This section contains the following information:

- Power-management architecture building blocks for the device.
- State-of-the-art power-management techniques supported by the power-management architecture of the device.

### *2.1.1 Device Power-Management Architecture Building Blocks*

To provide a versatile architecture supporting multiple power-management techniques, the power-management framework is built with three levels of resource management: clock, power, and voltage management.

These management levels are enforced by defining the managed entities or building blocks of the power-management architecture, called the clock, power, and voltage domains. A domain is a group of modules or subsections of the device that share a common entity (for example, common clock source, common voltage source, or common power switch). The group forming the domain is managed by a policy manager. For example, a clock for a clock domain is managed by a dedicated clock manager within the power, reset, and clock management (PRCM) module. The clock manager considers the joint clocking constraints of all the modules belonging to that clock domain (and, hence, receiving that clock).

#### 2.1.1.1 Clock Management

The PRCM module manages the gating (that is, switching off) and enabling of the clocks to the device modules. The clocks are managed based on the requirement constraints of the associated modules. The following sections identify the module clock characteristics, management policy, clock domains, and clock domain management.

Each module within the device has specific clock input characteristic requirements. Based on the characteristics of the clocks delivered to the modules, the clocks are divided into two categories: interface clocks and functional clocks.

The interface clocks have the following characteristics:

- They ensure proper communication between any module/subsystem and the interconnect.
- In most cases, they supply the system interconnect interface and registers of the module.
- A typical module has one interface clock, but modules with multiple interface clocks may also exist (that is, when connected to multiple interconnect buses).
- Interface clock management is done at the device level.
- From the standpoint of the PRCM module, an interface clock is identified by an _ICLK suffix.

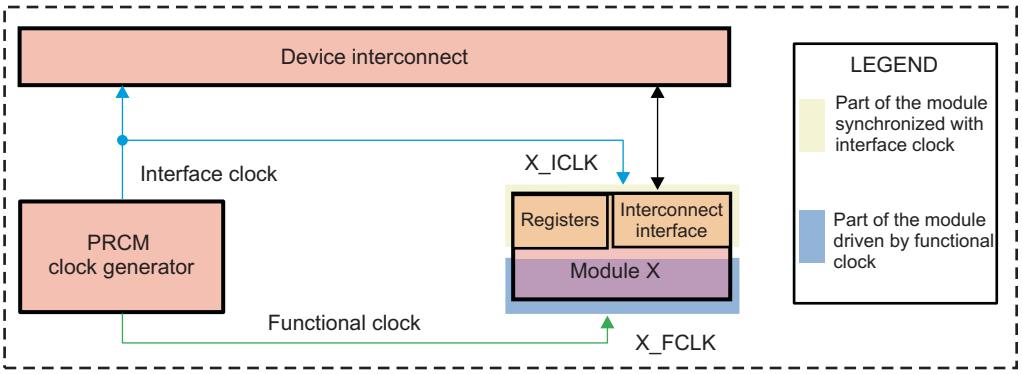
Functional clocks have the following characteristics:

- They supply the functional part of a module or subsystem.
- A module can have one or more functional clocks. Some functional clocks are mandatory, while others are optional. A module needs its mandatory clock(s) to be operational. The optional clocks are used for specific features and can be shut down without stopping the module activity.
- From the standpoint of the PRCM module, a functional clock is distributed directly to the related modules through a dedicated clock tree. It is identified with an _FCLK suffix.

Figure 2-1 provides a representation of the functional and interface clocks.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-1. Functional and Interface Clocks**




### 2.1.2 Module-Level Clock Management

Each module in the device may also have specific clock requirements. Certain module clocks must be active when operating in specific modes, or may be gated otherwise. Globally, the activation and gating of the module clocks are managed by the PRCM module. Hence, the PRCM module must be aware of when to activate and when to gate the module clocks. The PRCM module differentiates the clock-management behavior for device modules based on whether the module can initiate transactions on the device interconnect (called master module or initiators) or cannot initiate transactions and only responds to the transactions initiated by the master (called slave module or targets). Thus, two hardware-based power-management protocols are used:

- Master standby protocol: Clock-management protocol between the PRCM and master modules
- Slave idle protocol: Clock-management protocol between the PRCM and slave modules

#### 2.1.2.1 Master Standby Protocol

This protocol is used to indicate that a master module must initiate a transaction on the device interconnect and requests specific (functional and interface) clocks for the purpose. The PRCM module ensures that the required clocks are active when the master module requests the PRCM module to enable them. This is called a module wake-up transition and the module is said to be functional after this transition completes. Similarly, when the master module no longer requires the clocks, it informs the PRCM module, which can then gate the clocks to the module. The master module is then said to be in standby mode. Although the protocol is completely hardware-controlled, software must configure the clock-management behavior for the module. This is done by setting the module register bit field <Module>_SYSCONFIG. IDLEMODE or <Module>_SYSCONFIG. STANDBYMODE (see Table 2-1). The behavior, identified by standby mode values, must be configured.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-1. Master Module Standby-Mode Settings**

| Standby Mode Value | Selected Mode | Description |
|---|---|---|
| 0 | Force-standby | The module unconditionally asserts the standby request to the PRCM module, regardless of its internal operations. The PRCM module may gate the functional and interface clocks to the module. This mode must be used carefully because it does not prevent the loss of data at the time the clocks are gated. |
| 1h | No-standby | The module never asserts the standby request to the PRCM module. This mode is safe from a module point of view because it ensures that the clocks remain active. However, it is not efficient from a power-saving perspective because it never allows the output clocks of the PRCM module to be gated. |
| 2h | Smart-standby | The module asserts the standby request based on its internal activity status.The standby signal is asserted only when all ongoing transactions are complete and the module is idled. The PRCM module can then gate the clocks to the module. |
| 3h | Smart-standby wakeup-capable mode | The module asserts the standby request based on its internal activity status. The standby signal is asserted only when all ongoing transactions are complete and the module is idle. The PRCM module can then gate the clocks to the module. The module may generate (master-related) wake-up events when in STANDBY state. The mode is relevant only if the appropriate module mwakeup output is implemented. |

The standby status of a master module is indicated by the CM_<Power_domain>_<Module>_CLKCTRL[x].STBYST bit in the PRCM module. Table 2-2 describes the master module standby status.

**Table 2-2. Master Module Standby Status**

| STBYST Bit Value | Description |
|---|---|
| 0 | The module is functional. |
| 1 | The module is in standby mode. |

### 2.1.2.2 Slave Idle Protocol

This hardware protocol allows the PRCM module to control the state of a slave module. The PRCM module informs the slave module, through assertion of an idle request, when its clocks (interface and functional) can be gated. The slave can then acknowledge the request from the PRCM module and the PRCM module is then allowed to gate the clocks to the module. A slave module is said to be in IDLE state when its clocks are gated by the PRCM module. Similarly, an idled slave module may need to be wakened because of a service request from a master module or as a result of an event (called a wake-up event; for example, interrupt or DMA request) received by the slave module. In this situation the PRCM module enables the clocks to the module and then deasserts the idle request to signal the module to wake up. Although the protocol is completely hardware-controlled, software must configure the clock-management behavior for the slave module. This is done by setting the module register bit field <Module>_SYSCONFIG. SIDLEMODE or <Module>_SYSCONFIG. IDLEMODE (see Table 2-3). The behavior, listed in the Idle Mode Value column, must be configured by software.

**Table 2-3. Module Idle Mode Settings**

| Idle Mode Value | Selected Mode | Description |
|---|---|---|
| 0h | Force-idle | The module unconditionally acknowledges the idle request from the PRCM module, regardless of its internal operations. This mode must be used carefully because it does not prevent the loss of data at the time the clock is switched off. |
| 1h | No-idle | The module never acknowledges any idle request from the PRCM module. This mode is safe from a module point of view because it ensures that the clocks remain active. However, it is not efficient from a power-saving perspective because it does not allow the PRCM module output clock to be shut off, and thus the power domain to be set to a lower power state. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-3. Module Idle Mode Settings (continued)**

| Idle Mode Value | Selected Mode | Description |
|---|---|---|
| 2h | Smart-idle | The module acknowledges the idle request basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or direct memory access (DMA) requests are processed. This is the best approach to efficient system power management. |
| 3h | Smart-idle wakeup-capable mode | The module acknowledges the idle request basing its decision on its internal wakeup-capable mode activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or DMA requests are processed. This is the best approach to efficient system power management. The module may generate (IRQ- or DMA-request-related) wake-up events when in IDLE state. The mode is relevant only if the appropriate module's wakeup output(s) is implemented. |

The idle status of a slave module is indicated by the CM_<Powerdomain>_<Module>_CLKCTRL[x] IDLEST bit field in the PRCM module. Table 2-4 lists the possible idle status for a slave module.

**Table 2-4. Slave Module Idle Status**

| IDLEST Bit | Idle Status | Description |
|---|---|---|
| 0h | Functional | The module is fully functional. The interface and functional clocks are active. |
| 1h | In transition | The module is performing a wake-up or a sleep transition. |
| 2h | Interface idle | The module interface clock is idled. The module may remain functional if using a separate functional clock. |
| 3h | Full idle | The module is fully idle. The interface and functional clocks are gated. |

For the idle protocol management on the PRCM module side, the behavior of the PRCM module is configured in the CM_<Powerdomain>_<Module>_CLKCTRL[x] MODULEMODE bit field. Based on the configured behavior, the PRCM module asserts the idle request to the module unconditionally (that is, immediately when the software requests). Table 2-5 describes the configurable behavior of MODULEMODE.

**Table 2-5. Slave Module Mode Settings in PRCM**

| MODULE MODE Bit Value | Selected Mode | Description |
|---|---|---|
| 0h | Disabled | The PRCM module unconditionally asserts the module idle request. This request applies to the gating of the functional and interface clocks to the module. If acknowledged by the module, the PRCM module can gate all clocks to the module (that is, the module is completely disabled). |
| 1h | Reserved | Reserved |
| 2h | Enabled | This mode applies to a module when the PRCM module manages its interface and functional clocks. The functional clock to the module remains active unconditionally, while the PRCM module automatically asserts/deasserts the module idle request based on the clock-domain transitions. If acknowledged by the module, the PRCM module can gate only the interface clock to the module. |
| 3h | Reserved | Reserved |

In addition to the IDLE and STANDBY protocol, PRCM offers also the possibility to manage optional clocks, through a direct software control: “OPTFCLKEN” bit from programming register.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-6. Module Clock Enabling Condition**

| Clock Enabling Condition | Condition: AND | Condition: OR |
|---|---|---|
| Clock associated with STANDBY protocol | Clock domain is ready | MStandby is de-asserted<br>Mwakeup is asserted |
| Clock associated with IDLE protocol, as interface clock | Clock domain is ready | Idle status = FUNCT<br>Idle status = TRANS<br>SWakeup is asserted |
| Clock associated with IDLE protocol, as functional clock | Clock domain is ready | Idle status = FUNCT<br>Idle status = TRANS<br>Idle status = IDLE<br>SWakeup is asserted |
| Optional clock | Clock domain is ready | OPTFCLKEN = Enabled (0x1) |

### 2.1.3 Clock Domain

A clock domain is a group of modules fed by clock signals controlled by the same clock manager in the PRCM module (see Figure 2-2). By gating the clocks in a clock domain, the clocks to all the modules belonging to that clock domain can be cut to lower their active power consumption (that is, the device is on and the clocks to the modules are dynamically switched to ACTIVE or INACTIVE (GATED) states). Thus, a clock domain allows control of the dynamic power consumption of the device. The device is partitioned into multiple clock domains, and each clock domain is controlled by an associated clock manager within the PRCM module. This allows the PRCM module to individually activate and gate each clock domain of the device.

**Figure 2-2. Generic Clock Domain**

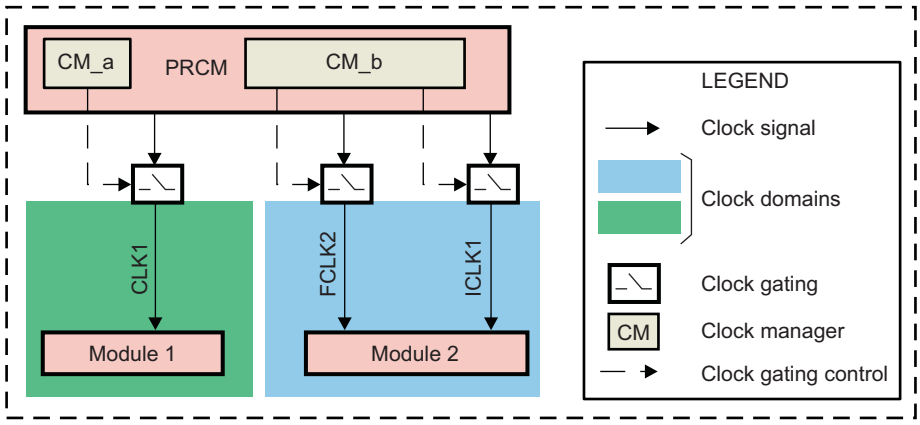



Figure 2-2 is an example of two clock managers: CM_a and CM_b. Each clock manager manages a clock domain. The clock domain of CM_b is composed of two clocks: a functional clock (FCLK2) and an interface clock (ICLK1), while the clock domain of CM_a consists of a clock (CLK1) that is used by the module as a functional and interface clock. The clocks to Module 2 can be gated independently of the clock to Module 1, thus ensuring power savings when Module 2 is not in use. The PRCM module lets software check the status of the clock domain functional clocks. The CM_<Clock_domain>_CLKSTCTRL[x] CLKACTIVITY_<FCLK/Clock name_FCLK> bit in the PRCM module identifies the state of the functional clock(s) within the clock domain. Table 2-7 shows the possible states of the functional clock.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-7. Clock Domain Functional Clock States**

| CLKACTIVITY Bit | Status | Description |
|---|---|---|
| 0 | Gated | The functional clock of the clock domain is inactive |
| 1 | Active | The functional clock of the clock domain is running |

### 2.1.3.1 Clock Domain-Level Clock Management

The domain clock manager can automatically (that is, based on hardware conditions) and jointly manage the interface clocks within the clock domain. The functional clocks within the clock domain are managed through software settings. A clock domain can switch between three possible states: ACTIVE, IDLE_TRANSITION, and INACTIVE. Figure 2-3 shows the sleep and wake-up transitions of the clock domain between ACTIVE and INACTIVE states.

**Figure 2-3. Clock Domain State Transitions**

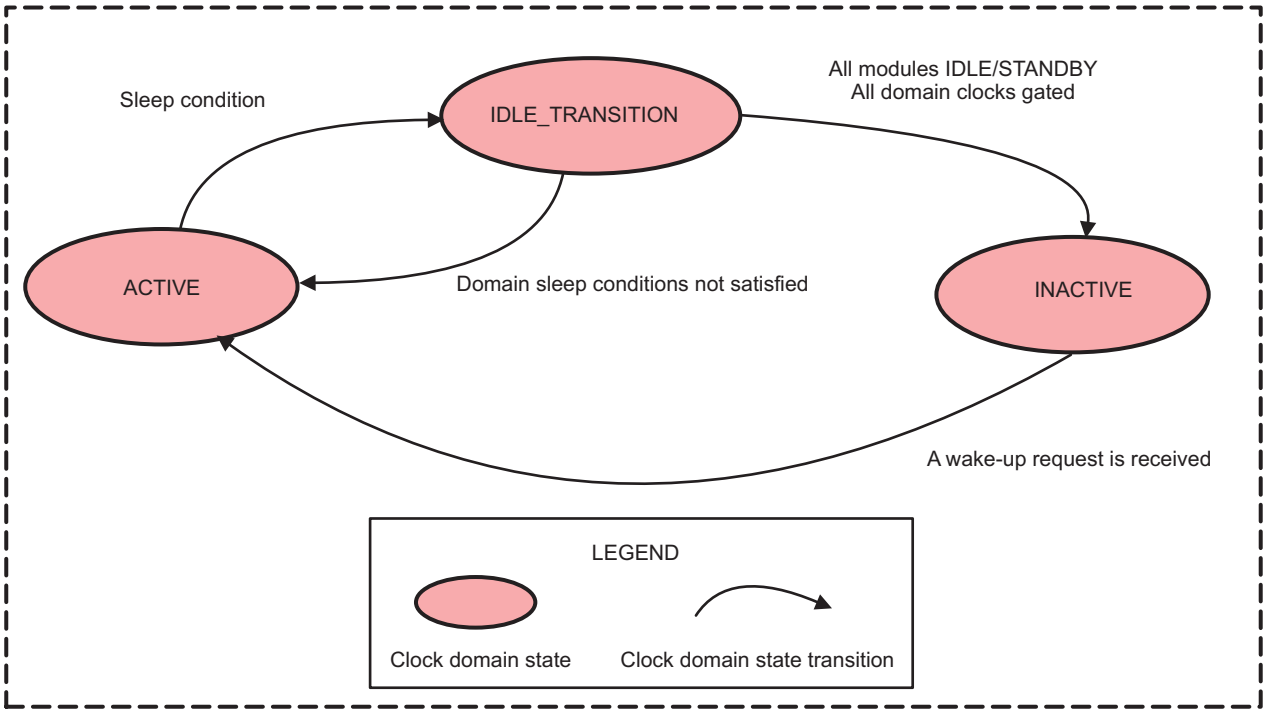



**Table 2-8. Clock Domain States**

| State | Description |
|---|---|
| ACTIVE | Every nondisabled slave module (that is, those whose MODULEMODE value is not set to disabled) is put out of IDLE state. · All interface clocks to the nondisabled slave modules in the clock domain are provided. All functional and interface clocks to the active master modules (that is, not in STANDBY) in the clock domain are provided.· All enabled optional clocks to the modules in the clock domain are provided. |
| IDLE_TRANSITION | This is a transitory state.· Every master module in the clock domain is in STANDBY state · Every idle request to all the slave modules in the clock domain is asserted.· The functional clocks to the slave module in enabled state (that is, those whose MODULEMODE values are set to enabled) remain active.· All enabled optional clocks to the modules in the clock domain are provided. |
| INACTIVE | All clocks within the clock domain are gated.· Every slave module in the clock domain (that is, those whose MODULEMODE is set to disabled or auto) is in IDLE state and set to disabled or auto mode.· Every optional functional clock in the clock domain is gated. |


Copyright © 2011–2016, Texas Instruments Incorporated

Each clock domain transition behavior is managed by an associated register bit field in the CM_<Clock domain>_CLKSTCTRL[x] CLKTRCTRL PRCM module. Table 2-9 describes the clock transition mode settings in the clock domain.

**Table 2-9. Clock Transition Mode Settings**

| CLKTRCTRL Bit | Selected Mode | Description |
|---|---|---|
| 0 | Reserved | Reserved |
| 1h | SW_SLEEP | A software-forced sleep transition. The transition is initiated when the associated hardware conditions are satisfied |
| 2h | SW_WKUP | A software-forced clock domain wake-up transition is initiated |
| 3h | Reserved | Reserved |

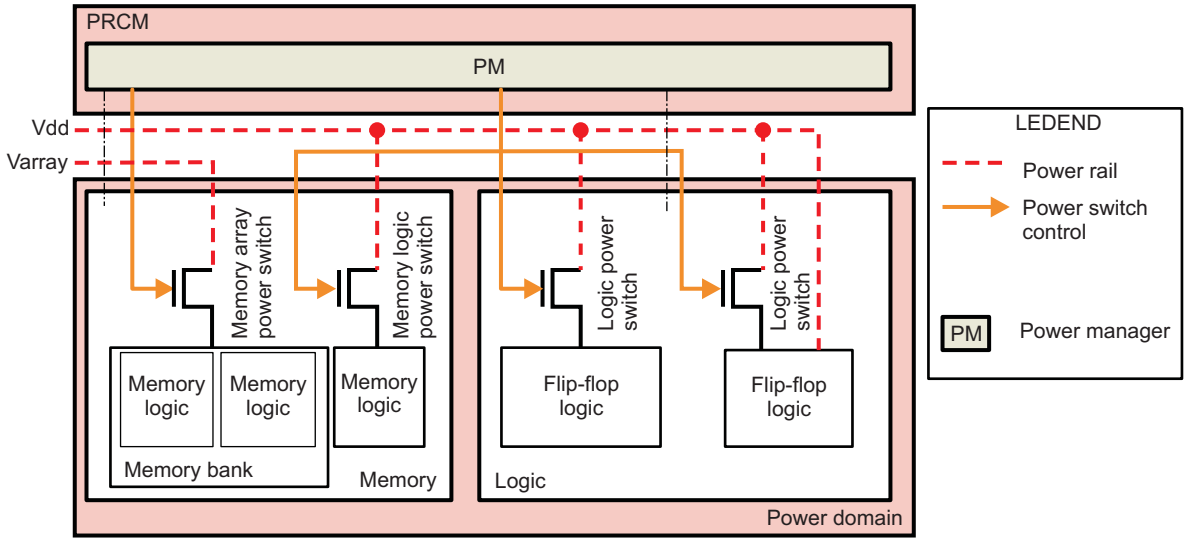
### 2.1.4 Power Management

The PRCM module manages the switching on and off of the power supply to the device modules. To minimize device power consumption, the power to the modules can be switched off when they are not in use. Independent power control of sections of the device allows the PRCM module to turn on and off specific sections of the device without affecting the others.

#### 2.1.4.1 Power Domain

A power domain is a section (that is, a group of modules) of the device with an independent and dedicated power manager (see Figure 2-4). A power domain can be turned on and off without affecting the other parts of the device.

**Figure 2-4. Power Domain Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

To minimize device power consumption, the modules are grouped into power domains. A power domain can be split into a logic area (Table 2-11) and a memory area (Table 2-10).

**Table 2-10. States of a Memory Area in a Power Domain**

| State | Description |
|---|---|
| On | The memory array is powered and fully functional. |
| Off | The memory array is powered down. |

**Table 2-11. States of a Logic Area in a Power Domain**

| State | Description |
|---|---|
| On | Logic is fully powered |
| Off | Logic power switches are off. All the logic (DFF) is lost. |

#### 2.1.4.2 Power Domain Management

The power manager associated with each power domain is assigned the task of managing the domain power transitions. It ensures that all hardware conditions are satisfied before it can initiate a power domain transition from a source to a target power state.

**Table 2-12. Power Domain Control and Status Registers**

| Register/Bit Field | Type | Description |
|---|---|---|
| PM_<Power domain>_PWRSTCTRL[1:0] POWERSTATE | Control | Selects the target power state of the power domain. It is OFF or ON. |
| PM_<Power domain>_PWRSTST[1:0] POWERSTATEST | Status | Identifies the current state of the power domain. It is OFF or ON. |
| PM_<Power domain>_PWRSTST[2] LOGICSTATEST | Status | Identifies the current state of the logic area in the power domain. It is OFF or ON. |
| PM_<Power domain>_PWRSTST[5:4] MEMSTATEST | Status | Identifies the current state of the memory area in the power domain. It is OFF or ON. |

#### 2.1.4.3 Power-Management Techniques

The following section describes the state-of-the-art power-management techniques supported by the device.

##### *2.1.4.3.1 Dynamic Voltage and Frequency Scaling*

Dynamic voltage and frequency scaling (DVFS) consists of opportunistically reducing the operating frequency of a logic block so that the operating voltage may be reduced accordingly. Dynamic power is related to the supply voltage so it is generally advantageous to reduce frequency and voltage instead of completing task faster and then going idle. The DVFS technique uses dynamic selection of the optimal operating frequency and voltage to allow a task to be performed in the required length of time. This reduces the active power consumption (power consumed while executing a task) of the device while still meeting task requirements.

**NOTE:** The values in Figure 2-5 are hypothetical. They are ment to clarify the concept only and do not represent valid test results on the device.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-5. Comparison of Energy Consumed With/Without DVFS**

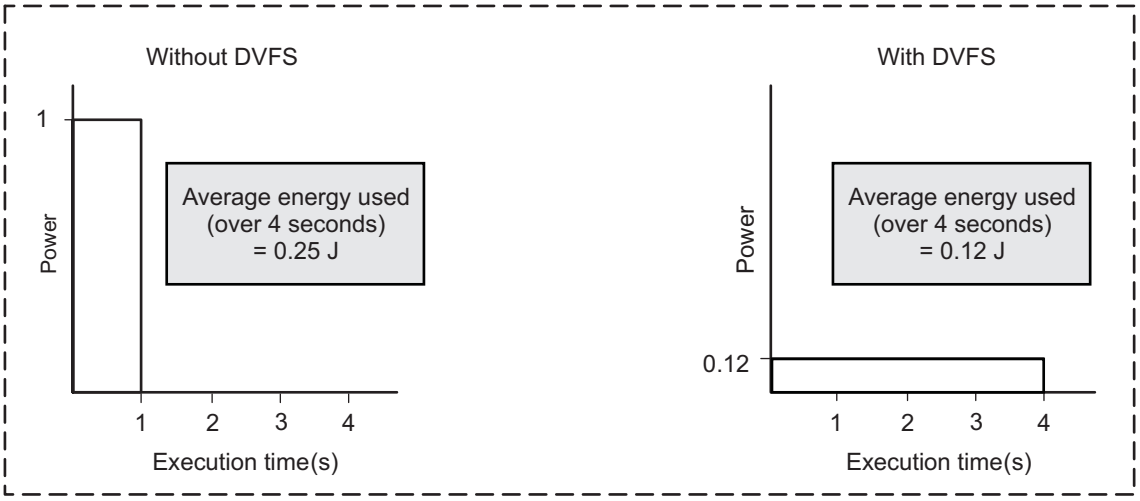



Figure 2-5 shows the DVFS technique by comparing a process executed at maximum frequency and operating voltage without applying DVFS to the same process executed at optimal frequency and voltage using DVFS, based on the task requirements. If a task that must terminate in 4 seconds is performed at maximum operating frequency (left side of the figure), it terminates in 1 second, and the remaining 3 seconds are spent in idle mode. With DVFS (right side of the figure), the operating frequency is reduced to an optimal level; the task takes the full 4 seconds to complete, but power consumption is reduced. In addition, the voltage can be reduced further to save power so the dynamic and leakage power consumption are reduced.

DVFS requires control over the clock frequency and the operating voltage of the device elements. By intelligently switching the individual elements of the device to their OPPs, the power consumption of the device for a given task can be minimized.

For practical reasons related to device making (flow, tools), DVFS can be used only for a few discrete steps, not over a continuum of voltage and frequency values. Each step, or OPP, is composed of a voltage (V) and frequency (F) pair. For an OPP, the frequency corresponds to the maximum frequency allowed at a voltage, or reciprocally; the voltage corresponds to the minimum voltage allowed for a frequency.

When applying DVFS, a processor or system always runs at the lowest OPP that meets the performance requirement at a given time. The user determines the optimal OPP for a given task and then switches to that OPP to save power.


Copyright © 2011–2016, Texas Instruments Incorporated

## 2.2 Power Reset Clock Management Structure

### 2.2.1 Introduction

The PRCM is structured using the architectural concepts presented in the 5000x Power Management Framework. This framework provides:

- A set of modular, reusable FSM blocks to be assembled into the full clock and power management mechanism.
- A register set and associated programming model.
- Functional sub-block definitions for clock management, power management, system clock source generation, and master clock generation.

The device supports an enhanced power management scheme based on:

- Six functional power domains:

Generic power domains:

- Active (DSP)
- AlwaysOn
- ISP (ISS)
- HDVICP
- SGX
- DSS

The PRCM provides the following functional features:

- Software configurable for direct, automatic, or a combination thereof, functional power domain state transition control
- Device power-up sequence control
- Device sleep / wake-up sequence control
- Centralized reset generation and management
- Centralized clock generation and management

The PRCM modules implement these general functional interfaces, further described in Table 2-13:

- OCP configuration ports
- Direct interface to device boundary
- Power switch control signals
- Device control signals
- Clocks control signals
- Resets signals
- A set of power management protocol signals for each module to control and monitor standby, idle and wake-up modes (CM and PRM)
- Emulation signals

**Table 2-13. PRCM Functional Interfaces**

| Interface | Description |
|---|---|
| OCP configuration ports | The OCP port for the PRCM module is used to control power, reset and wake-up Management. |
| OCP slave | PRCM implements a 32-bit OCP target interface. |
| Power switch control signals | The device has power domain switches over the device. This interface provides PRCM control over power domain switches and receives responses from the power domains which indicate the switch status. It also controls the isolation signals. The control for power domain switches is latched in the PRCM Status Registers. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-13. PRCM Functional Interfaces (continued)**

| Interface | Description |
|---|---|
| Device control signals | This interface provides PRM management of several device-level features which are not specific to any single power domain. This PRCM interface controls signals to/from the device for global control:<br>• Device type coding<br>• I/Os isolation control |
| Clocks control signals | This interface gathers all clock inputs and outputs managed by PRCM modules. |
| Resets signals | This interface gathers all resets inputs and outputs managed by PRCM module. |
| Modules Power Management Control | Modules or subsystems in the device are split over two categories:<br>• **Initiator:** an initiator is a module able to generate traffic on the device interconnects (typically: processors, MMU, eDMA...). The PRCM module handles all initiator modules' power management interfaces: MStandby signal MWait signal.<br>• **Target:** a target is a module that cannot generate traffic on the device interconnects, but that can generate interrupts or DMA request to the system (typically, peripherals). PRCM handles a power management handshake protocol with each module or subsystem. This protocol allows performing proper clock and power transition, taking into account each module activity or state. PRCM module handles all target modules' power management interfaces: SIdleReq signal, SIdleAck signal, FCLKEN signal. It also handles all target modules' wake-up: SWakeup signal (1). |

(1) Only USB and PCIe support SWakeUp.


Copyright © 2011–2016, Texas Instruments Incorporated

## 2.3 Clock Generation and Management

PRCM provides a centralized control for the generation, distribution and gating of most clocks in the device. PRCM gathers externally generated clocks and device internally generated clocks (DPLL clocks) for distribution to the other modules in the device. The PRCM module manages system clock generation.

### 2.3.1 Types of Clocks

The PRCM produces two types of clocks: interface and functional.

**Interface clocks:** these clocks primarily provide clocking for the system interconnect modules and the portions of device's functional modules which interface to the system interconnect modules. In most cases, the interface clock supplies the functional module's system interconnect interface and registers. For some modules, the interface clock is also used as functional clock. In this specification, interface clocks are represented by blue lines.

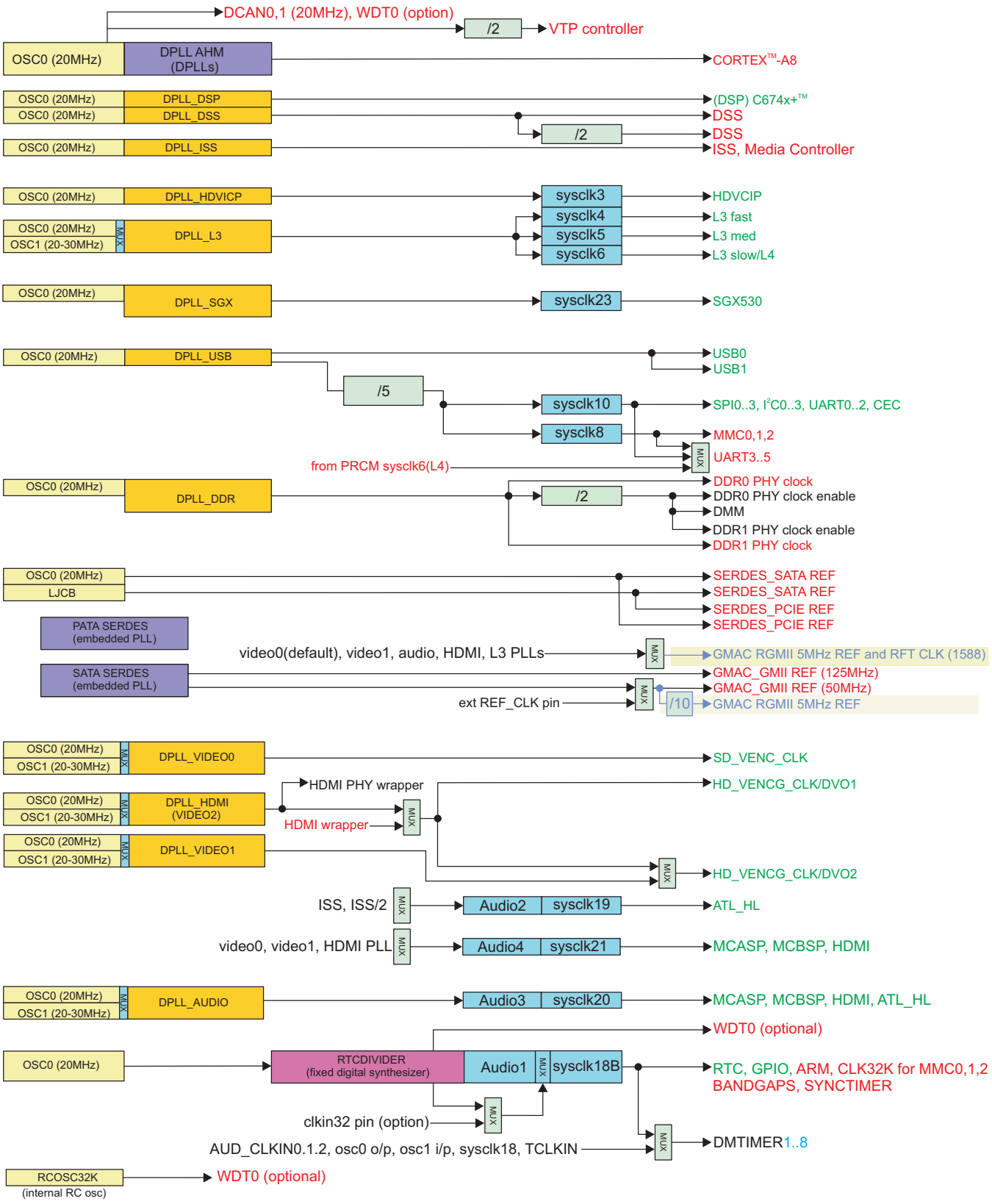
**Functional clock:** this clock supplies the functional part of a module or a sub-system. In some cases, a module or a subsystem may require several functional clocks: 1 or several main functional clock(s), 1 or several optional clock(s). A module needs its main clock(s) to be operational. Optional clocks are used for specific features and can be shutdown without stopping the module.

### 2.3.2 Clock Structure

The device has 12 on-chip PLLs at top level, and three additional inside hard macros (two of them in SERDES, one of them in Cortex™-A8 Subsystem). All of the twelve top level DPLLs are DPLLLJ instances which includes bypass feature inside. The Cortex™-A8 Subsystem is a DPLLS instance. Figure 2-6 shows the detailed clock structure of the device.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-6. Clock Structure**





Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-14. PLL Names and Types**

| PLL Name | PLL Type |
|---|---|
| DPLL_MPU(For Cortex™-A8) | DPLLS |
| DPLL_DSP | DPLLLJ |
| DPLL_ISS (DPLL_ISP) | DPLLLJ |
| DPLL_HDVICP | DPLLLJ |
| DPLL_L3 | DPLLLJ |
| DPLL_DDR | DPLLLJ |
| DPLL_USB | DPLLLJ |
| DPLL_SGX530 | DPLLLJ |
| DPLL_AUDIO | DPLLLJ |
| DPLL_HDMI | DPLLLJ |
| DPLL_VIDEO0 | DPLLLJ |
| DPLL_VIDEO1 | DPLLLJ |
| DPLL_DSS | DPLLLJ |
| DPLL_SATA_SERDES | DPLLLJ |
| DPLL_PATA_SERDES | DPLLLJ |

The device has two reference clocks which are generated by on-chip oscillators or externally. The main oscillator on the device is 20 MHz (DEVOSC). Optionally, if DEVOSC is 20 MHz and a precise audio or video frequency is desired, a second oscillator (AUXOSC) may be utilized for video and/or audio PLLs (e.g., 27, 22.5792 MHz or other frequency between 20-30 MHz). For instance, if a precise audio reference clock is needed, AUXOSC would provide this (e.g., 22.5792 for 44.1k audio) and then the video clock must be derived from Video PLL based on a multiply/divide of 20 MHz (pre-divide in PLL allows precise 27 MHz multiples from 20 MHz reference).

It is also possible to have one off-chip VCXO to supply the AUXOSC reference clock and in this case, the on-chip oscillator is bypassed.

In addition to the DEVOSC (20 MHz) and AUXOSC (20-30 MHz) reference clocks, there are an optional external 100 MHz differential clock input for the PCIe SERDES. The SATA SERDES also provides references for EMAC switch IP (125 MHz for GMII, 50 MHz RMII, 5 MHz RGMII).

A separate 250 MHz reference is used for EMAC RGMII mode, this comes from a mux which also feeds the 1588 reference clock. This clock has multiple options (see SERDES and Ethernet clocking diagram), but defaults to VIDEO0 PLL. Therefore to use reset isolation, the user must use VIDEO0 PLL at 250 MHz, which is also a suitable for 1588 reference clock rate.

There is third clock input for the RTC and GPIO0,1,2,3 and MMC0,1,2 debounce. This is an optional 32768 Hz clock input (no on-chip oscillator). Alternatively a fixed divider of the on-chip 20 MHz oscillator (DEVOSC) may be used to generate the 32768 Hz clock (provided by RTCDIVIDER). The 32k clock (regardless of source: pin or RTCDIVIDER source) is also used as optional reference for the eight timers, a reference clock for memory LDOs and bandgaps, and SYNCTIMER. The direct RTCDIVIDER output is also used as optional clock for WDT0 (Cortex™-A8 watchdog and as an optional clock source for ARM functional clock for very low power mode). By the Control Module MMR bits is provided a glitchless clock gating (enable/disable). This allows the clocks to be turn off when the modules are inactive state with a goal to minimize power consumption. During reset however, free running clock should be allowed such that reset can fully propagate.

**NOTE:** The ADPLLS in Cortex™-A8 Subsystem and the two ADPLLJ in SERDES could be controlled by Control Module MMR bits through a wrapper which utilizes the direct mode control ports on the DPLLs.

There are four individual clock mux instances that select between the DEVOSC and AUXOSC in the Main/Video/Audio PLL structures. In addition, the Cortex™-A8 subsystem clock has a mux to select between DEVOSC clock and RTCDIVIDER clock. The mux is controlled by the register MPU_CLKSRC.MPU_SOURCE.


Copyright © 2011–2016, Texas Instruments Incorporated

**CAUTION**

"Glitch-free" muxes have an output which is free from any glitches. It requires the previously selected clock and the newly selected clock to both be free-running for switch-over to occur. Switchover could take 1-2 clock cycles of previously selected and newly selected clock to complete. If newly selected clock is idle, a switchover never occurs (previously selected clock continues to pass through the mux). Do not switch to a non-existent clock.

### 2.3.3 Main PLL Clock Structure

The PLL clock structure includes the DSP, SGX530, HDVICP, L3, ISS, and DSS PLLs. Figure 2-7 shows the clock structure.

**Figure 2-7. Main PLL Clock Structure**

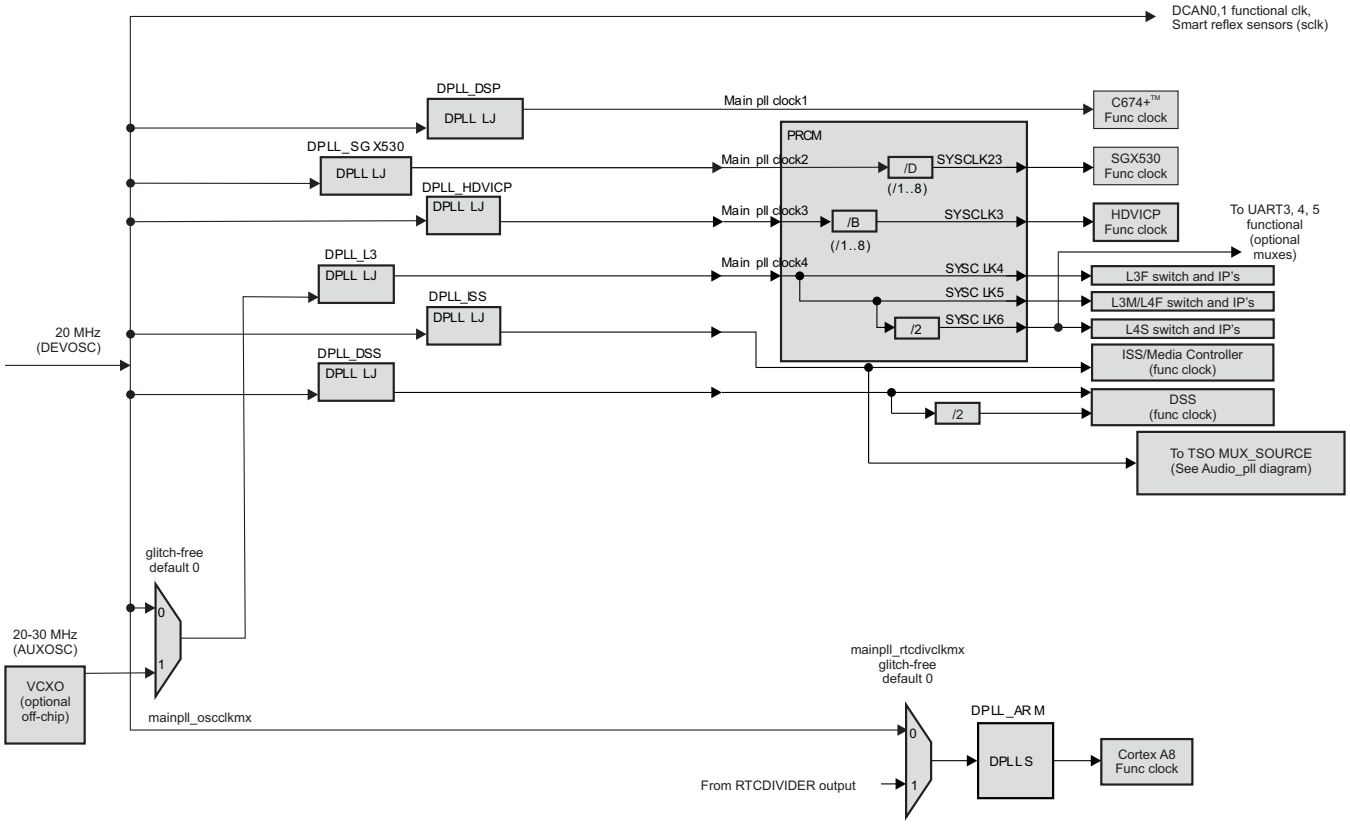



**Table 2-15. Main PLL Clock Source**

| SYSCLK | Source | Destination |
|---|---|---|
| - | DPLL_DSP | To C674x+™SubSystem |
| SYSCLK23 | DPLL_SGX530 | SGX530 functional clock (CORECLK) |
| SYSCLK3 | DPLL_HDVICP | To HDVICP |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-15. Main PLL Clock Source (continued)**

| SYSCLK | Source | Destination |
|---|---|---|
| SYSCLK4 | DPLL_L3 | L3_FAST, OCP clock for Cortex™-A8, HDVICP, expansion slot(master), MMU,C674x+ Subsystem, ISS, TPTCs, C674x+Subsystem, HDVICP SL2, TPCC |
| | | L3_MID, L4_FAST, OCP clock for SGX530 (SYSCLK and MEMCLK), ISS Media Controller, 3PGSW, SATA, PCI ex, VCP2, MLB, FDIF, DAP, OCMC RAM, MMCHS2. |
| SYSCLK6 | | L3_SLOW, L4_SLOW, OCP clock for UART0..2,3..5, I2C, SPI, SDIO, TIMER, GPIO, PRCM, McASP, McBSP, GPMC, HDMI, MLB, WDT0, PATA, Mailbox, Spinlock, , USB SS. |
| | | Func clock for UART3,4,5 (optional), Efuse control, MMU config, P1500, ELM, MMCHS0..1, ATL, FDIF config. |
| | DPLL_ISS (DPLL_ISP) | ISS/Media Controller |
| | DPLL_DSS | DSS (proc, l3) also div2 of this clock (top level divider) goes to DSS proc_d2 and l4 clocks. |

As shown in Table 2-15, each of the DPLLLJs can be configured as appropriate for the desired functional clock of each major core (C674x+ subsystem, HDVICP, ISS, SGX530, DSS, L3/4, A8). They are entirely independent from one another other than the shared reference clock (DEVOSC) except for the DPLL_L3, can optionally choose to use AUXOSC. This is useful as some IPs may use their L4 clock to derive other key frequencies so a slight change to L3/L4 speed could help achieve this while still being in reasonable range for L3/L4 functionality.

The DCAN0..1 functional clocks come from the 20 MHz oscillator to allow fixed rate for baud rate generation and minimal jitter (DEVOSC clock versus PLL clock). Also, the Smart Reflex sensor clock comes from fixed DEVOSC oscillator (20MHz) and the VTP controller clock comes from DEVOSC/2 (fixed /2 divider at top level) such that the frequency of these is the same across all operating points (OPPs).

Table 2-16 lists the divide ratios supported for each of the divider in Main PLL.

**Table 2-16. : Main PLL Supported Configurable Dividers**

| PLL Source | Destination | Control Bit Filed | Divider | Supported Divide Ratios | Default Value |
|---|---|---|---|---|---|
| DPLL_HDVICP | HDVICP Clocks | CM_SYSCLK3_CLKSEL.[CLKSEL] | B | 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 | 1/1 |
| DPLL_SGX530 | SGX530 Clocks | CM_SYSCLK23_CLKSEL.[CLKSEL] | D | 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 | 1/3 |

**Table 2-17. Clock Multiplexing Used With Main PLLs**

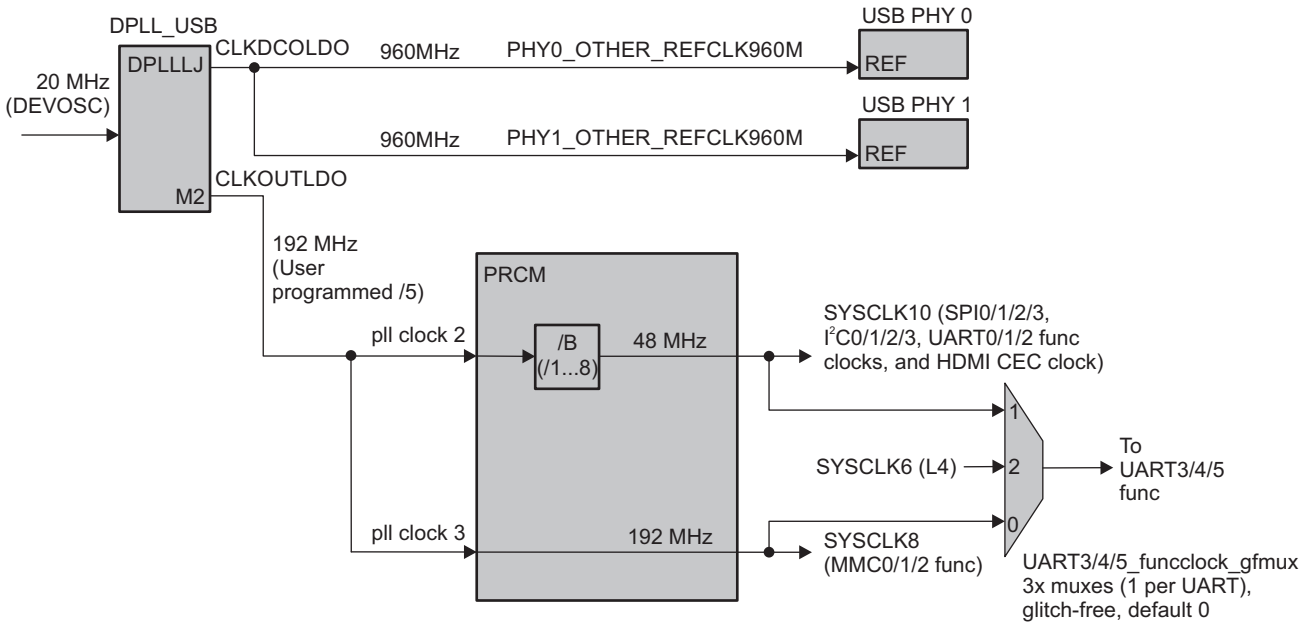
| Mux | Mux Source 0 | Mux Source 1 | Control Bit Field |
|---|---|---|---|
| DPLL_L3 Input Clock Mux | DEVOSC | AUXOSC | OSC_SRC. L3_PLL_SOURCE (control module register) |
| DPLL_MPU Clock Source | DEVOSC | RTC_DIVIDER_OUTPUT | MPU_CLKSRC. MPU_SOURCE |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.3.4 USB PLL Clock Structure

Figure 2-8 shows the clock structure of the USB PHY. The PRCM gets the inputs from the divide by 5 output of the DPLL for clocking the UART, McBSP. These are driven by the DPLL_USB (DPLLLJ), SYSCLK10 which is 48 MHz and goes to SPI0,1,2,3, I2C0,1,2,3, UART0,1,2, UART3,4,5 (optional via glitchless muxes), and the CEC clock of HDMI wrapper. SYSCLK8 is used to generate 192 MHz to UART3,4,5 (optional via glitchless muxes) and MMC0,1,2. Optionally, the PRCM SYSCLK6 (L4 clock) clock output could optionally source each of UART3,4 or 5 functional clocks (this provides additional baud rate options for UART3,4,5, without affecting MMC0,1,2). This mux is controlled by McBSP_UART_CLKSRC register in the PRCM.

**Figure 2-8. USB PLL Clock Structure**




**NOTE:** The /5 divider (M2 post divider) is embedded inside the DPLL_USB module (see Figure 2-19 for details). The /5 divider can be configured in the USBPLL_M2NDIV[22:16] M2 bit field. It is stated as a /5 divider, as the reset (default) value of the M2 bit field is 0x5.

Table 2-18 shows supported divide ratios for dividers in PRCM.

**Table 2-18. USB PLL Supported Dividers**

| PLL Source | Destination | Control Bit Filed | Divider | Supported Divide Ratios | Default Value |
|---|---|---|---|---|---|
| PLL_USB | SPI0..3,I2C0..3, UART0..2, HDMI CEC | CM_SYSCLK10_CLKSEL.[CLKSEL] | B | 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 | 1/1 |

**Table 2-19. Clock Multiplexing used with USB PLL**

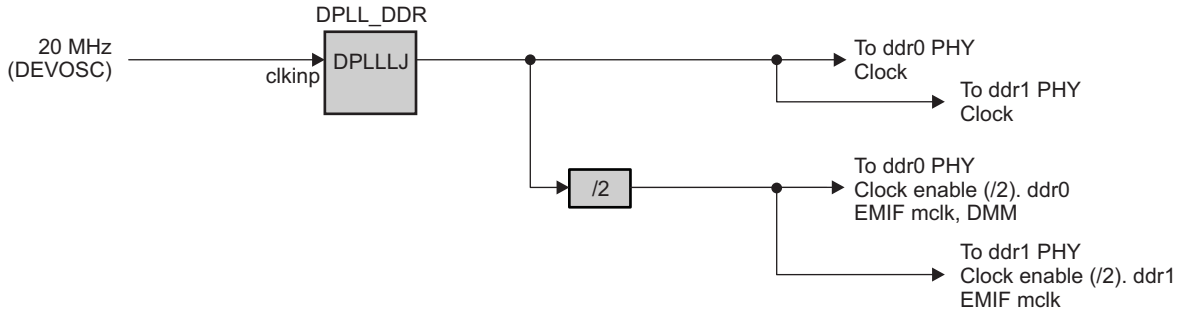
| Mux | Mux Source 0 | Mux Source 1 | Mux Source 2 | Control Bit Field |
|---|---|---|---|---|
| UART3 clock source | SYSCLK8 | SYSCLK10 | SYSCLK6 | MCBSP_UART_CLKSRC.UART3_CLK_SOURCE |
| UART4 clock source | SYSCLK8 | SYSCLK10 | SYSCLK6 | MCBSP_UART_CLKSRC.UART4_CLK_SOURCE |
| UART5 clock source | SYSCLK8 | SYSCLK10 | SYSCLK6 | MCBSP_UART_CLKSRC.UART5_CLK_SOURCE |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.3.5 DDR PLL Clock Structure

Figure 2-9 shows the DDR PLL. The two DDR PHY interfaces (DDR0 or DDR1) share a single PLL. DMM is synchronous to the DDR EMIF controller.

**Figure 2-9. : DDR PLL Clock Structure**




**NOTE:** Each EMIF must be configured to the same rate due to the shared PLL (If different memory configurations are used even with same rates, there could be some performance impact through DMM for asymmetrical configurations when interleaving).

PRCM does not provide a clock signal to DDR_PHY.

### 2.3.6 SERDES and Ethernet Clock Structure

Figure 2-10 shows the SERDES and Ethernet clocking. The SATA SERDES reference input is driven from the 20 MHz on-chip oscillator. The SATA SERDES also provides reference clocks to the Ethernet GMII (125 MHz) and RMII (50 MHz) and RGMII 5 MHz clock. Ethernet clocking provides several options to receive or output the RMII reference clock from a pin. This mux is controlled by the bit RMII_REFCLK_SRC. REFCLK_SOURCE in the PRCM. Also, several clock sources are required for the 1588 reference clock (cpts_rft_clk), also shown in Figure 2-10.

This 1588 reference clock also feeds the RGMII 250 MHz reference clock in case of RGMII. To use reset isolation, the VIDEO0 PLL option must be used for this clock since the clock mux control is not reset-isolated and VIDEO0 is the default clock. Note that 250 MHz is also a suitable frequency for 1588 reference. Also note that the use of this clock also means that VIDEO0 cannot be set to 54 MHz to support AVDAC use, which is a use case limitation. To use AVDAC, lose the reset isolation feature and select another clock.

The PCIe SERDES always requires its reference clock to be sourced from an external 100 MHz reference from the LJCB input pads.

The SATA SERDES could optionally use the external 100 MHz reference as well (same LJCB inputs as used for PCIe). This may be practical if spread spectrum clocking (SSC) is desired to reduce EMI on the SATA interface and especially if such clock is already present due to PCIe usage. If the clock is not present, then using the DEVOSC 20 MHz as reference for SATA SERDES is another solution.

**NOTE:** The PLL inside of SATA SERDES is isolated from the rest of the SERDES naturally by design. The PLL may remain active while the rest of the SERDES is put in low power mode (feature of SERDES). The control to the PLL within SERDES is located at top level in the control module to allow the ability to keep PLL in its active/configured state for reset-isolation of Ethernet.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-10. SERDES and Ethernet Clock Structure**

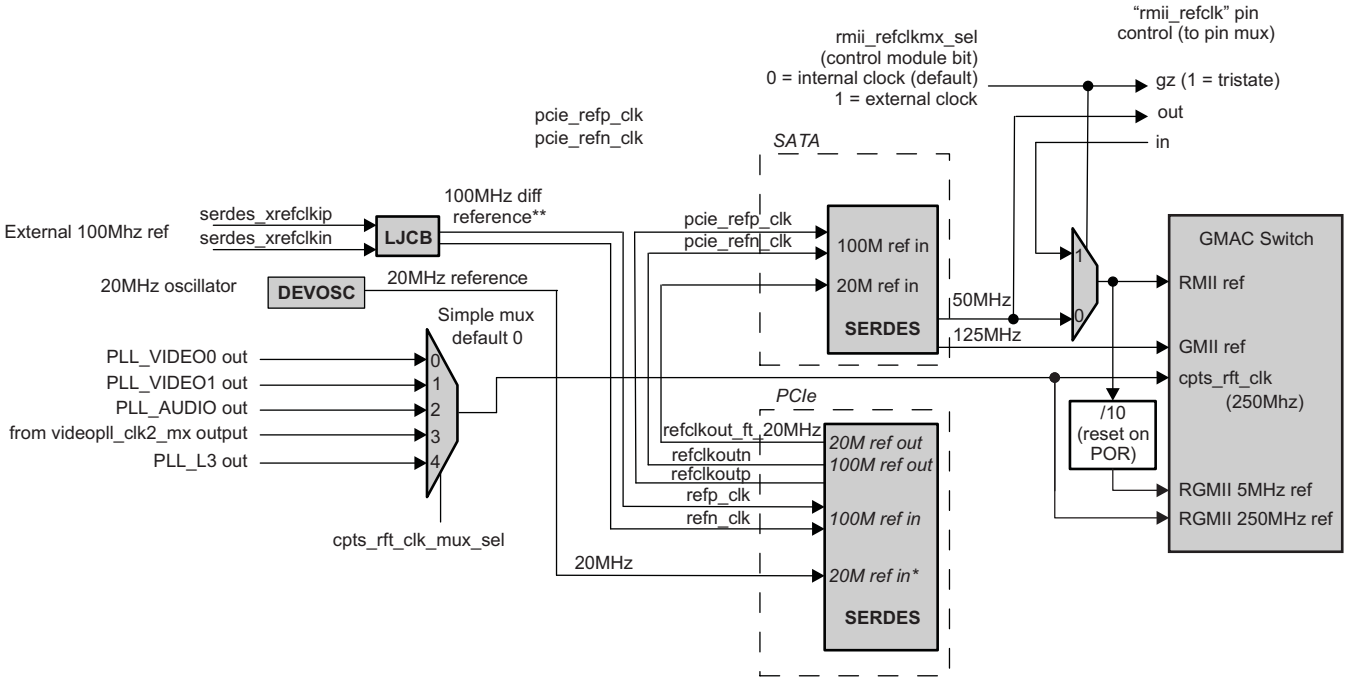



**Table 2-20. Clock Multiplexing Used With SERDES PLL**

| Mux | Mux Source 0 | Mux Source 1 | Mux Source 2 | Mux Source 3 | Mux Source 4 | Bit Field |
|---|---|---|---|---|---|---|
| Cpts_RFT_CLK source Mux | VIDEO0_PLL_ OUT | VIDEO1_PLL_ OUT | AUDIO_PLL_ OUT | VIDEO_PLL_ CLK2 | L3_PLL_ OUT | RMII_REFCLK_ SRC.CPTS_RFT _CLK |
| RMII Ref Clk Mux | 50 MHz SATA SerDes | REFCLK_PIN | | | | RMII_REFCLK_ SRC.REFCLK_ SOURCE |

### 2.3.7 Video PLL Clock Structure

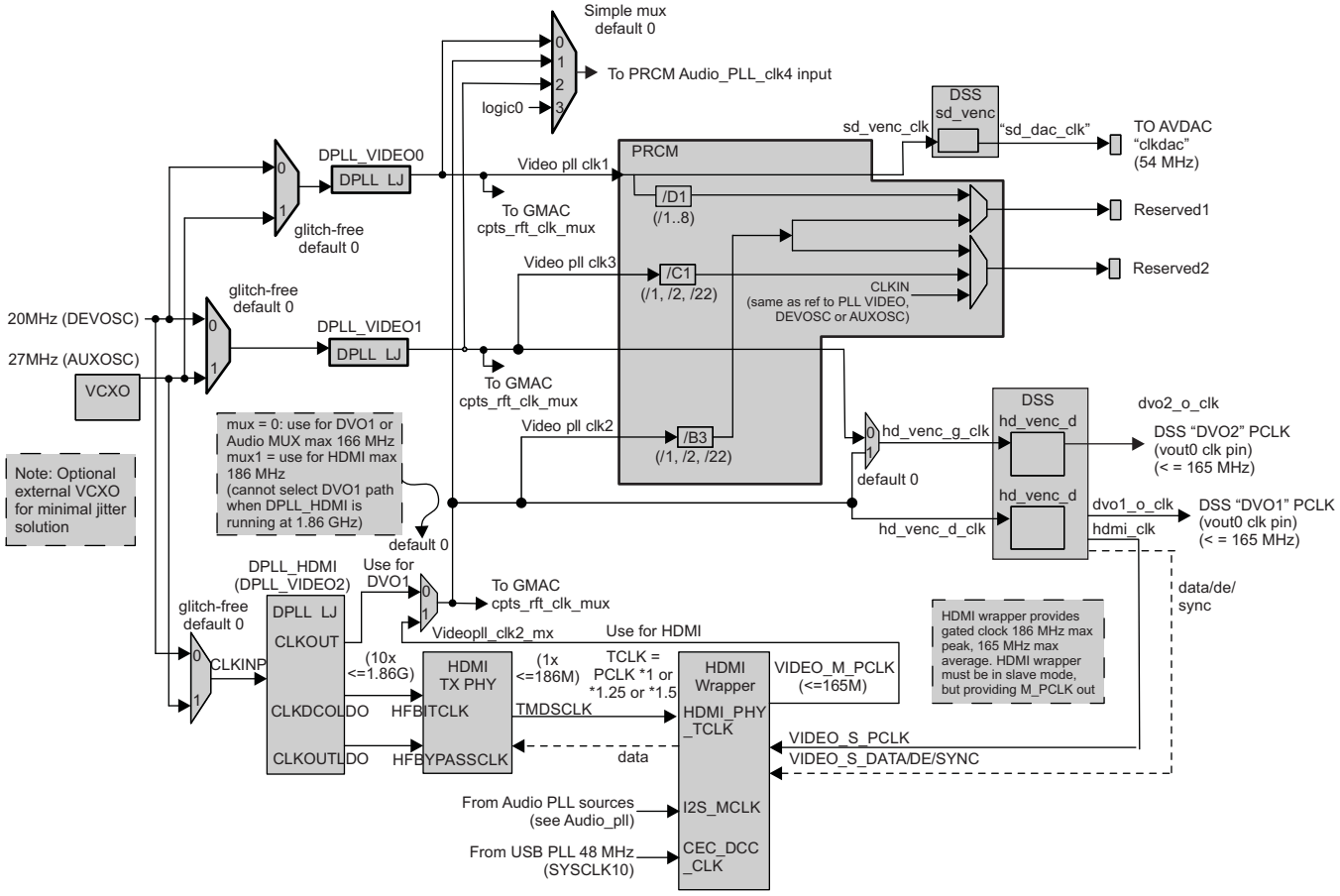
As shown in Figure 2-11, there are three DPLLLJs for video. Each of the video DPLLs (DPLL_VIDEO0, DPLL_VIDEO1, DPLL_HDMI), can be optionally supply from DEVOSC (20 MHz) or AUXOSC (20-30 MHz osc). Deriving 54/27 MHz clocks from 20 MHz DPLL reference is useful in case it is desired to use AUXOSC instead for an precise audio frequency crystal (e.g., 22.5792). Because the device uses DPLLLJ, this is easy to achieve, as the user may pre-divide (e.g,. to 1 or 2 MHz) and then multiply up to 27 or 54MHz.

PLL_VIDEO0 supplies the SD VENC clock to display subsystem directly. This clock is always 54 MHz. DPLL_HDMI provides clock to HD_VENC_D_CLK either directly or via HDMI wrapper logic. This clock is used by the display subsystem to derive DVO1 clock (vout1). HD_VENC_D_CLK needs to support 13.5 MHz, 27 MHz, 54 MHz, 74.25 MHz, 148.5 MHz, 162 MHz and 165 MHz frequencies. It supports 1000/1001 times all of these frequencies (e.g., via fractional multiply adjustment). DPLL_VIDEO1 supplies clock to HD_VENC_G_CLK of display subsystem which is used to derive the DVO2 clock (vout0). This clock has similar frequency requirements as HD_VENC_D_CLK.

Optionally the HD_VENC_G_CLK could also be supplied from DPLL_HDMI. HD_VENC_G (DVO2) has similar frequency requirements, and supports independent frequency from HDMI/DVO1. This addresses the requirements for two digital displays on DVO2 and DVO1 with different resolutions and or timings.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-11. Video Clock Structure**




The device has options for DCXO/VCXO type feature to adjust video frequency on the fly: Use an off-chip VCXO on AUXOSC (can feed clock to AUXOSC_XI pad and run AUXOSC in bypass). This provides the most smooth, low jitter solution.

In each case, software controls the frequency adjustment (via manipulation of off-chip VCXO or other means). There is a 27 or 54 MHz clock generated from each of these PLLs (via PRCM) which is routed back to the STC to generate the error with respect to the clock embedded in the in-coming transport stream.

The HDMI PHY requires a high speed clock (up to 1.85 GHz) from a dedicated DPLLLJ. From this clock a TMDS clock is derived which is 1/10 the high speed clock and which is driven to the HDMI wrapper. The clock to DSS used to generate PCLK is equal to TMDS clock /1, /1.25 or /1.5 depending on deep color mode. The HDMI wrapper provides a shaped clock for this purpose. This clock is muxed into the DSS clock inputs.

**NOTE:** If any of the three video DPLLs are not needed for video, any of these can be optionally used to generate a second audio master frequency, as the sources are muxed and routed to the Audio pll clk4 input of the PRCM and to ATL and McASP 3,4,5 references (see Figure 2-12).



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-21. Video PLL Supported Dividers**

| Control Bit Filed | Divider | Supported Divide Ratios | Default Value |
|---|---|---|---|
| CM_VPD1_CLKSEL [CLKSEL] | D1 | 1/1,1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 | 1/8 |
| CM_VPB3_CLKSEL [CLKSEL] | B3 | 1/1,1/2, 1/22 | 1/22 |
| CM_VPC1_CLKSEL [CLKSEL] | C1 | 1/1,1/2, 1/22 | 1/22 |

**Table 2-22. Clock Multiplexing Used With Video PLLs**

| Mux | Mux Source 0 | Mux Source 1 | Mux Source 2 | Bit Field |
|---|---|---|---|---|
| DPLL_VIDEO0 Source Mux | DEVOSC | AUXOSC | | OSC_SRC.VIDEO0_PLL_SOURCE |
| DPLL_VIDEO1 Source Mux | DEVOSC | AUXOSC | | OSC_SRC.VIDEO1_PLL_SOURCE |
| DPLL_HDMI Source Mux | DEVOSC | AUXOSC | | OSC_SRC.HDMI_PLL_SOURCE |
| VIDEO_PLL_ CLK2 Mux | HDMI_PLL_CLKOUT | VIDEO_M_PCLK | | VIDEO_PLL_CLKSRC.VIDEO_PLL_CLK2_SOURCE |
| VIDEO_PLL_ CLK Mux | VIDEO0_PLL_OUT | HDMI_PLL_OUT | VIDEO1_PLL_OUT | VIDEO_PLL_CLKSRC.VIDEO_PLL_OUT_MUX_SOURCE |
| HD_VENC_G_CLK Source Mux | VIDEO1_PLL_OUT | VIDEO_PLL_CLK2 from VIDEOPLL_CLK2 Mux | | VIDEO_PLL_CLKSRC.HD_VENC_G_CLK_SOURCE |

### 2.3.8 Audio PLL Clock Structure

Audio clocking is based on a single dedicated Audio PLL which is a DPLLLJ. It provides several options for other sources of audio clocks including directly using AUXOSC input pad (drive clock onto AUXOSC XI pad and use AUXOSC in bypass), external pins (AUD_CLKIN0,1,2) and optionally using any of the video PLLs (PLL_VIDEO0, PLLVIDEO1, PLL_HDMI, and more, if any of these resources are not used for video).


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-12. Audio Clock Structure**

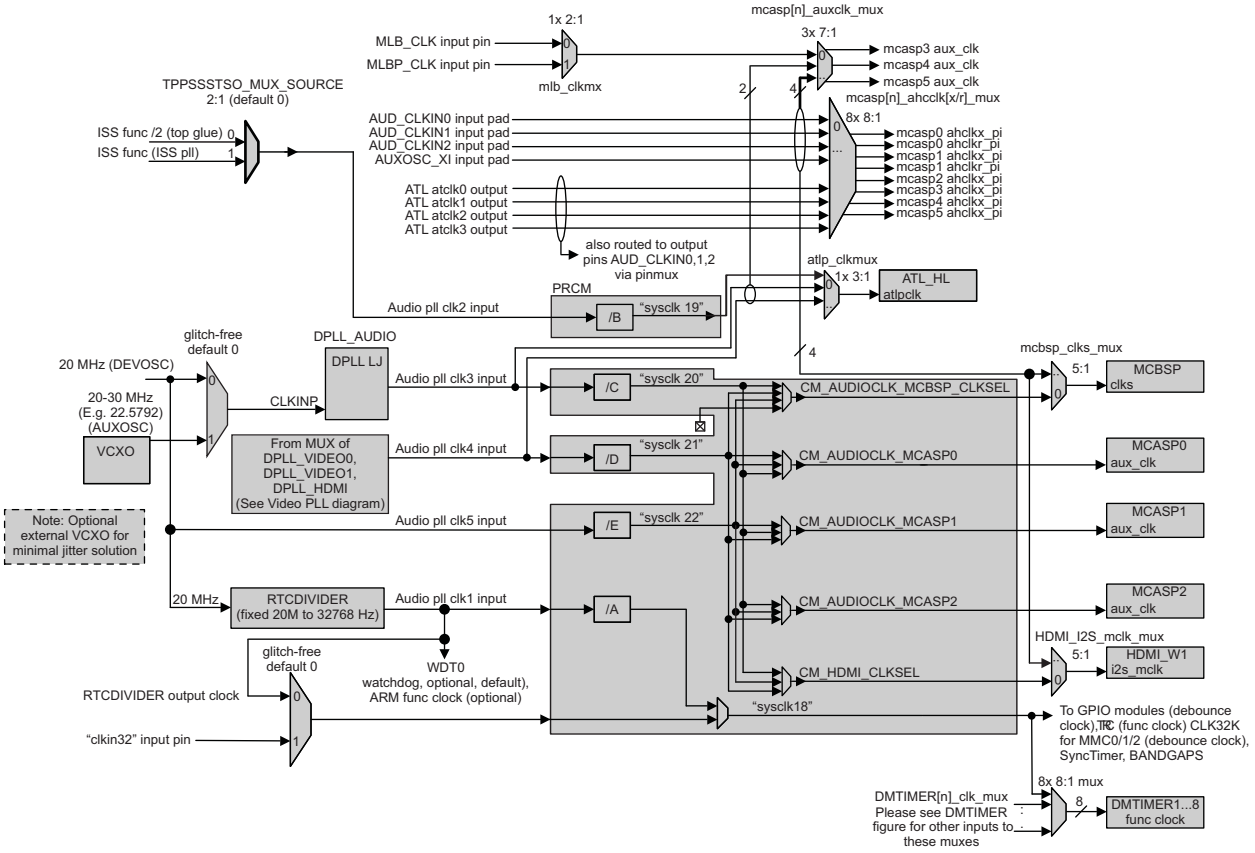



**Table 2-23. Audio Frequencies**

| | Over-Sample Rate | | | | | |
|---|---|---|---|---|---|---|
| **Sampling Frequency (KHz)** | **128** | **256** | **384** | **512** | **768** | **1024** |
| 16 | 2.0480 | 4.0960 | 6.1440 | 8.1920 | 12.2880 | 16.3840 |
| 22.05 | 2.8224 | 5.6448 | 8.4672 | 11.2896 | 16.9344 | 22.5792 |
| 24 | 3.0720 | 6.1440 | 9.2160 | 12.2880 | 18.4320 | 24.5760 |
| 32 | 4.0960 | 8.1920 | 12.2880 | 16.3840 | 24.5760 | 32.7680 |
| 44.1 | 5.6448 | 11.2896 | 16.9344 | 22.5792 | 33.8688 | 45.1584 |
| 48 | 6.1440 | 12.2880 | 18.4320 | 24.5760 | 36.8640 | 49.1520 |
| 64 | 8.1920 | 16.3840 | 24.5760 | 32.7680 | 49.1520 | 65.5360 |
| 88.2 | 11.2896 | 22.5792 | 33.8688 | 45.1584 | 67.7376 | 90.3168 |
| 96 | 12.2880 | 24.5760 | 36.8640 | 49.1520 | 73.7280 | 98.3040 |
| 128 | 16.3840 | 32.7680 | 49.1520 | 65.5360 | 98.3040 | 131.0720 |
| 176.4 | 22.5792 | 45.1584 | 67.7376 | 90.3168 | 135.4752 | 180.6336 |
| 192 | 24.5760 | 49.1520 | 73.7280 | 98.3040 | 147.4560 | 196.6080 |

Refer back to Figure 2-12 to see the connection of audio clocks to McASP0,1,2, McBSP and HDMI modules. In addition to the mux options provided inside PRCM, top-level muxes provide for McASP3,4,5 and ATL to share any of the audio clock sources as well as to add path from any of the sources to McASP0,1,2, McBSP or HDMI. Notice also that the McASP peripherals also have a setup of clock muxes to feed various clocks to their AHCLKX and AHCLKR inputs. Since there are six McASPs, rather than pin mux all 12 possible AHCLKX and AHXLKR pins, there are three loosely dedicated pins referred to as


Copyright © 2011–2016, Texas Instruments Incorporated

AUD_CLKIN0,1,2 which are used as optional sources for all of the McASP AHCLK ports as well as for various other timing functions (e.g., dmtimer, STC, etc). Any function which optionally uses an AUD_CLKIN0,1,2 pin as an input timing reference should use the signal coming off the input pad itself (and not through pin muxing logic). This is because some timing sources are also optionally routed as an output to these pins (e.g., ATL clock out and McASP AHCLKX outputs), in which case another peripheral could use that same pin as an input timing reference.

**NOTE:** Similar to the video case, there are several options for implementing adjustments to audio clocks. The smoothest is external VCXO (e.g., on AUXOSC For McASPs one can also use the ATL clock outputs as reference which is adjusted via ATL register writes.

**Table 2-24. Audio PLL Supported Dividers**

| Control Bit Filed | Divider | Supported Divide Ratios | Default Value |
|---|---|---|---|
| CM_APA_CLKSEL[CLKSEL] | A | 1/1,1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 | 1/1 |
| CM_SYSCLK19_CLKSEL[CLKSEL] | B | 1/1,1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 | 1/1 |
| CM_SYSCLK20_CLKSEL[CLKSEL] | C | 1/1,1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 | 1/1 |
| CM_SYSCLK21_CLKSEL[CLKSEL] | D | 1/1,1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 | 1/1 |
| CM_SYSCLK22_CLKSEL[CLKSEL] | E | 1/1,1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 | 1/1 |

A fixed 32,768 Hz clock frequency is provided on PRCM SYSCLK18. This clock has two sources. The Audio PLL Clock1 which is fed from a fixed divider (RTCDIVIDER) on the 20 MHz clock, and the “clkin32” input pin. SYSCLK18 provides clock for: RTC, DMTIMER1..8 (optional via mux), BANDGAPS, synctimer and debounce circuit for MMC0,1,2 and GPIO0,1,2,3.

**Table 2-25. Audio PLL Clock Sources**

| Mux | Source 0 | Source 1 | Source 2 | Source 3 | Source 4 | Source 5 | Source 6 | Source 7 | Bit Field |
|---|---|---|---|---|---|---|---|---|---|
| DPLL AUDIO SOURCE Mux | DEVOSC | AUXOSC | | | | | | | OSC_SRC.AUDIO_PLL_SOURCE |
| SYSCLK18 Source Mux1 | RTCDIVIDER_OUT | CLKIN32_PIN | | | | | | | SYSCLK18_CLKSRC.SYSCLK18_SOURCE |
| SYSCLK18 CLK SEL Mux | 32 Khz clock from SYSCLK18 Source Mux 1 | Audio PLL generated Clock after Divider A | | | | | | | CM_SYSCLK18_CLKSEL.CLKSEL |
| MLB_CLKMUX | MLB_CLK_INPUT | MLBP_CLK_INPUT | | | | | | | MLB_ATL_CLKSRC.MLB_SOURCE |
| MCASP3_AUXCLK_MUX | Output of MLB_CLKMUX | AUDIO_PLL_OUT | VIDEO_PLL_OUT_MUX | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | AUXOSC_XI | Reserved | MCASP345_AUX_CLKSRC.MCASP3_AUX_SOURCE |
| MCASP4AUXCLK_MUX | Output of MLB_CLKMUX | AUDIO_PLL_OUT | VIDEO_PLL_OUT_MUX | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | AUXOSC_XI | Reserved | MCASP345_AUX_CLKSRC.MCASP4_AUX_SOURCE |
| MCASP5AUXCLK_MUX | Output of MLB_CLKMUX | AUDIO_PLL_OUT | VIDEO_PLL_OUT_MUX | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | AUXOSC_XI | Reserved | MCASP345_AUX_CLKSRC.MCASP5_AUX_SOURCE |
| MCASP0_AHCLKX_PI Mux | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | AUXOSC_XI | ATCLK0 | ATCLK1 | ATCLK2 | ATCLK3 | MCASP_AHCLK_CLKSRC.MCASP0_AHCLKX_SOURCE |
| MCASP0_AHCLKR_PI Mux | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | AUXOSC_XI | ATCLK0 | ATCLK1 | ATCLK2 | ATCLK3 | MCASP_AHCLK_CLKSRC.MCASP0_AHCLKR_SOURCE |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-25. Audio PLL Clock Sources (continued)**

| Mux | Source 0 | Source 1 | Source 2 | Source 3 | Source 4 | Source 5 | Source 6 | Source 7 | Bit Field |
|---|---|---|---|---|---|---|---|---|---|
| MCASP1_AHCLKX_PI Mux | AUD_CLKIN 0 | AUD_CLKIN 1 | AUD_CLKIN 2 | AUXOSC_XI | ATCLK0 | ATCLK1 | ATCLK2 | ATCLK3 | MCASP_AHCLK_CLKSRC.MCASP1_AHCLKX_SOURCE |
| MCASP1_AHCLKR_PI Mux | AUD_CLKIN 0 | AUD_CLKIN 1 | AUD_CLKIN 2 | AUXOSC_XI | ATCLK0 | ATCLK1 | ATCLK2 | ATCLK3 | MCASP_AHCLK_CLKSRC.MCASP1_AHCLKR_SOURCE |
| MCASP2_AHCLKX_PI Mux | AUD_CLKIN 0 | AUD_CLKIN 1 | AUD_CLKIN 2 | AUXOSC_XI | ATCLK0 | ATCLK1 | ATCLK2 | ATCLK3 | MCASP_AHCLK_CLKSRC.MCASP2_AHCLKX_SOURCE |
| MCASP3_AHCLKX_PI Mux | AUD_CLKIN 0 | AUD_CLKIN 1 | AUD_CLKIN 2 | AUXOSC_XI | ATCLK0 | ATCLK1 | ATCLK2 | ATCLK3 | MCASP_AHCLK_CLKSRC.MCASP3_AHCLKX_SOURCE |
| MCASP4_AHCLKX_PI Mux | AUD_CLKIN 0 | AUD_CLKIN 1 | AUD_CLKIN 2 | AUXOSC_XI | ATCLK0 | ATCLK1 | ATCLK2 | ATCLK3 | MCASP_AHCLK_CLKSRC.MCASP4_AHCLKX_SOURCE |
| MCASP5_AHCLKX_PI Mux | AUD_CLKIN 0 | AUD_CLKIN 1 | AUD_CLKIN 2 | AUXOSC_XI | ATCLK0 | ATCLK1 | ATCLK2 | ATCLK3 | MCASP_AHCLK_CLKSRC.MCASP5_AHCLKX_SOURCE |
| ATLP_CLK_MUX | SYSCLK19 | AUDIO_PLL_OUT | VIDEO_PLL_OUT_MUX | | | | | | MLB_ATL_CLKSRC.ATL_SOURCE |
| MCBSP_CLKS_MUX | PRCM Output MCBSP_CLKSEL Mux | AUD_CLKIN 0 | AUD_CLKIN 1 | AUD_CLKIN 2 | AUXOSC_XI | | | | MCBSP_CLKSRC.MCBSP_CLKS_SOURCE |
| MCBSP_CLKSEL | SYSCLK20 | SYSCLK21 | SYSCLK22 | CLKS | | | | | CM_AUDIOCLK_MCBSP_CLKSEL.CLKSEL. CLKSEL |
| McASP0 Aux Clk Mux | SYSCLK20 | SYSCLK21 | SYSCLK22 | | | | | | CM_AUDIOCLK_MCASP0_CLKSEL. CLKSEL |
| McASP1 Aux Clk Mux | SYSCLK20 | SYSCLK21 | SYSCLK22 | | | | | | CM_AUDIOCLK_MCASP1_CLKSEL. CLKSEL |
| McASP2 Aux Clk Mux | SYSCLK20 | SYSCLK21 | SYSCLK22 | | | | | | CM_AUDIOCLK_MCASP2_CLKSEL. CLKSEL |
| HDMI_I2S_clk_mux | PRCM output i2s_mclk | AUD_CLKIN 0 | AUD_CLKIN 1 | AUD_CLKIN 2 | AUXOSC_XI | | | | HDMI_I2S_CLKSRC.HDMI_I2S_SOURCE |

### 2.3.9 Timer Clock Structure

The output of RTCDIVIDER is also taken directly to optionally feed the WDT0 watchdog (Cortex™-A8 watchdog). The external 32KHz clock is not an option for WDT0. The other option for WDT0 is internal RCOSC32 oscillator. RTCDIVIDER output may also be optionally used as Cortex™-A8 Subsystem functional clock (via mux) for very low power operation (e.g., waiting in loop to bring clocks/system back up to full rate).

The following figures show the optional functional/reference clocks for DMTIMER1..8 and the functional/reference clocks for the watchdog timer (WDT0) and the Synctimer (which is available for generic use).



Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-13. Timer Clock Structure**

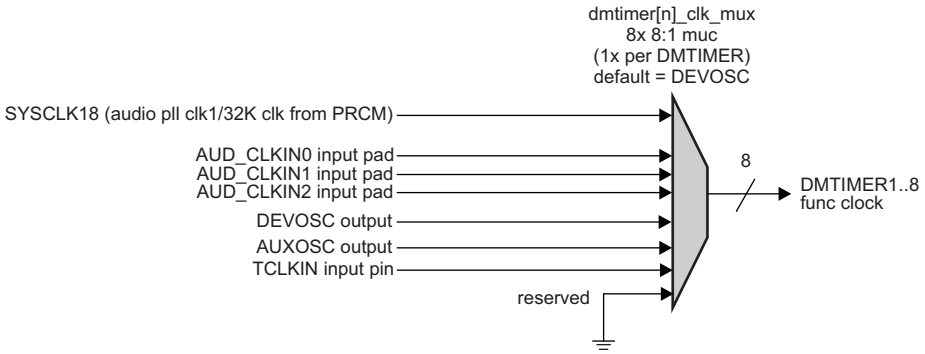
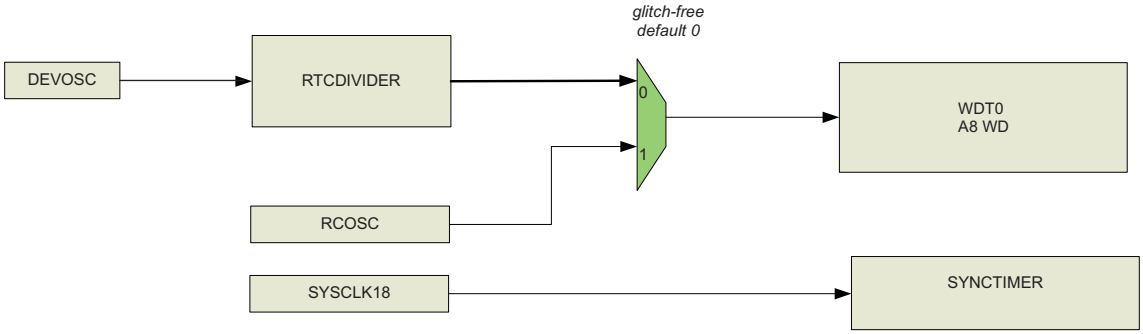



**Figure 2-14. Watchdog and Sync Timer Clock Structure**





Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-26. Timer Clock Structure**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Dmtimer1 Func Clk Mux | SYSCLK18 | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | DEVOSC output | AUXOSC_XI | TCLK Pin | Reserved | DMTIMER_CLKSRC.DMTIMER1_SOURCE |
| Dmtimer2 Func Clk Mux | SYSCLK18 | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | DEVOSC output | AUXOSC_XI | TCLK Pin | Reserved | DMTIMER_CLKSRC.DMTIMER2_SOURCE |
| Dmtimer3 Func Clk Mux | SYSCLK18 | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | DEVOSC output | AUXOSC_XI | TCLK Pin | Reserved | DMTIMER_CLKSRC.DMTIMER3_SOURCE |
| Dmtimer4 Func Clk Mux | SYSCLK18 | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | DEVOSC output | AUXOSC_XI | TCLK Pin | Reserved | DMTIMER_CLKSRC.DMTIMER4_SOURCE |
| Dmtimer5 Func Clk Mux | SYSCLK18 | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | DEVOSC output | AUXOSC_XI | TCLK Pin | Reserved | DMTIMER_CLKSRC.DMTIMER5_SOURCE |
| Dmtimer6 Func Clk Mux | SYSCLK18 | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | DEVOSC output | AUXOSC_XI | TCLK Pin | Reserved | DMTIMER_CLKSRC.DMTIMER6_SOURCE |
| Dmtimer7 Func Clk Mux | SYSCLK18 | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | DEVOSC output | AUXOSC_XI | TCLK Pin | Reserved | DMTIMER_CLKSRC.DMTIMER7_SOURCE |
| Dmtimer8 Func Clk Mux | SYSCLK18 | AUD_CLKIN0 | AUD_CLKIN1 | AUD_CLKIN2 | DEVOSC output | AUXOSC_XI | TCLK Pin | Reserved | DMTIMER_CLKSRC.DMTIMER8_SOURCE |
| WDT Clk mux | RTCDIVIDER_OUT | RCOSC 32K OUT | | | | | | | WDT0_CLKSRC.WDT_SOURCE |

### 2.3.10 Device Clocking from External Sources

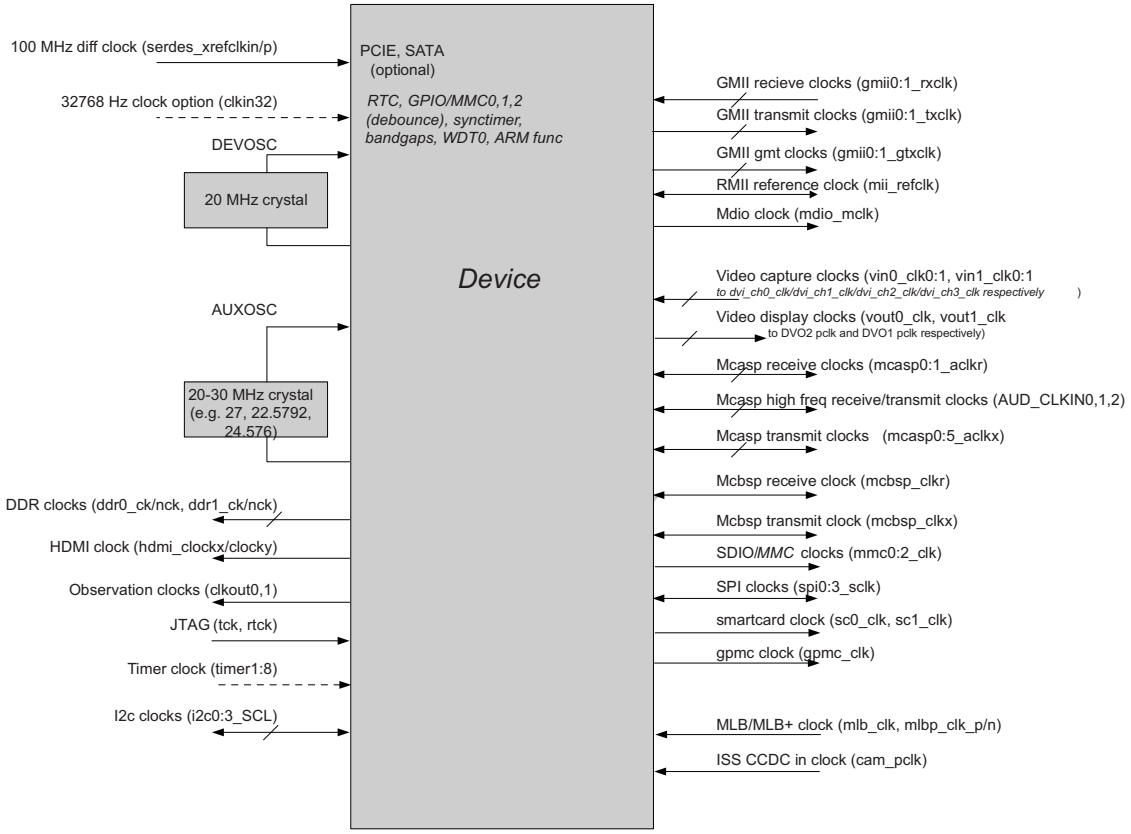
As shown in Figure 2-15, AUXOSC (20-30MHz) and DEVOSC (20 MHz) crystals are connected to generate the reference clocks. It is also possible to supply AUXOSC using an off-chip VCXO. A 100 MHz external differential clock must be used for PCIe. SATA uses 20 MHz oscillator as a reference.

Output clock SYSCLK_OUT can deliver clock outputs of each DPLL. The CM_CLKOUT_CTRL register controls the clock selection and division ratios.



Copyright © 2011–2016, Texas Instruments Incorporated

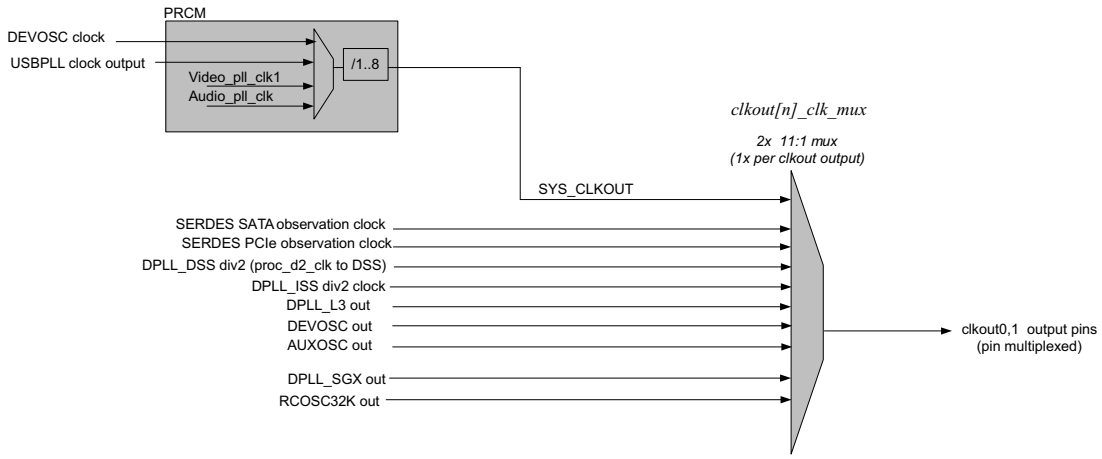
**Figure 2-15. Device Clocking External Clock Sources**




### 2.3.11 Clock Output Pin

The device has two clock output options. When chosen from the mux, the PLL clocks can be divided further before being exported out. The clock out is controlled by the register CM_CLKOUT_CTRL. The CLKOUT also has an enable option CM_CLKOUT_CTRL.CLKOUTEN.

**Figure 2-16. Clock Output Pin Options**





Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-27. Clkout Divider Options**

| Control Bit Filed | Divider | Supported Divide Ratios | Default Value |
|---|---|---|---|
| CM_CLKOUT_CTRL [CLKOUTDIV] | A | 1/1,1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 | 1/1 |

The clock outputs are controlled by two mux options respectively CLKOUT0 and CLKOUT1. Refer to the CLKOUT_MUX register for more details on the options of CLKOUT multiplexing.

## 2.4 Oscillator

As discussed in previous sections, the device has two conventional on-chip oscillators: DEVOSC and AUXOSC.

Figure 2-17 shows the conventional oscillator connections. DEVOSC is always needed, while AUXOSC is optional for video/audio reference and can run at any desired frequency in range of 20-30 MHz (e.g., typically could be 27 MHz or 22.5792 MHz). Table 2-28 and Table 2-29 show the settings for the two on-chip oscillators.

**Figure 2-17. Oscillator Connection**

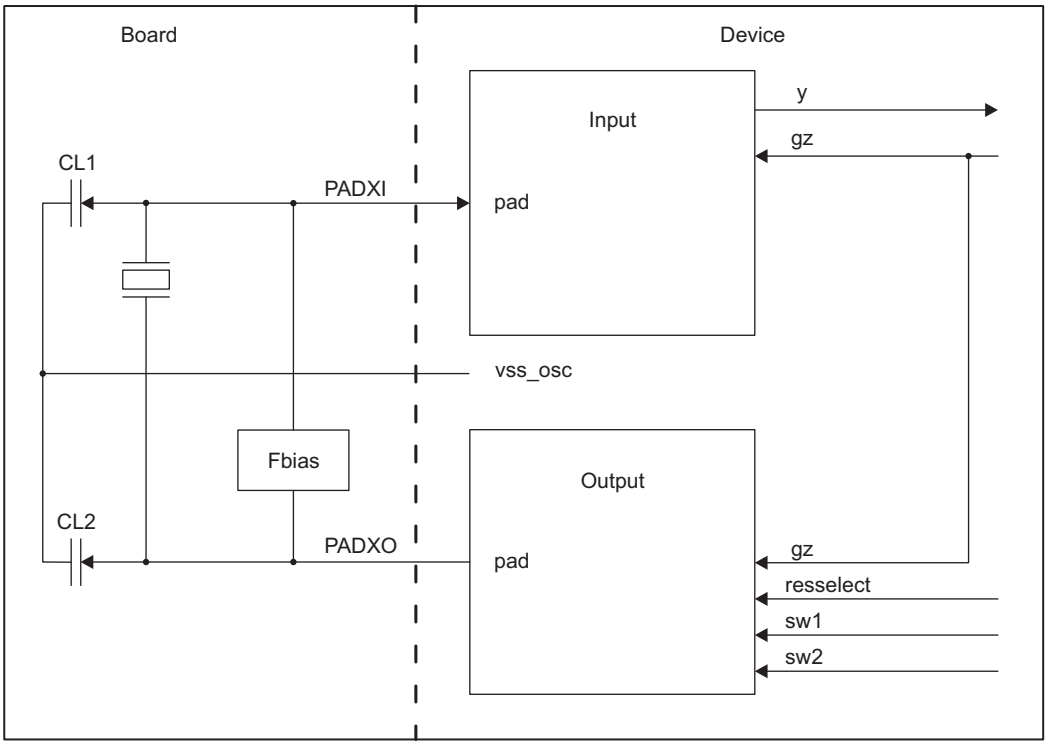



**Table 2-28. AUXOSC Oscillator Settings**

| Pin | Value |
|---|---|
| sw1 | 1 (select 15-35 MHz range) |
| sw2 | 0 (select 15-35 MHz range) |
| gz | MMR (Default 0 - Setting this to “1” disable the oscillator) |
| reselect | MMR (Default 0 - Use internal feedback resistor) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-29. DEVOSC Oscillator Settings**

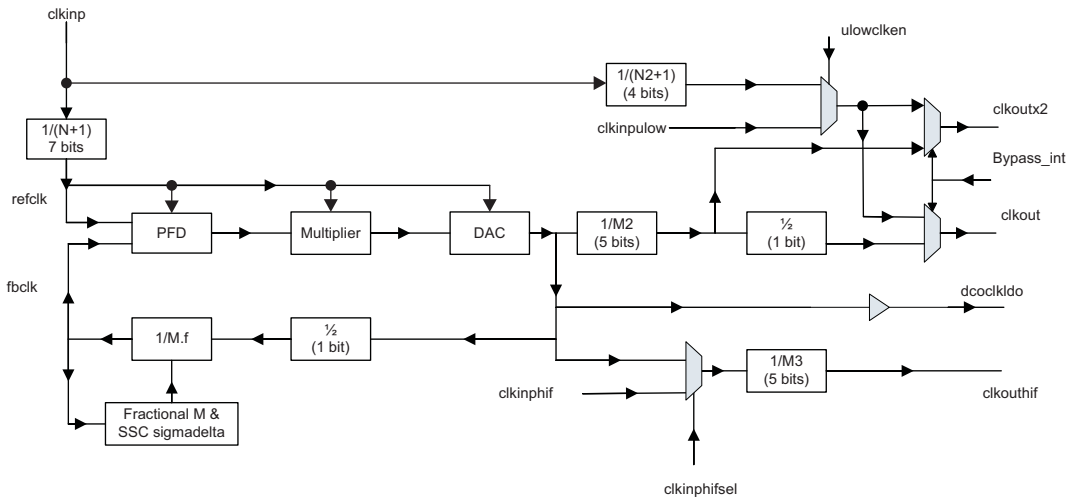
| Pin | Value |
|---|---|
| sw1 | 1 (select 15- 20 MHz range) |
| sw2 | 0 (select 15-20 MHz range) |
| gz | Controlled solely by Sleep control logic (Default 0 - sleep control logic sets to “1” to disable the oscillator while in sleep) |
| reselect | MMR (Default 0 - Use internal feedback resistor) |

**NOTE:** To enter sleep, the software must configure the desired sense of the wakeup (OSC_WAKE pin) and write to AUXOSC.GZ in the control module to force a hardware mechanism to turn off DEVOSC. The DEVOSC is cleanly re-started upon a wakeup event (rising or falling edge on the OSC_WAKE pin depending on configured sense before going into sleep). In case that an external oscillator/vcxo, a clock can directly be connected to XI pin and oscillator is put in bypass mode.

## 2.5 DPLLS

The DPLLS is a high resolution frequency synthesizer PLL with built in level shifters which allows the generation of frequencies upto 1GHz. DPLLLS has a pre-divide feature which allows the user to divide, for instance, a 20 MHz or 27 MHz reference clock to 1 MHz and then multiply up to 1 GHz maximum frequency. All PLLs come up in bypass mode at reset. SW needs to program all the PLL settings appropriately and then wait for PLL to be locked. Once it is locked, then PLLs can be taken out of bypass mode. The Cortex™-A8 PLL is of this type.

**Figure 2-18. DPLLS Clocks Structure**




The DPLLS has three input clocks:

- CLKINP : Reference input clock
- CLKINPULOW : Low frequency input clock for bypass mode only
- CLKINPHIF : High Frequency Input Clock for post-divider M3
- The DPLLS has four output clocks:
- CLKOUTHIF: High Frequency Output Clock from Post divider M3
- CLKOUTX2: Secondary 2x Output
- CLKOUT: Primary output clock
- DCOCLKLDO: Oscillator(DCO) output clock with no bypass

The DPLL has internal clocks: REFCLK (Internal reference clock) and BCLK.


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.5.1 DPLLS Frequency Range

The internal reference clock, REFCLK, is generated by dividing the input clock CLKINP by the programmed value N+1. The entire loop of the PLL runs on the REFCLK. Here, REFCLK = CLKINP/(N+1).

### 2.5.2 DPLLS Fractional M-divider Programming

The frequency multiplication factor M can be programmed as fractional. This is achieved by having a sigma delta feedback divider (M). A fractional value (FractionalM) of 18 bits is supported enabling control to an accuracy of 1ppm.

M is the software-configured Integer Feedback Multiplier Value.

N is the software-configured Input Divider Value.

M2 is the software-configured post-divider value

N2 is the software-configured Bypass divider value.

Frac M is software-configured Fractional Feedback Multiplier.

M3 is the software-configured additional divider

For more information for DPLLS programing and detailed descriptions of DPLLS registers see section *PLLSS Registers* in the device TRM.

### 2.5.3 DPLLS Clock Functions

The DPLL clock functions are shown in the tables Table 2-30:

**Table 2-30. Clock Functions**

| **Output Clocks in Locked Condition** | | |
|---|---|---|
| **REGM4XEN = '0'** | | |
| **Pin Name** | **Frequency** | **Comments** |
| CLKOUT | [M /(N+1)] * CLKINP * [1/M2] | |
| CLKOUTX2 | 2 * [M /(N+1)] * CLKINP * [1/M2] | |
| DCOCLKLDO | 2 * [M /(N+1)] * CLKINP | |
| CLKOUTHIF | CLKINPHIF / M3 | CLKINPHIFSEL = '1' |
| | 2 * [M /(N+1)] * CLKINP * [1/M3] | CLKINPHIFSEL = '0' |
| **REGM4XEN = '1'** | | |
| **Pin Name** | **Frequency** | **Comments** |
| CLKOUT | [4M /(N+1)] * CLKINP * [1/M2] | |
| CLKOUTX2 | 2 * [4M /(N+1)] * CLKINP * [1/M2] | |
| DCOCLKLDO | 2 * [4M /(N+1)] * CLKINP | |
| CLKOUTHIF | CLKINPHIF / M3 | CLKINPHIFSEL = '1' |
| | 2 * [4M /(N+1)] * CLKINP * [1/M3] | CLKINPHIFSEL = '0' |
| **Output Clocks before Lock & during Relock modes** | | |
| **Pin Name** | **Frequency** | **Comments** |
| CLKOUT | CLKINP/(N2+1) | ULOWCLKEN = '0' |
| | CLKINPULOW | ULOWCLKEN = '1' |
| CLKOUTX2 | CLKINP/(N2+1) | ULOWCLKEN = '0' |
| | CLKINPULOW | ULOWCLKEN = '1' |
| DCOCLKLDO | Low | |
| CLKOUTHIF | CLKINPHIF/M3 | CLKINPHIFSEL = '1' |
| | Low | CLKINPHIFSEL = '0' |


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** Since M3 divider is running on the internal LDO domain, in the case when CLKINPHIFSEL='1,' CLKOUTHIF could be active only when internal LDO is ON. Hence, whenever LDOPWDN goes low to high to powerdown LDO (happens when TINITZ activated / when entering slow relock bypass mode), output CLKOUTHIF glitch and stop. To avoid this glitch, it is recommended to gate CLKOUTHIF using control CLKOUTHIFEN before asserting TINITZ entering any slow relock bypass mode.

The Min and Max Values of the Clock generation supported by PLLs across different OPPs.

**NOTE:** For detailed information about PLL frequency limits at all OPPs see section *PLL Frequency Limits* in the device *Datasheet*.

### 2.5.4 DPLLS Frequency Factors

The divide values have to be selected with the ranges in the Table 2-31 so as to generate the required CLOCKOUT in the Frequency Ranges mentioned in the previous sections are satisfied.

**Table 2-31. Frequency Factors**

| Feedback Multiplier | M[1] | 2 - 2047 | 00000000010 - 11111111111 |
|---|---|---|---|
| **Input Divider** | **N + 1** | 1 - 128 | (0000000 - 1111111) + 1 |
| **Bypass Divider** | **N2 + 1** | 1 - 16 | (0000 - 1111) + 1 |
| **Post Divider** | **M2**[2] | 0 - 31 | 00000 - 11111 |
| **Additional Divider** | **M3**[3] | 1 - 31 | 00001 - 11111 |

(1) M = 0 and 1 correspond to MN Bypass mode setting.
(2) When M2 = 0, the M2 post divider is bypassed.
(3) M3 = 0 is an invalid setting.

## 2.6 DPLLLJ

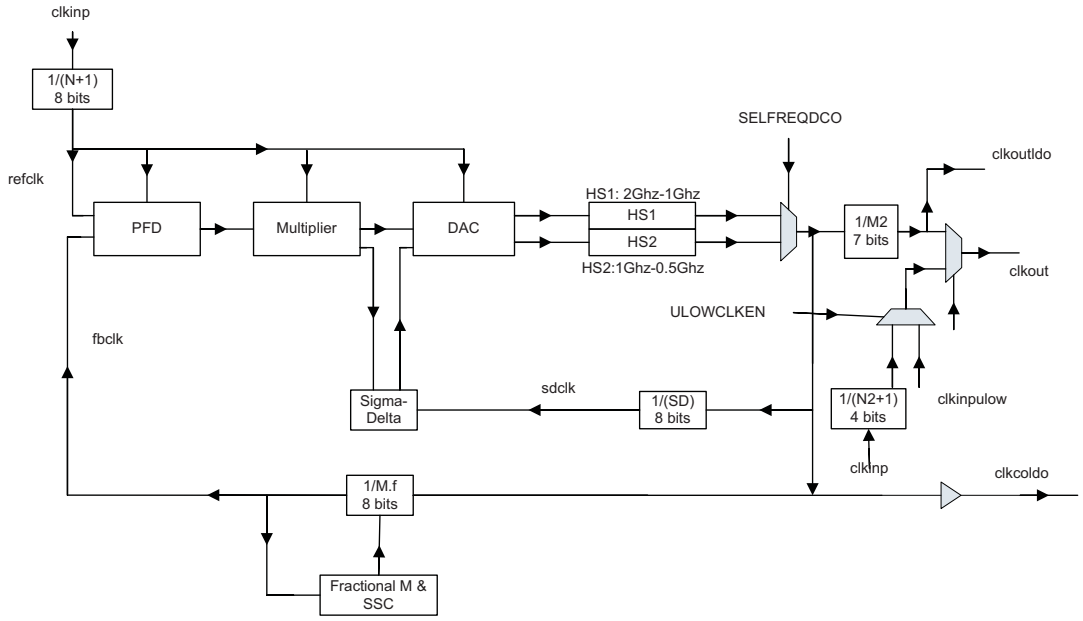
The DPLLLJ is a low-jitter PLL with a 2GHz maximum output. The DPLLLJ has a pre-divide feature which allows users to divide, for instance, a 20 MHz or 27 MHz reference clock to 1 MHz and then multiply up to 2 GHz maximum.

The DPLLLJ also has an 18b fractional multiplier which is useful for configuring very specific frequencies as desired (for example, audio and video).

All PLLs come up in bypass mode at reset. The software needs to program all the DPLL settings appropriately and then wait for theDPLL to be locked. Once the DPLL is locked, then DPLLs can be taken out of bypass mode.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-19. DPLLJ Basic Structure**




The following PLLs belong to type DPLLJ:

- DSPPLL
- SGXPLL
- HDVICPPLL
- L3PLL
- ISSPLL
- DSSPLL
- VIDEO0PLL
- VIDEO1PLL
- HDMIPLL
- AUDIOPLL
- USBPLL
- DDRPLL

The DPLL has two input clocks:

- CLKINP: Reference input clock
- CLKINPULOW: Bypass input clock

The DPLL has two internal clocks: REFCLK (Internal reference clock): This is generated by dividing the input clock CLKINP by the programmed value N+1. The entire loop of the PLL runs on the REFCLK. Here, REFCLK = CLKINP/(N+1)

DCOCLK (Internal Oscillator clock.): This is the raw clock directly out of the digitally controlled oscillator (DCO) before the post-divider. The PLL output clock is synthesized by an internal oscillator which is phase locked to the REFCLK. There are two oscillators built within DPLLLJ. The oscillators are user selectable based on the synthesized output clock frequency requirement. In locked condition, DCOCLK= CLKINP *[M/(N+1)].

The DPLL has three external output clocks:

- CLKOUTLDO
  Primary output clock in VDDLDOOUT domain. Bypass option not available on this output
  CLKOUTLDO = (M / (N+1))*CLKINP*(1/M2)


Copyright © 2011–2016, Texas Instruments Incorporated

- CLKOUT
  Primary output clock on digital core domain
  CLKOUT = (M / (N+1))*CLKINP*(1/M2)
- CLKDCOLDO
  Oscillator (DCO) output clock before post-division in VDDLDOOUT domain. Bypass option is not available on this output.
  CLKDCOLDO = (M / (N+1))*CLKINP

All clock outputs of the DPLL can be gated. The Control module provides the DPLL with a clock gating control signal to enable or disable the clock, and the DPLL provides the PRCM module with a clock activity status signal to let the PRCM module hardware know when the clock is effectively running or effectively gated. Output clock gating control for various clockouts: CLKOUTEN/CLKOUTLDOEN/CLKDCOLDOEN.

### 2.6.1 DPLLLJ Fractional M-divider Programming

The frequency multiplication factor M can be programmed as fractional. This is achieved by having a sigma delta feedback divider (M). A fractional value (FractionalM) of 18 bits is supported enabling control to an accuracy of 1ppm.

M is the software-configured Integer feedback multiplier value.

N is the software-configured input divider value.

M2 is the software-configured post-divider value

N2 is the software-configured bypass divider value.

SD is the software-configured Sigma2Delta divider.

Frac M is software-configured fractional feedback multiplier.

For more information for DPLLs programing and detailed descriptions of DPLLSJ registers see section *PLLSS Registers* in the device TRM.

### 2.6.2 DPLLLJ Clock Functions

The clock functions are shown in Table 2-32 and Table 2-33 the tables below.

**Table 2-32. Output Clocks in Locked Condition**

| PINNAME | FREQUENCY |
|---|---|
| CLKOUT | [M /(N+1)] * CLKINP * [1/M2] |
| CLKOUTLDO | [M /(N+1)] * CLKINP * [1/M2] |
| CLKDCOLDO | [M /(N+1)] * CLKINP |

**Table 2-33. Output Clocks Before Lock and During Relock Modes**

| PINNAME | FREQUENCY | Conditions |
|---|---|---|
| CLKOUT | CLKINP/N2+1 | ULOWCLKEN = ’0’ |
| | CLKINPLOW | ULOWCLKEN = ’1’ |
| CLKOUTLDO | LOW | |
| CLKDCOLDO | LOW | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.6.3 DPLLLJ Frequency Range (MHz)

The Min and Max Values of the Clock generation supported by PLLs across different OPPs.

**NOTE:** For detailed information about PLL frequency limits at all OPPs see section *PLL Frequency Limits* in the device *Datasheet*.

### 2.6.4 DPLLLJ Frequency Factors

The divide values have to be selected with the ranges in the Table 2-34 so as to generate the required CLOCKOUT in the Frequency Ranges mentioned in the previous sections are satisfied.

**Table 2-34. DPLLLJ Frequency Factors**

| Multiplier/Divider | | Range | |
|---|---|---|---|
| Feedback Multiplier (Integer) | M | 2 - 4095 | 00000000010 – 11111111111 |
| Feedback Multiplier (Fractional) | Frac M | | 000000000000000000 - 111111111111111111 |
| Input Divider | N+1 | 1-256 | (00000000 – 11111111) + 1 |
| Post Divider | M2[(1)] | 1-127 | 0000001 - 1111111 |
| Sigma2Delta Divider | SD | 2-255 | 000000010-11111111 |
| Bypass Divider | N2+1 | 16-1 | (0000 – 1111) + 1 |

(1) The M2 divider is bypassed when the divisor is equal to 0 or 1.

### 2.6.5 Mode of Operations

**Table 2-35. Modes of Operations**

| Mode | Control Signal | CLKINP | REFCLK | Digital | Analog Bias | DCO OSC | CLKOUT | CLKDCOLDO(CLKOUTLDO) | FreqLock time (Phaselock time) from specific mode to "Active & Locked mode" |
|---|---|---|---|---|---|---|---|---|---|
| OFF | SYSRESET = 1 or TINITZ = 0 | ON or OFF | OFF | ON | OFF | OFF | Bypass clk | LOW | 350REFCLKs (500REFCLKs) |
| Active & Locked | SYSRESET = 0 TINITZ = 1 IDLE = 0 | ON | CLKINP /(N+1) | ON | ON | ON | M/(N+1) * CLKINP * (1/M2) | DCOCLK (DCOCLK/M2) | Locked |
| Idle Bypass low power | SYSRESET = 0 TINITZ = 1 IDLE = 1 | ON | OFF | ON | OFF | OFF | Bypass clk | LOW | 20REFCLKs (150REFCLKs) |
| Lossclk – Bypass low power | SYSRESET = 0 TINITZ = 1 | OFF | OFF | ON | OFF | OFF | Bypass clk | LOW | 20REFCLKs (150REFCLKs) |
| Standby Retention-Bypass low power | STBYRET= 1 ULOWCLK EN=0/1 | ON | OFF | ON | OFF | OFF | Bypass clk | LOW | 7.5us+30REFCLKs (7.5us+125 REFCLKs) |
| MN Bypass | M = 0 or 1 | ON | OFF | ON | OFF | OFF | Bypass clk | LOW | 350REFCLKs (500REFCLKs) |
| Retention | Enter into RETENTION from Idle Bypass Mode | | | OFF | OFF | OFF | Low | LOW | 7.5us + 30REFCLKs (7.5us+125 REFCLKs) |


Copyright © 2011–2016, Texas Instruments Incorporated

**TINITZ:** TINITZ activation (Low) gives softreset to DPLLJ. TINITZ does not reset the entire digital control logic. It forces the FSM into RESET State so that DPLLJ could restart. DPLLJ starts calibration and lock sequence when TINITZ is disabled (Low to High).

**TENABLEDIV:** TENABLEDIV rising edge loads the values of M2REG and N2REG into PLL register (PLL_M2NDIV). It is not necessary to recalibrate PLL with the updated values. The change in values of M2REG reflects appropriately in the output clocks CLKOUT and CLKOUTLDO.

**OFF MODE:** SYSRESET= 1 or TINITZ = 0 PLL is in off mode when SYSRESET/TINITZ is configured to the values SYSRESET= 1 or TINITZ = 0. CLKOUT is bypass clk and CLKOUTLDO, and CLKDCOLDO are low irrespective of the presence of the inputclock.

**ACTIVE and Locked Mode:** SYSRESET = 0 & TINITZ = 1 , IDLE = ‘0’ PLL is in Active and locked mode when SYSRESET = 0 & TINITZ = 1, IDLE = ‘0’

- TENABLE rising edge loads the values of MREG and NREG into the ADPLLJ registers.
- TENABLEDIV rising edge loads the values of M2REG and N2REG into the ADPLLJ register.
- ADPLLJ starts calibration and lock sequence begins from the first CLKINP edge when TINITZ is disabled (low to high).
- TINITZ rising edge should follow the TENABLE rising edge, so that ADPLLJ performs the calibration and lock sequence after the loop control values are loaded.
- The change in values of M2REG reflects appropriately in the output clocks CLKOUT and CLKOUTLDO.
- Following this internal calibration sequence is performed followed by locking sequence. ADPLLJ has two lock outputs viz. FREQLOCK and PHASELOCK.
- The FREQLOCK output indicates frequency lock. FREQLOCK is asserted when internally the normal clk frequency is within + 1 / 2% of the final output frequency. The PHASELOCK output indicates phase tracking between output clocks (CLKOUT, CLKOUTLDO and CLKDCOLDO) and input clock (CLKINP). The FREQLOCK, PHASELOCK gets asserted in the ADPLLJ status register. The ouptut clocks CLKOUT, CLKOUTLDO and CLKDCOLDO are generated as per the register values (M, N, M2, N2) values.

**IDLE Bypass:** TINITZ =1 IDLE =1 The IDLE input signal can force ADPLLJ into bypass mode. Also the internal reference clock REFCLK is gated inside the ADPLLJ digital control logic to save power. In the IDLE mode, the PHASELOCK and FREQLOCK output signals are asserted low.CLKOUT is bypass clk and CLKOUTLDO, CLKDCOLDO are low

### *2.6.6 BYPASS Mode*

After digital power-up and before SYSRESET is active (high), DPLL is in an undefined state. Hence, power-up with SYSRESET active (high). On SYSRESET = ‘1’, VDD should be active, PONIN = ‘1’, PGOODIN = ‘1’ as shown in <MODULE>PLL_PWRCTRL register in PRCM and also the acknowledge signals PONOUT = ‘1’ and PGOODOUT = ‘1’ shown in <MODULE>PLL_STATUS register of PRCM.

CLKOUT is in the bypass mode and would give out bypass clk during and after sysreset.

<MODULE>PLL_CLKCTRL.ULOWCLKEN = ‘1’ – BYPASS CLK = CLKINPULOW

<MODULE>PLL_CLKCTRL.ULOWCLKEN = ‘0’ – BYPASS CLK = CLKINP/(N2+1); N2 =0

The bypass mode CLKOUT could be kept active or inactive during sysreset by controlling the status of <MODULE>PLL_CLKCTRL. CLKOUTEN during SYSRESET = ‘1’.

<MODULE>PLL_CLKCTRL.CLKOUTEN = ‘1’ during SYSRESET = ‘1’ asynchronously enables CLKOUT. And the acknowledge bit in the register is <MODULE>PLL_STATUS.CLKOUTENACK = ‘1’

<MODULE>PLL_CLKCTRL.CLKOUTEN = ‘0’ during SYSRESET = ‘1’ asynchronously disables CLKOUT. And the acknowledge bit in the register <MODULE>PLL_STATUS.CLKOUTENACK = ‘0’

CLKOUTLDO and CLKDCOLDO are low during and after SYSRESET.

Check the status bits CLKOUTENACK, BYPASS, BYPASSACK in the status register <MODULE>PLL_STATUS in the PRCM.


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.6.7 M2 & N2 Change On-the-Fly

The dividers M2 and N2 are designed to change on the fly. The output clock CLKOUT/ CLKOUTLDO perform a clean switch from OLD period to NEW period without any clock period being different from OLD or NEW period.

### 2.6.8 Gating and Power Down of Output Clocks

The high to low transition of the input signal CLKOUTEN/ CLKOUTLDOEN/ CLKDCOLDOEN performs synchronous gating of the output clock CLKOUT/ CLKOUTLDO/ CLKDCOLDO respectively.The gating acknowledge output CLKOUTENACK/ CLKOUTLDOENACK/CLKDCOLDOENACK is asserted Low once the gating is complete.

The low to high transition of the input signal CLKOUTEN/ CLKOUTLDOEN/ CLKDCOLDOEN performs synchronous enabling of the output clock CLKOUT/ CLKOUTLDO/ CLKDCOLDO.The gating acknowledge output CLKOUTACK/ CLKOUTLDOACK/CLKDCOLDOACK is asserted High once the enabling is complete.

### 2.6.9 M2PWDNZ / CLKDCOLDOPWDNZ bits

<MODULE>PLL_CLKCTRL. M2PWDNZ active (low) will powerdown the M2 divider. Hence, the output clock CLKOUT is in bypass mode. Also, even after DPLL has locked and <MODULE>PLL_STATUS. FREQLOCK has gone high, and <MODULE>PLL_STATUS. BYPASSACK output continues to remain high. Changing this bit in active locked condition is not supported.

INITIALIZATION should follow the change of these bits <MODULE>PLL_CLKCTRL.CLKDCOLDOPWDNZ / CLKOUTLDOPWDNZ active (low) will powerdown the CLKDCOLDO/CLKOUTLDO output. CLKDCOLDO/CLKOUTLDO is Low in this condition. Changing these bits in active locked condition is not supported. INITIALIZATION should follow change of these bits.


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.6.10 Connected Outputs of DPLLS

| PLL Instance | CLKOUT | CLKOUTX2 | DCOCLKLDO | CLKOUTHIF |
|---|---|---|---|---|
| DPLL_ARM | Connected to FCLK of Cortex™-A8 | Not Connected | Not Connected | Not Connected |

### 2.6.11 Connected Outputs of DPLLLJ

| PLL Instance | CLKOUT | CLKDCOLDO | CLKOUTLDO |
|---|---|---|---|
| DPLL_DSP | Functional C674x™ Clocks | Not Connected | Not Connected |
| DPLL_SGX | Functional SGX Clk | Not Connected | Not Connected |
| DPLL_HDVICP | Functional HDVICP Clk | Not Connected | Not Connected |
| DPLL_L3 | L3/L4 Clocks | Not Connected | Not Connected |
| DPLL_ISS | Functional ISS Clk | Not Connected | Not Connected |
| DPLL_VIDEO0 | Video PLL input clock to PRCM | Not Connected | Not Connected |
| DPLL_VIDEO1 | Video PLL input clock to PRCM | Not Connected | Not Connected |
| DPLL_DSS | Functional DSS Clock | Not Connected | Not Connected |
| DPLL_HDMI | DVO1 | HFBITBLK | HFBYPASSCLK |
| DPLL_AUDIO | Audio PLL Clock | Not Connected | Not Connected |
| DPLL_USB | PRCM Input for 192 Mhz(/5 option) | USB Phy Clock(960 Mhz) | Not Connected |
| DPLL_DDR | DDR0_PHY Clock, DDR1_PHY Clock, DMM clock and EMIF mclk. | Not Connected | Not Connected |

## 2.7 Reset Management

### 2.7.1 Overview

The PRCM manages the reset signals to all power domains. And, it manages generation of a single reset output signal through device pin, SYS_RESWARM_RST, for external use. The PRCM has no control over resets generated locally within a module, e.g., via the OCP configuration register bit <IPName>_SYSCONFIG.SOFTRESET.

All PRM reset outputs are asynchronously asserted. These outputs are active-low except for the DPLL resets. Deassertion is synchronous to the clock which runs a counter used to stall, or delay, reset de-assertion upon source deactivation. This clock is SYS_CLK used by all the reset managers. All modules receiving a PRCM generated reset are expected to treat the reset as asynchronous and implement local re-synchronization upon de-activation as needed.

One or more reset managers are required per power domain. Independent management of multiple reset domains is required to meet the reset sequencing requirements of all modules in the power domain.

### 2.7.2 Reset Concepts and Definitions

The PRCM manages many sources of reset signals with different characteristics. Reset signals are categorized as:

- Scope: Global or Local reset.
  - Global reset: Affects the entire logic of the device; all modules are reset. Generally, occurs when the device powers up or an abnormal operation is detected (the eFuse bad device is detected, etc.).
  - Local reset: Affects one power domain, reset domain, or module. Generally, when a power domain transitions from off mode to active mode, or when a software-reset control bit for a domain is set, only the group of modules within that domain is affected.


Copyright © 2011–2016, Texas Instruments Incorporated

- Occurrence: Cold or Warm reset.
  - Cold reset: global reset that affects every module in the device. It usually corresponds to the initial power-on reset.
  - Warm reset: A module may use a warm reset to reset a subset of its logic.
- Source type: Software or Hardware triggered reset
  - Software reset: Triggered by setting a bit in a configuration register of the PRCM module.
  - Hardware reset: Triggered by a signal from a hardware module inside or outside the PRCM module

Each reset source is specified as being a cold or warm type. Cold types sources are applied globally within each receiving entity (i.e., subsystem, module, macro-cell) upon assertion. Cold reset events include: device power-up, power-domain power-up, and E-Fuse programming failures.

Warm reset types are not necessarily applied globally within each receiving entity. This is often done to speed-up reset recovery time, i.e., the time to transition to a safe operating state, compared to the time required upon receipt of a cold reset. Warm reset events include: software initiated per power-domain, watch-dog time-out, externally triggered, and emulation initiated.

Reset sources, warm or cold types, intended for device-wide effect are classified as global sources. Reset sources intended for regional effect are classified as local sources.

Each reset manager provides two reset outputs. One is a cold reset generated from the group of global and local cold reset sources it receives. The other is a warm+cold reset generated from the combined groups of, global and local, cold and warm reset sources it receives.

The reset manager asserts one, or both, of its reset outputs asynchronously upon reset source assertion. Reset deassertion is extended beyond the time the source gets de-asserted. The reset manager will then extend the active period of the reset outputs beyond the release of the reset source, according to the PRCM's internal constraints and device's constraints. Some reset durations can be software-configured. Most (but not all) reset sources are logged by PRCM's reset status registers. The same reset output can generally be activated by several reset sources and the same reset source can generally activate several reset outputs. All the reset signals output of the PRCM are active low. Several conventions are used in this document for signal and port names. They include:

- "_RST" in a signal or port name is used to denote reset signal.
- "_PWRN_RST" in a signal or port name is used to denote a cold reset source


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.7.3 Power On Reset (PORz)

The source of power on reset is the PORz pin on the device. Everything on device is reset with assertion of power on reset. This reset is non-blockable. Power on Reset is also referred to as a cold reset since it is required to be asserted when the device goes through a power-up cycle. However, a device power-up cycle is not required to initiate a Power-on Reset.

### 2.7.4 External Warm Reset (RESETz)

The source for this reset is the RESETz pin on the device. This signal resets everything except test and emulation logic, and efuse to provide a clean starting stage. This reset is also optionally blocked to the GMAC switch and it's reference clock source PLL (controlled in protected control module MMR). This reset DOES reset PLL, PRCM and EMIF. This reset is different from POR as during Warm reset, reset controller can assume that clocks and power to the chip are stable from assertion through deassertion, whereas for POR they can become stable during assertion. This reset source is fully blockable only via emulation.

### 2.7.5 Emulation Cold Reset (ICEPICK POR)

The source for this reset is on-chip emulation logic. This reset is same as POR reset. Only difference is that ICEPICK POR reset does not affect emulation logic and efuse.

### 2.7.6 Emulation Warm Reset

The source for this reset is on-chip emulation logic. This reset is same as external Warm reset (RESETz).

### 2.7.7 Reset Requestor

There is a watchdog timer on the device that can act as a reset requestor. This reset is not blockable. The watchdog is optionally combined with other device-level resets based on bootstrap latch and goes as chip output as board reset on the RSTOUT_WD_OUT pin. If BTMODE[11] (from gpmc_ad11 pin) is 0, then RSTOUT is asserted when a Watchdog Timer reset, POR, RESET, or Emulation/Software-Global Cold/Warm reset occurs. If BTMODE[11] is 1, then the RSTOUT_WD_OUT is asserted only when a Watchdog Timer reset occurs. The register PRM_RSTST in the PRCM logs the global reset sources. Each bit is set upon release of the domain reset signal. Must be cleared by software.

### 2.7.8 Software Controlled Reset

The device also has software controlled reset. There is three types of software resets:

- SW Global cold reset (SWGCR)
- SW Global warm reset (SWGWR)
- Local reset (for IPs with CPUs)

### 2.7.9 Test Reset (TRSTz)

This reset is triggered from TRSTz pin on JTAG interface. This is a non-blockable reset and it resets test and emulation logic.

**NOTE:** A PORz reset assertion should cause entire device to reset including all test and emulation logic regardless of the state of TRSTz. Therefore, PORz assertion achieves full reset of the device even if TRSTz pin is pulled permanently high and no special toggling of TRSTz pin is required during power ramp to achieve full POR reset to the device. Further, it should also be perfectly acceptable for TRSTz input to be pulled permanently low during normal functional usage of the device in the end-system to ensure that all test and emulation logic is kept in reset.


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.7.10 Reset Priority

If more than one of these reset sources are asserted simultaneously then following priority order should be used:

1. POR
2. TRSTz
3. External Warm reset
4. Emulation
5. Reset requestors
6. Software resets

### 2.7.11 Reset Characteristics

Table 2-36 shows the characteristics of each reset source.

**Table 2-36. Reset Sources**

| Characteristics | Pin POR (PORz) | SW Global Cold Reset | Emu Cold Reset (Icepick POR) | Bad Device (eFUSE) | Pin Warm Reset (RESETz) | Emu Warm Reset | Watchdog Timer | SW Global Warm Reset | TRSTz |
|---|---|---|---|---|---|---|---|---|---|
| Resets PLLs and enable bypass mode | Y | Y | Y | Y | Y | Y | Y | Y | N |
| Resets Dividers | Y | Y | Y | Y | Y | Y | Y | Y | N |
| Resets Fusefarm | Y | N | N | N | N | N | N | N | N |
| Resets chip functional logic | Y | Y | Y | Y | Y | Y | Y | Y | N |
| Resets GMAC switch and related chip logic | Y | N(1) | Y | Y | N(1) | N(1) | N(1) | N(1) | N |
| Resets test and emulation logic | Y | Y | N | N | N | N | N | N | Y |
| Resets Memory (DDR controller/PHY) | Y | Y | Y | Y | Y | Y | Y | Y | N |
| Resets power state for each IP | Y | Y | Y | Y | Y | Y | Y | Y | N |
| Reset source blockable | N | Y | N | N | Y | Y | Y (only emulation) | Y | N |
| Config pins latched4 | Y | N | N | N | Y | N | N | N | N |
| Puts Ios in tristate | Y | Y | Y(2) | Y(2) | Y(2) | Y(2) | Y(2) | Y(2) | N |
| Pin mux registers | Y | Y | Y(3) | Y(3) | Y(3) | Y(3) | Y(3) | Y(3) | N |
| RSTOUTn pin asserted | Y(4) | Y(4) | Y(4) | Y(4) | Y(4) | Y(4) | Y | Y(4) | N |

(1) Only true if GMAC switch reset isolation is enabled in control registers, otherwise is in reset.
(2) Some special IOs like test and emulation related is not in reset.
(3) Some special IOs like test and emulation related muxing is not in reset.
(4) RSTOUT_WD_OUT pin asserted only in case BTMODE[11] = 0 (latched from gpmc_ad11 pin). For pin reset conditions (PORz and RESETz), the pin is asserted only after the bootstrap value is latched (therefore asserts during internal stretched version of the reset, but not during assertion of the external pin).

**NOTE:** The Bandgap macros for the LDO's is in reset on PORz only.

### 2.7.12 Trace Functionality Across Reset

Other than POR, TRSTz and SW cold reset, none of the other resets affect trace functionality. This requires that Debug_SS has required reset isolation for the trace logic. Also the IOs and muxing control (if any) for trace IOs should not get affected by any other reset. Since none of the PLLs getting reset with other resets, clocks are any way stable.


Copyright © 2011–2016, Texas Instruments Incorporated

## 2.7.13 Reset Isolation

### 2.7.13.1 PCIe Reset Isolation

The device supports reset isolation for the PCI Express IP in the limited fashion as described below. It means that the user should be able to control the chip warm reset and PCI express in-band reset independently.

1. **RC (Root Complex), PCIE_SS reset and chip not reset**
   Local reset is asserted to the PCIE_SS from the PRCM. SW needs to make sure that there were no ongoing transactions at this point of time. This can be ensured by first taking the PCIE_SS to IDLE state. After coming out reset the bus enumeration needs to be performed again. Driver SW should behave as if all downstream devices are just connected again
2. **RC (Root Complex), PCIESS not reset and chip reset**
   In Root Complex mode, the chip is master of the PCI Express network. There is no utility of keeping PCIE_SS running if the chip is rebooting. All state information is lost and it is simpler to restart. When resuming, the Root Complex can issue Hot Reset to all downstream devices and start with the bus enumeration stage.
3. **EP (Endpoint), PCIE_SS reset and chip not reset**
   This can happen due to an in-band reset or because of PCIERST getting asserted. PCIE_SS indicates this by interrupting the Cortex™-A8 core. SW should then take the PCIE_SS to IDLE state and then local reset can be asserted to the PCIE_SS.
4. **EP (Endpoint), PCIE_SS not reset and chip reset**
   This is not supported as PCIE is a point to point connection and resetting one endpoint does not impact the complete system. So whenever chip gets reset even PCIE_SS gets reset.

### 2.7.13.2 EMAC Switch Reset Isolation

The device supports reset isolation for the Ethernet Switch IP by register RESET_ISO[0] ISO_CONTROL in the control module. This allows the device to undergo a warm reset without disrupting the switch or traffic being routed through the switch during the reset condition.

If configured by registers in control module that EMAC reset isolation ISO_CONTROL is active, then behavior is as follows:

Any “Warm” reset source (except the software warm reset) is blocked to the EMAC switch logic in the IP (the IP has two reset inputs to support such isolation) and to PLL (and it’s control bits) which is sourcing the EMAC switch clocks as required by the IP (50 or 125MHz reference clocks). Also, the EMAC switch related IO pins must retain their pin muxing and not glitch (continuously controlled by the EMAC switch IP) by blocking reset to the controlling register bits.

If configured by registers in control module that EMAC reset isolation ISO_CONTROL is NOT active (default state), then the warm reset sources are allowed to propagate as normal including to the EMAC Switch IP (both reset inputs to the IP).

All cold or POR resets are always propagate to the EMAC switch IP as normal (as otherwise defined in this document).

## 2.7.14 Reset Output ($\overline{\text{RSTOUT_WD_OUT}}$)

The device has one pin $\overline{\text{RSTOUT_WD_OUT}}$ which reflects chip reset status. This output is always assert asynchronously when any chip watchdog timer reset occurs (any of the 3 chip watchdogs) and if BTMODE[11] bootstrap is 0, then additionally the pin asserts if any of the following reset events occurs:

1. POR (only internal stretched portion of reset event after bootstrap is latched)
2. External Warm reset ($\overline{\text{RSTOUT_WD_OUT}}$ pin, only internal stretched portion of reset event after bootstrap is latched)
3. Emulation reset (Cold or warm from ICEPICK)
4. Reset requestor
5. SW Cold/Warm reset
   This output remains asserted as long as PRCM keeps reset to the host processor asserted.


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:**

- The RSTOUT_WD_OUT pin is output only and not bidirectional.
- TRST does not cause RSTOUT_WD_OUT assertion

### 2.7.15 Power-on-Reset (PORz) Sequence

The following is the sequence of actions for PORz:

1. PORz pin at chip boundary gets asserted (goes low).

**NOTE:** The state of the Reset Output during PORz assertion should be a don't care, it should not affect PORz (only implication is if they are both asserted and RSTOUT_WD_OUT is deasserted after PORz gets re-latching of boot config pins and may see warm RESET flag set in PRCM versus POR).

2. All IOs go to tri-state immediately.
3. When power comes-up, PORz value propagates to the PRCM.
4. PRCM fans out reset to the complete chip and all logic which uses async reset gets reset. RSTOUT_WD_OUT goes low to indicate reset-in-progress.
5. External clocks start toggling and PRCM propagates these clocks to the chip keeping PLLs in bypass mode.
6. All logic using sync reset gets reset.
7. When power and clocks to the chip are stable, PORz is de-asserted.
8. Boot Config pins are latched upon de-assertion of PORz pin
9. FuseFarm reset is de-asserted to start eFuse scanning.
10. Once eFuse scanning is complete, reset to the host processor and to all other peripherals (peripherals without local processor) is de-asserted.
11. RSTOUT_WD_OUT is de-asserted.
12. Once host processors finish booting then all remaining peripherals see reset de-assertion.

**NOTE:** All IPs with local CPUs have local reset asserted by default at PORz and reset de-assertion requires host processor to write to corresponding registers in PRCM.

### 2.7.16 RESETz Sequence

The following is the sequence of actions for RESETz:

1. RSTOUT_WD_OUT pin at chip boundary gets asserted (goes low).

**NOTE:** For RESETz sequence to work as described, it is expected that PORz pin is always inactive, otherwise refer to PORz functionality as described in previous section.

2. All IOs (except test and emulation) go to tri-state immediately.
3. PRCM propagates reset to the complete chip and all logic. RSTOUT_WD_OUT goes low to indicate reset-in-progress.
4. Chip clocks are not affected as both PLL and dividers are intact.
5. RESETz gets de-asserted after 30 cycles
6. Boot Config pins are latched upon de-assertion of PORz pin
7. PRCM de-asserts reset to the host processor and all other peripherals without local CPUs.
8. Once host processors finish booting then all remaining peripherals see reset de-assertion.


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** All IPs with local CPUs have local reset asserted by default at RESETz and reset de-assertion requires host processor to write to corresponding registers in PRCM.

### 2.7.17 Power-Up/Down Sequence

The power-up/down sequence is as follows:

1. Each power domain has dedicated warm and cold reset.
2. Warm reset gets asserted each time there is any warm reset source requesting reset. Warm reset is also asserted when power domain moves from ON to OFF state.
3. Cold reset for the domain is asserted in response to cold reset sources. When domain moves from OFF to ON state then also cold reset gets asserted as this is similar to power-up condition for that domain.

### 2.7.18 IO State

All IOs except for JTAG and Reset output (and any special cases mentioned in pinlist) should have their output drivers tri-state, and internal pulls enabled during assertion of all reset sources. JTAG IO is affected only by TRSTz.

**NOTE:** The DSP, HDVICP, and Media Controller are held under reset after global warm reset by assertion of software source of reset. Other domains such are held under reset after global warm reset until the MPU software enables their respective interface clock.

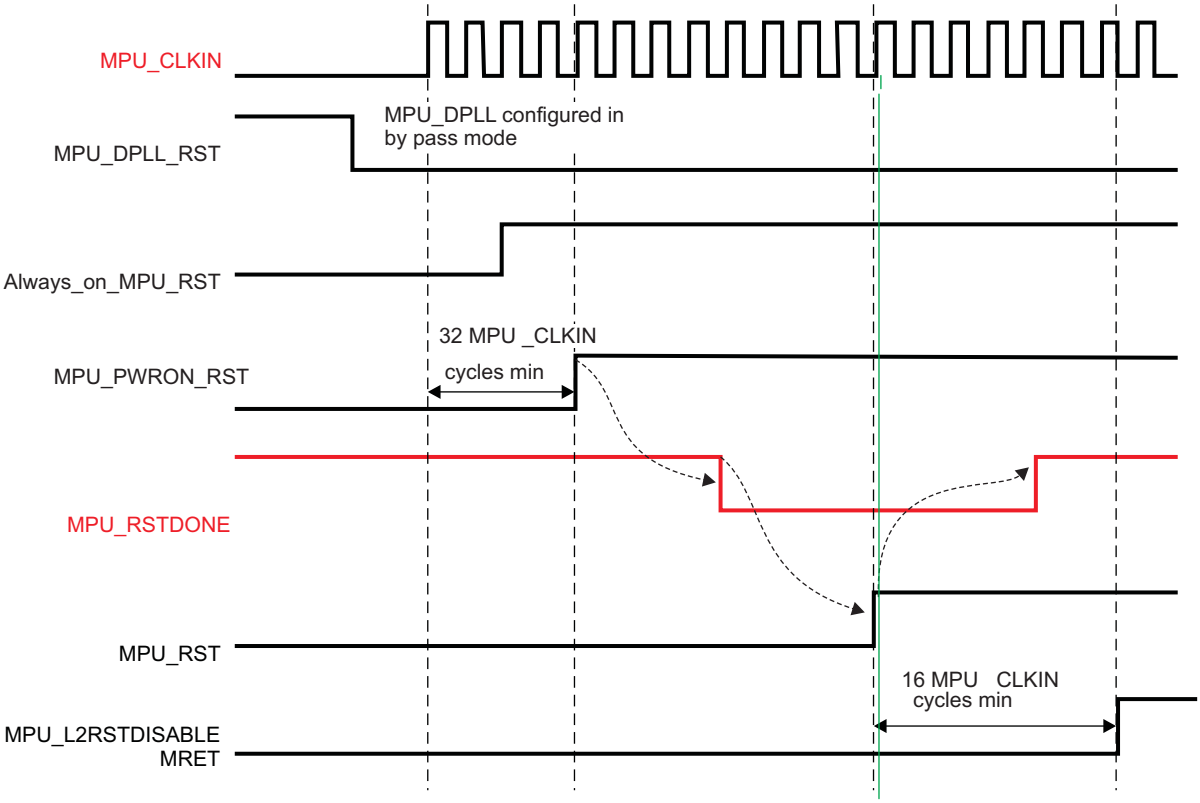
### 2.7.19 MPU Subsystem POR Sequence

The power on reset must be applied when MPU SS is first powered up. Following is the detailed description of MPU power on reset sequence.

1. During the initial power up sequence, the PRCM it asynchronously releases the DPLL_MAIN_RST signal for the main PLL. Once the main PLL reset is released, it starts the PLL initialization. The PLL is configured in bypass mode and it provides the bypass output CLK to all the modules inside MPU SS.
2. The PRCM releases the reset to INTC module inside MPUSS i.e. it asynchronously releases the MPU_AO_RST signal to MPUSS.
   Before the MPU power on reset is de-asserted, the PRCM must ensure that the MPU CLK is running so that MPU can be reset properly.
3. The PRCM releases MPU_PWRON_RST and it waits until MPUSS power-on Reset is complete. That is, the MPU_PWRON_RSTDONE signal is asserted HIGH by the MPUSS.
4. Following the de-assertion of MPU_PWRON_RST signal, the MPU SS de-asserts the nPORESET signal to the Cortex™-A8 CPU and after sufficient number of MPU CLK cycles MPU SS asserts the MPU_PWRON_RSTDONE signal high to indicate to the PRCM that synchronous power on reset sequence is completed inside MPU SS.
5. Once the PRCM sees that this MPU_PWRON_RSTDONE signal is asserted HIGH, PRCM de-asserts the MPU_RST signal and then the MPU starts booting.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-20. MPU SS POR Sequence**




### 2.7.20 MPU Subsystem Warm Sequence

During the warm reset of MPU, the PRCM asserts the MPU_RST signal only and power on reset signal is not asserted.

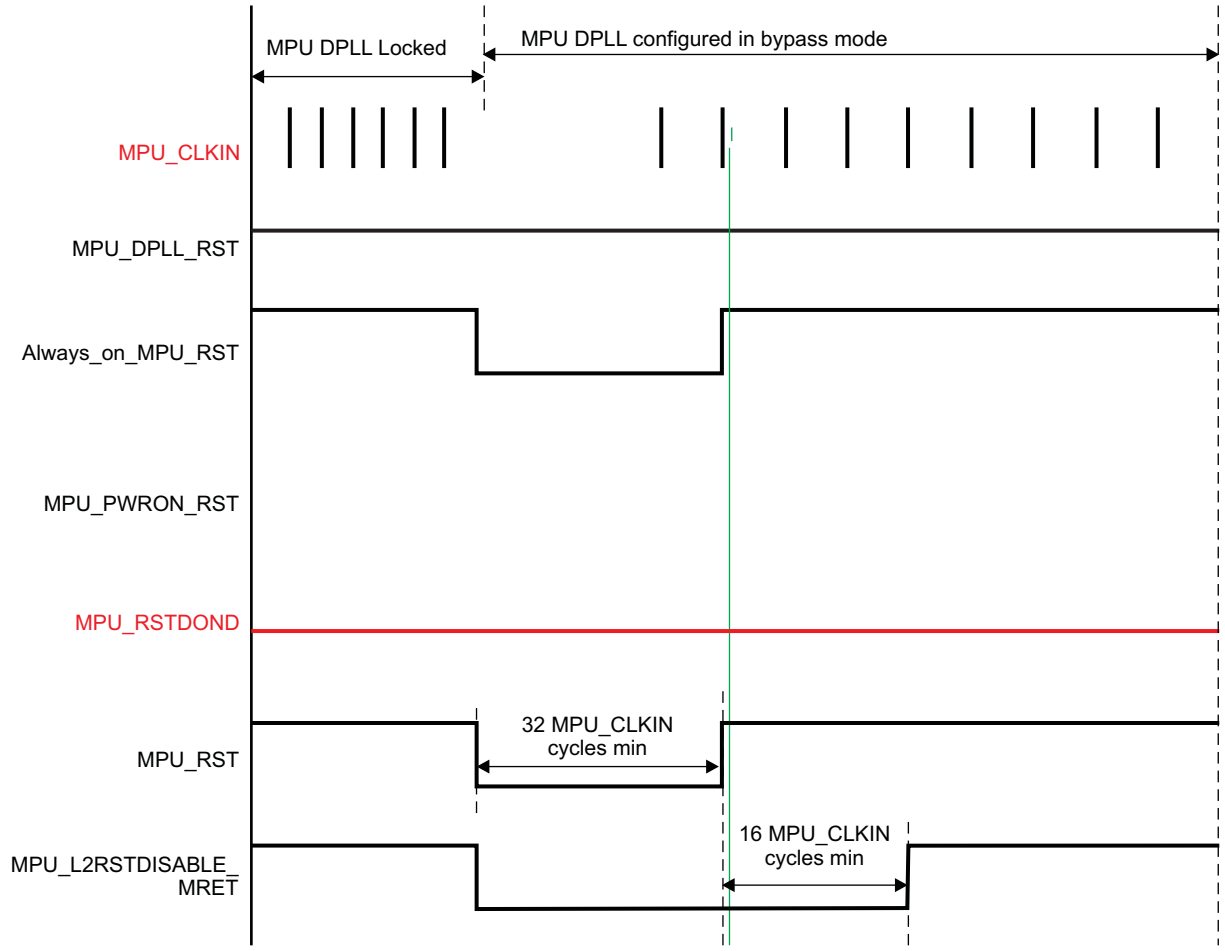
The MPU_RST is kept asserted for minimum 32 MPU clock cycles, that is, SYS_CLK in this case since during warm the reset of MPU.

Figure 2-21 illustrates the timing diagram for MPU warm reset sequence.



Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-21. MPU Warm Reset Sequence**

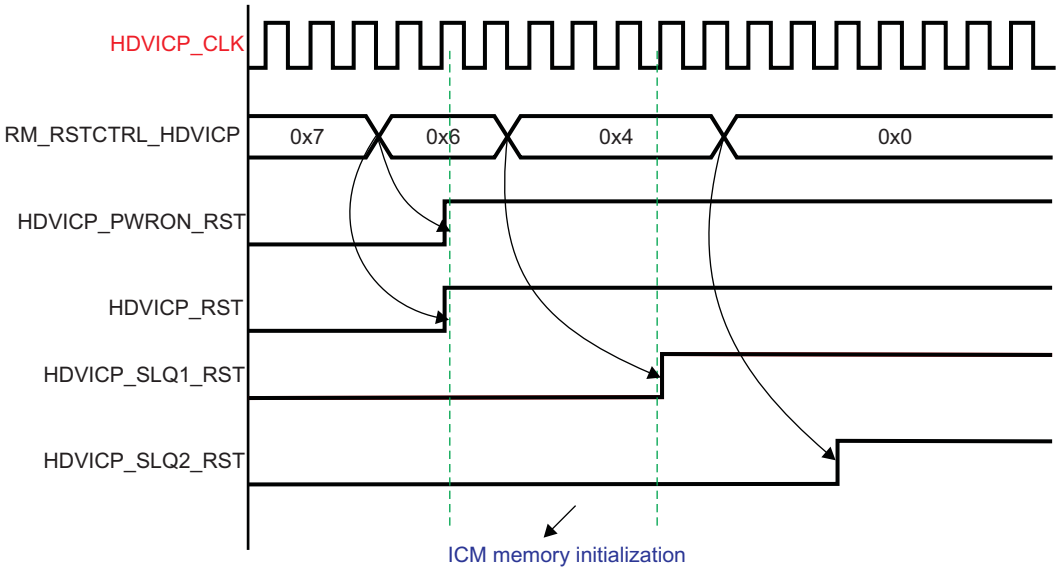



### 2.7.21 HDVICP Power-On Reset Sequence

Figure 2-22 illustrates the POR sequence of the HDVICP subsystem.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 2-22. POR Sequence of the HDVICP2**




POR to HDVICP is applied when PD_HDVICP is powered up.

Whenever the HDVICP is enabled by software control, the following sequence happens:

- The PRCM module provides the functional clock HDVICP_CLK to the HDVICP SUBSYSTEM
- The POR to HDVICP is de-asserted

After POR de-assertion, the sequence is:

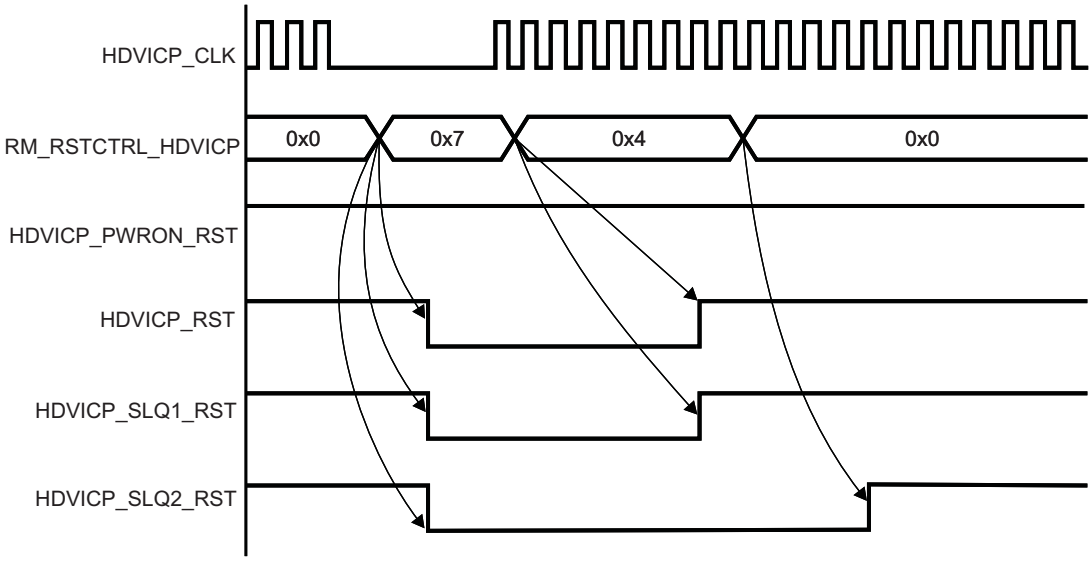
- Software clears the RM_HDVICP_RSTCTRL[2] RST3 bit. This causes the PRCM module to release the HDVICP_PWRN_RST reset, which is used inside HDVICP primarily to reset the emulation logic and the HDVICP_RST reset, which is used to reset all logic inside HDVICP. Then, software can download data into the TCM memory while keeping the sequencer CPUs under reset.
- Software clears the RM_HDVICP_RSTCTRL[0] RST1 bit. This releases the HDVICP_SEQ1_RST reset to the Sequencer1 CPU.
- Software can clear the RM_HDVICP_RSTCTRL[1] RST2 bit. This releases the HDVICP_SEQ2_RST reset to the Sequencer2 CPU.


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.7.22 HDVICP Software Warm Reset Sequence

Figure 2-23 illustrates the software warm reset sequence of the HDVICP.

**Figure 2-23. Software Warm Reset Sequence of the HDVICP**




Before asserting the software reset to the HDVICP, the MPU software must ensure that:

- The HDVICP sequencer CPUs are in IDLE state (CM_HDVICP_CLKCTRL[17:16] IDLEST).
- The HDVICP2 is in STANDBY state CM_HDVICP_CLKCTRL[[18] STBYST).
- The functional clock to the HDVICP has been gated by the PRCM module (CM_HDVICP_CLKSTCTRL[8] CLKACTIVITY_HDVICP_CLK).

The software reset sequence occurs when the MPU software:

1. Sets the RM_HDVICP_RSTCTRL[2] RST3, RM_HDVICP_RSTCTRL[1] RST2, and RM_HDVICP_RSTCTRL[0] RST1 bits. This causes the PRCM module to assert the HDVICP_RST, HDVICP_SEQ1_RST, and HDVICP_SEQ2_RST resets to the HDVICP2. HDVICP_PWRN_RST remains deasserted.
2. Enables the functional clock to the HDVICP.
3. clears the RM_HDVICP_RSTCTRL[2] RST3 and RM_HDVICP_RSTCTRL[0] RST1 bits. This causes the PRCM module to release the HDVICP_RST and HDVICP_SEQ1_RST resets to the HDVICP.
4. clears the RM_HDVICP_RSTCTRL[1] RST2 bit. This releases the HDVICP_SEQ2_RST reset to the Sequencer2 CPU.

### 2.7.23 C674x DSP Power-On Reset Sequence

The assumptions about POR de-assertion are:

- The MPU software sets the CM_ACTIVE_GEM_CLKCTRL [1:0] MODULEMODE bit field to enabled.
- The MPU software sets the CM_ACTIVE_GEM_CLKSTCTRL [1:0] CLKTRCTRL bit field to SW_WKUP.
- The PRCM is providing the clocks to C674x DSP.

The POR sequence is:

- The PRCM module releases GEM_POR when the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) reaches its limit.
- The MPU software clears the RM_ACTIVE_RSTCTRL[1] GEM_SW_RST bit in the PRCM module register to release the DSP, MMU, and CACHE interface from reset.


Copyright © 2011–2016, Texas Instruments Incorporated

- Once the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) expires, the PRCM module releases the GEM_GRST signal.
- On deassertion of the GEM_GRST signal, the C674x subsystem starts the initialization sequence. During the initialization sequence all the internal registers inside the C674x subsystem are properly reset and the reset for the DSP, MMU, and CACHE interface completes.
- The MPU software must configure the MMU once the MMU is out of reset. After the MMU is configured, the MPU software clears the RM_ACTIVE_RSTCTRL[0] GEM_LRST bit in the PRCM module register.

The PRCM module releases GEM_LRST, which causes the DSP to start booting.

### 2.7.24 C674x DSP Warm Reset Sequence

The assumptions about WARM reset de-assertion are:

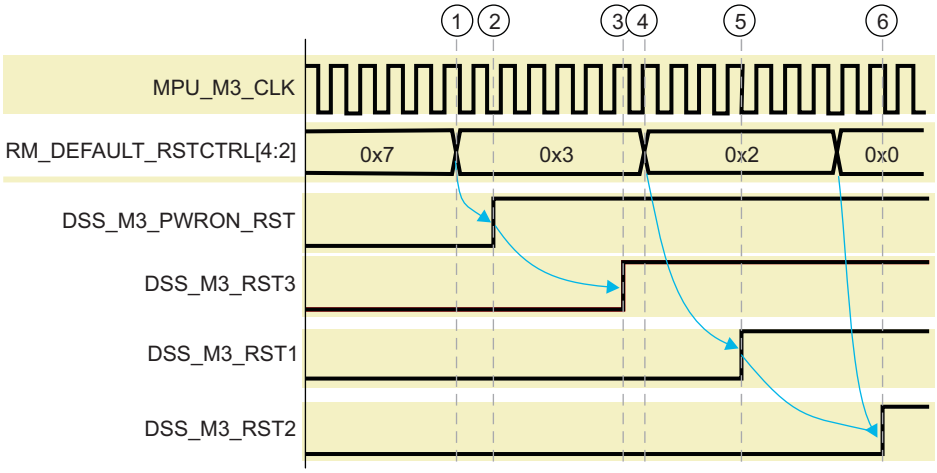
- The DSP is in IDLE state

The sequence is:

1. The MPU software sets the RM_ACTIVE_RSTCTRL[1] GEM_SW_RST and RM_ACTIVE_RSTCTRL[0] GEM_LRST bits.
2. The PRCM module asserts the GEM_GRST and GEM_LRST reset signals.GEM_POR remains deasserted in this case.
3. The MPU software re-enables the C674x DSP Clocks and clears the RM_ACTIVE_RSTCTRL[1] GEM_SW_RST bit in the PRCM module register to reset the DSP, MMU, and CACHE interface. The DSP subsystem starts the partial initialization sequence for the warm reset.
4. The MPU software clears the RM_ACTIVE_RSTCTRL[0] GEM_LRST bit in the PRCM module register.
5. The PRCM module releases GEM_LRST, which causes the DSP to start booting.

### 2.7.25 Media Controller Power-On Reset Sequence

**Figure 2-24. Media Controller Power-On Reset Sequence**




The assumptions about POR de-assertion are:

- The Media Controller MPU subsystem is held in reset by the PRCM module and the following are asserted:
  - DSS_PWRN_RST
  - DSS_RST1
  - DSS_RST2
  - DSS_RST3


Copyright © 2011–2016, Texas Instruments Incorporated

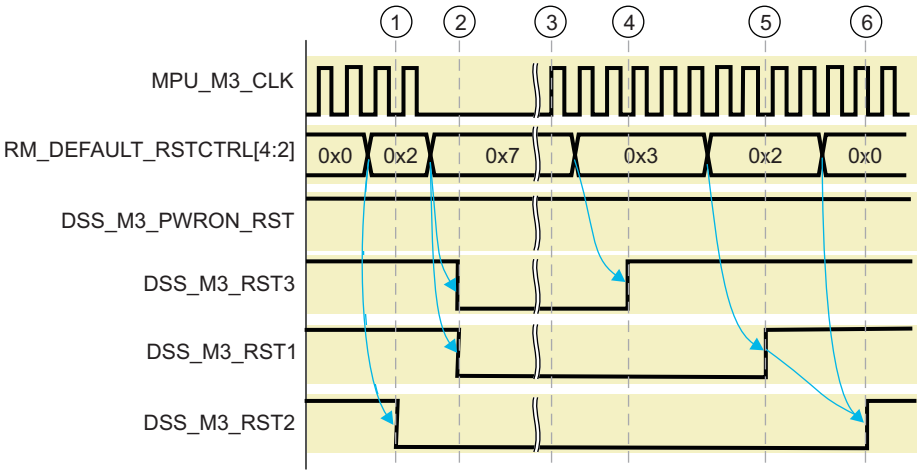
The POR sequence is:

1. Software enables Media Controller MPU subsystem.
2. Software clears the RM_DEFAULT_RSTCTRL[4] RST3 bit in the PRCM module register to release the Media Controller MPU Cache and MMU from reset.
3. Once the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) expires, the PRCM module releases the DSS_RST3 signal.
4. Upon the deassertion of the DSS_M3_RST3 signal, the Media Controller MPU subsystem starts the CPU and MMU initialization sequence
5. The MPU software must configure the MMU once the MMU is out of reset. After MMU configuration and cache initialization is done, the MPU software clears the RM_DEFAULT_RSTCTRL[3] RST1 bit in the PRCM module register.
6. The PRCM module releases DSS_RST1, which causes the Media Controller MPU Core 1 to start booting.
7. The MPU software can clear the RM_DEFAULT_RSTCTRL[2] RST2 bit in the PRCM module register so that the PRCM module releases DSS_RST2 to the Media Controller MPU Core 2.


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.7.26 Media Controller Warm Reset Sequence

**Figure 2-25. Media Controller MPU Subsystem Software Warm Reset Sequence**




The assumption about Warm reset assertion is:

- The Media Controller MPU Core2 is in IDLE state.

The sequence is:

1. Media Controller MPU Core 1 software or Cortex™-A8 MPU software sets the RM_DEFAULT_RSTCTRL[3] RST2 bit. The PRCM module asserts the DSS_RST2 reset signal to the Media Controller Core 2.
2. The Cortex™-A8 MPU cuts the MPU_M3_CLK and software sets the RM_DEFAULT_RSTCTRL[4] RST3 and RM_DEFAULT_RSTCTRL[2] RST1 bits.
3. The PRCM module asserts the DSS_RST3, DSS_RET_RST, and the DSS_RST1 reset signals. The DSS_PWRN_RST remains deasserted in this case.
4. The MPU software re-enables the MPU_M3_CLK and the initialization sequence starts inside the Media Controller subsystem. Software clears the RM_DEFAULT_RSTCTRL[4] RST3 bit in the PRCM module register.
5. When the reset sequence in Step 4 completes and the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) expires, the PRCM module releases the DSS_RST3, and DSS_RST1 reset signals. Media Controller Core 1 starts rebooting.
6. Software clears the RM_DEFAULT_RSTCTRL[3] RST2 bit in the PRCM module register. The PRCM module releases the DSS_RST2 reset signal to the Media Controller Core 2. Core 2 starts to boot.

## 2.8 Voltage and Power Domains

Table 2-37 shows how the device core logic is partitioned into 4 core logic voltage domains, 4 memory voltage domains and 6 power domains. The table lists which voltage and power domain a functional module belongs. All voltage domains must be supplied during any mode of operation while 5 of the 6 power domains can turned off by internal switch.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-37. Volage and Power Domains**

| Logic Voltage Domain Name | Module |
|---|---|
| ARM | Cortex™-A8, |
| CORE | DCAN0, DCAN1, DDR_PHY_LOGIC0, DDR_PHY_LOGIC1, DMM EDMA, ELM, EMIF0, EMIF1, 2 Channel GMAC Switch, GPIO_CNTL0, GPIO_CNTL1, GPIO_CNTL2, GPIO_CNTL3,GPMC, I2C0, I2C1, I2C2, I2C3, Interconnect, IPC, MCASP0, MCASP1, MCASP2, MCASP3, MCASP4, MCASP5, ATL, MCBSP, MLB, OCMC SRAM, PATA, PBIST, PCI, PRCM, RTC, SATA, SCR, SD/MMC0, SD/MMC1, SD/MMC2, SPI0, SPI1, SPI2, SPI3, Switch fabric, Timer1, Timer2, Timer3, Timer4, Timer5, Timer6, Timer7, Timer8, UART0, UART1, UART2, UART3, UART4, UART5, USB0, USB1, WDT0, VCP, XBAR, Debug_SS |
| | FACEDETECT |
| | SGX530 Subsystem |
| | HDD_SS, HDMI, SD-DAC LOGIC |
| ACTIVE (DSP) | C674x+™ Subsystem. , C674x+™ L2 SRAM, |
| HDVICP | HDVICP, |

### 2.8.1 Voltage Domains

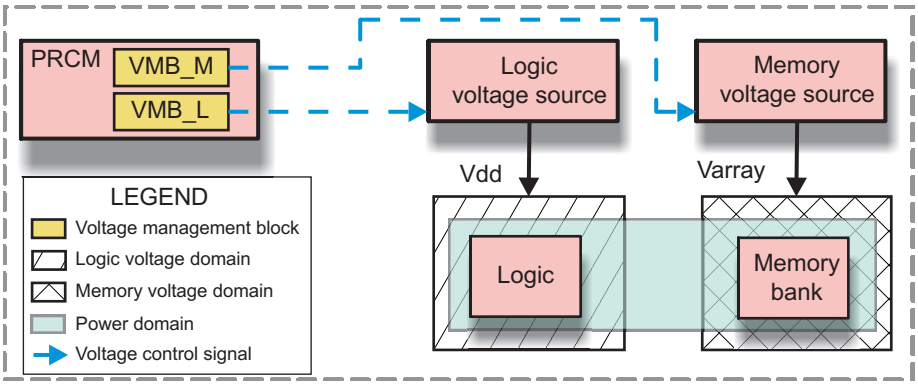
The PRCM module controls the voltage scaling (that is, switching the voltage in discrete steps or in a continuum within a range of possible values) of the power sources of the device. This allows control of the device power consumption according to the performance criteria defined. Higher performance is ensured with higher voltage and clock frequencies (and hence higher power consumption), while lower performance can be supported with lowered power consumption by reducing or completely gating the power supply to specific areas of the device and gating the associated clocks.
A voltage domain is a section of the device supplied by a dedicated voltage source (that is, an internal LDO or external switched mode power supply [SMPS]). A voltage domain may or may not be controlled by the PRCM module. When a voltage domain is controlled by the PRCM module, a dedicated voltage manager is associated. The voltage manager allows regulation of the voltage level of the source for the voltage domain, independently of other voltage domains of the device.
The core logic is divided into four voltage domains. This division permits the system to scale down voltage to some functional blocks while maintaining highest performance on other blocks.

The voltage managers in the PRCM module can be of two types:

- Hardware controlled to automatically switch the domain voltage levels according to the associated hardware conditions.

Figure 2-26 shows a voltage domain.

**Figure 2-26. Generic Voltage Domain**





Copyright © 2011–2016, Texas Instruments Incorporated

### 2.8.2 Power Domains

In order to reduce power due to leakage, the core logic supply voltage to six of the device power domains can be turned OFF with internal power switches. The internal power switches are controlled through memory mapped registers in the PRCM. In an use case whereby all the modules within a power domain are not used that power domain can be place in the OFF state.

Table 2-38 shows the allowable combination power domain ON/OFF states and which power domains are switched via internal power switches. At power-on-reset, all domains except always-on are in the power domain OFF state.

**Table 2-38. Power Domain State Table**

| | POWER DOMAIN | | | | | |
|---|---|---|---|---|---|---|
| MODE | ALWAYS_ON | ISP (ISS) | GFX | DSS | ACTIVE (DSP) | HDVICP |
| No Voltage Supply | N/A | N/A | N/A | N/A | N/A | N/A |
| Power On Reset | ON | OFF | OFF | OFF | OFF | OFF |
| ALL OTHER FUNCTIONAL MODES | ON | DON'T CARE | DON'T CARE | DON'T CARE | DON'T CARE | DON'T CARE |
| Internal Power Switch | NO | YES | YES | YES | YES | YES |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.8.3 Device Modules and Power Management Attribute List

The device modules and power management list are described below:

**Table 2-39. Device Modules**

| Power Domain Name | Module |
|---|---|
| ALWAYS_ON | Cortex™-A8, , DCAN0, DCAN1, DDR_PHY_LOGIC0, DDR_PHY_LOGIC1, DMM EDMA, ELM, EMIF0, EMIF1, 2 Channel GMAC Switch, GPIO_CNTL0, GPIO_CNTL1, GPIO_CNTL2, GPIO_CNTL3,GPMC, I2C0, I2C1, I2C2, I2C3, Interconnect, IPC, MCASP0, MCASP1, MCASP2, MCASP3, MCASP4, MCASP5, ATL,MCBSP, MLB, OCMC SRAM, PATA, PBIST, PCI, PRCM, RTC, SATA, SCR, SD/MMC0, SD/MMC1, SD/MMC2, SPI0, SPI1, SPI2, SPI3, Switch fabric, Timer1, Timer2, Timer3, Timer4, Timer5, Timer6, Timer7, Timer8, UART0, UART1, UART2, UART3, UART4, UART5, USB0, USB1, WDT0, VCP, XBAR, Debug_SS |
| ISS (ISP) | FACEDETECT, ISS |
| SGX530 | SGX530 Subsystem |
| DSS | HDD_SS, HDMI, SD-DAC LOGC |
| ACTIVE (DSP) | C674x+™ Subsystem. , C674x+™ L2 SRAM, |
| HDVICP | HDVICP, SRSense2 |

### 2.8.4 Power Domain Power-Down Sequence

Below is the sequence of steps which happen during power down of a power domain.

All IPs (belonging to a power domain) with STANDBY interface asserts STANDBY. The STANDBY assertion should get triggered by the IP based on its activity on OCP initiator port. The IP should assert STANDBY whenever initiator port is IDLE. Some of the DSPS IPs may not have this feature and they require SW write to standby-mode register to get STANDBY assertion from IP.

1. SW requests all modules in given power domain to go to disable state by programming module control register inside PRCM.
2. PRCM starts and waits for completion of power management handshake with IPs (IdleReq/IdleAck).
3. PRCM gates-off all the clocks to the power domain.
4. SW requests all clock domains in given power domain to go to “force sleep” mode by programming functional clock domain register in PRCM. PRCM has already gated-off clocks and this register programming may look redundant.
5. SW requests PRCM to take this power domain to OFF state by programming PWRSTCTRL register. This step can be skipped if PWRSTCTRL is permanently programmed to OFF state. When this is done, functional clock domain register decides when power domain is taken to OFF state. Only reason not to have OFF state in PWRSTCTRL is to take power domain to just clock gate state without power gating.
6. PSCON specific to this power domain asserts isolation enable for the domain.
7. PRCM asserts warm and cold reset to the power domain.
8. PSCON asserts control signals to switch-off power using on-die switches.
9. On-die switches sends acknowledge back to PSCON.

### 2.8.5 Power Domain Power-Up Sequence

Below is the sequence of steps during power-up of a power domain. This sequence is not relevant to ALWAYS_ON power domain as this domain never goes to OFF state as long as the device is powered. This sequence is repeated each time a domain is taken to ON state from OFF (including first time power-up).

There can be multiple reasons to start power-up sequence for a domain. For example it can be due to an interrupt from one of the IPs which is powered-up.


Copyright © 2011–2016, Texas Instruments Incorporated

1. SW requests required clock domains inside this power domain to go to force wake-up state by programming the functional clock domain register.
2. PRCM enables clocks to the required clock domains.
3. PSCON specific to this power domain asserts control signal to ungate the power.
4. Once power is ungated, on-die switches sends acknowledge back to PSCON.
5. PRCM deasserts cold and warm reset to the power domain.
6. PRCM turns off isolation cells.
7. SW requests PRCM to enable the required module in the power domain by programming the module control register.
8. PRCM initiates and waits for completion of PM protocol to enable the modules (IdleReq/IdleAck).


Copyright © 2011–2016, Texas Instruments Incorporated

## 2.9 Voltage Management Functional Description

This section describes the voltage domains and voltage control architecture. It also explains the interactions between the device and the external power IC.

### 2.9.1 Overview

The voltage-management architecture of the device is based on voltage sources managed by the PRCM module. They define the voltage domains within the device (see Section 2.8.1, *Voltage Domains*). This partition of the voltage domains ensures independent voltage control of each voltage domain through dedicated SMPS or LDO. The following voltage domains are managed by the PRCM module:

- VDD_ARM_L
- VDD_CORE_L
- VDD_DSP_L
- VDD_HDVICP_L
- LDO_ARM_M
- LDO_CORE_M
- LDO_DSP_M
- LDO_HDVICP_M
- LDO_ARM_ABB
- LDO_DSP_ABB
- LDO_HDVICP_ABB

**NOTE:** See the device *Datasheet* document for device power supply pin association to the power domain.

Table 2-40 lists the supported DVFS OPPs for each voltage domain. The exact values depend on the silicon; for more information, see the device *Datasheet*.

**Table 2-40. Supported DVFS OPPs Summary**

| Voltage Domain | Operating Point | Control |
|---|---|---|
| VDD_ARM_L (cvdd_arm) | OPP100 | Hardware and software |
| | OPP120 | |
| | OPP166 | |
| VDD_HDVICP_L (cvdd_hdvicp) | OPP100 | Hardware and software |
| | OPP120 | |
| | OPP166 | |
| VDD_CORE_L (cvdd) | OPP100 | Hardware and software |
| | OPP120 | |
| | OPP166 | |
| VDD_DSP_L (cvdd_dsp) | OPP100 | Hardware and software |
| | OPP120 | |
| | OPP166 | |

**NOTE:** For the supported OPPs for each device in the *Davinci* family, see in the device *Datasheet*.

#### DVFS OPPs Dependencies

For detailed information about DVFS OPP dependencies, see the device *Datasheet*.


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.9.2 Internal LDOs Control

#### 2.9.2.1 Memory LDOs

Embedded SRAM LDOs (that is, LDO_CORE_M, LDO_HDVICP_M, LDO_ARM_M and LDO_DSP_M) are used to supply power to the split-rail memory arrays. The PRM generates the controls used to select LDO operating mode: on-active, or off.

Split-rail type SRAMs are used in the device to implement the larger memories. These SRAMs feature memory array and periphery logic which are on separate supplies to allow independent power management. Proper memory operation, however, requires the SRAM array voltage to never be operated at a level lower than the SRAM periphery logic.

Memory LDO can switch to on, and off mode:

- On mode: 1.2 V is the normal voltage reference used through all functional OPPs, whenever memories must be functional. When logic voltage level VDD_x_L (where x can be ARM, CORE, HDVICP and DSP) becomes higher than the associated memory voltage level VDD_x_M, the LDO operates in the tracking mode and follows its respective the VDD_x_L voltage level.
- Off mode: 0 V is set when the device switches to off mode.

**NOTE:** The voltage levels associated with the different modes may depend on the device characteristics.

#### 2.9.2.2 ABB LDOs Control

The ABB LDO supports three voltage modes:

- Bypass mode: In this mode the <x>_ABB LDO is bypassed and outputs the VDD_<x>_L voltages (x refers to ARM, DSP, HDVICP). This mode is activated when ABB is not required, or when voltage domain enters low-power mode.
- ABB Set1: This mode is enabled at OPP_166.
- ABB Set2: This mode is enabled at OPP_120.

The Control module provides the SGX_VBBLDO_CTRL, HDVICP_VBBLDO_CTRL, C674x_DSP_VBBLDO_CTRL, ARM_CORTEX_A8_VBBLDO_CTRL registers for configuration with the following controls:

- LDOBYPASSZ – Bypass the ABB power management
- BBSEL – high selects FBB

The register RAMLDO_CTRL in Control module controls RAM LDOs. It provides following controls:

- AIPOFF - Disable LDO and enables IDDQ mode.
- SRAMALLRET - puts VDDAR in retention mode.
- ABBOFF - shorts VDDAR to VNWA.

## 2.10 PRCM Registers

The following are memory-mapped registers for the PRCM.

**Table 2-41. PRCM Memory Mapped Registers**

| Module Name | Base Address Offset | Size |
|---|---|---|
| PRM_DEVICE | 0x0000 | 256 Bytes |
| CM_DEVICE | 0x0100 | 256 Bytes |
| OCP_SOCKET_PRM | 0x0200 | 256 Bytes |
| CM_DPLL | 0x0300 | 256 Bytes |
| CM_ACTIVE (DSP) | 0x0400 | 256 Bytes |
| CM_DEFAULT | 0x0500 | 256 Bytes |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-41. PRCM Memory Mapped Registers (continued)**

| Module Name | Base Address Offset | Size |
|---|---|---|
| CM_HDVICP | 0x0600 | 256 Bytes |
| CM_ISS | 0x0700 | 256 Bytes |
| CM_DSS | 0x0800 | 256 Bytes |
| CM_SGX530 | 0x0900 | 256 Bytes |
| PRM_ACTIVE (DSP) | 0x0A00 | 256 Bytes |
| PRM_DEFAULT | 0x0B00 | 256 Bytes |
| PRM_HDVICP | 0x0C00 | 256 Bytes |
| PRM_ISP (ISS) | 0x0D00 | 256 Bytes |
| PRM_DSS | 0x0E00 | 256 Bytes |
| PRM_SGX530 | 0x0F00 | 256 Bytes |
| CM_ALWON | 0x1400 | 1 KBytes |
| PRM_ALWON | 0x1800 | 1 KBytes |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1 PLLSS Registers

Table 2-42 lists the memory-mapped registers for the PLLSS. All register offset addresses not listed in Table 2-42 should be considered as reserved locations and the register contents should not be modified.

**Table 2-42. PLLSS REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | CONTROL_REVISION | The CONTROL_REVISION register is a read only register containing the revision number of the Control module.<br>A write to this register has no effect. | Section 3.2.1 |
| 4h | CONTROL_HWINFO | The CONTROL_HWINFO register contains information about the module hardware configuration (typically HDL generics, if any). | Section 3.2.2 |
| 10h | CONTROL_SYSCONFIG | The CONTROL_SYSCONFIG register implements only a reduced set of the register defined in the OCP Design Guidelines. | Section 3.2.3 |
| 40h | PLLSS_MMR_LOCK | The PLLSS_MMR_LOCK Register is used allow/block writes to the registers in address region Start_Addess_x -End_Address_x.<br>This register has 2 states LOCKED and UNLOCKED<br>When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked.<br>When in UNLOCKED state all writes to Start_Addess_x -End_Address_x is allowed.<br>All reads to Start_Addess_x -End_Address_x should not be blocked by the state of PLLSS_MMR_LOCK registers but they should follow the underlying security specified in each section, e.g.<br>Firewall setting, Non Secure Supervisior etc. | Section 2.10.1.4 |
| 48h | MPUPLL_PWRCTRL | The MPU PLL Power Control register is used to control the power state of the MPU _PLL. | Section 2.10.1.5 |
| 4Ch | MPUPLL_CLKCTRL | The MPU PLL Clock Control register is used to control the different clock control state of the MPU_PLL | Section 2.10.1.6 |
| 50h | MPUPLL_TENABLE | The MPU PLL TENABLE register is used to load the M, N dividers of MPU_PLL | Section 2.10.1.7 |
| 54h | MPUPLL_TENABLEDIV | The MPU PLL TENABLEDIV register is used to load the M2,N2 dividers of MPU_PLL | Section 2.10.1.8 |
| 58h | MPUPLL_M2NDIV | MPU PLL M2NDIV register is for programming M2 and N values of MPU _PLL | Section 2.10.1.9 |
| 5Ch | MPUPLL_MN2DIV | MPU PLL MN2DIV register is for programming M and N2 values of MPU _PLL | Section 2.10.1.10 |
| 60h | MPUPLL_FRACDIV | The MPU PLL FRACDIV register is for controlling the fractional values of MPU _PLL | Section 2.10.1.11 |
| 64h | MPUPLL_BWCTRL | The MPU PLL_BWCTRL register is for controlling the loop bandwidth of MPU _PLL | Section 2.10.1.12 |
| 68h | MPUPLL_FRACCTRL | Controlls the fractional portion of Modena PLL | Section 2.10.1.13 |
| 6Ch | MPUPLL_STATUS | The MPU PLL Status register is for viewing the status of the MPU PLL. | Section 2.10.1.14 |
| 70h | MPUPLL_M3DIV | The MPU PLL_M3DIV register is for controlling the M3 divider of MPU _PLL | Section 2.10.1.15 |
| 74h | MPUPLL_RAMPCTRL | The MPUPLL_RAMPCTRL register is for controlling the ramping of MPU _PLL | Section 2.10.1.16 |
| 80h | DSPPLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.17 |
| 84h | DSPPLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.18 |
| 88h | DSPPLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.19 |
| 8Ch | DSPPLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.20 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-42. PLLSS REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 90h | DSPPLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.21 |
| 94h | DSPPLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.22 |
| 98h | DSPPLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.23 |
| 9Ch | DSPPLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.24 |
| A0h | DSPPLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.25 |
| A4h | DSPPLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.26 |
| B0h | SGXPLLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.27 |
| B4h | SGXPLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.28 |
| B8h | SGXPLLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.29 |
| BCh | SGXPLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.30 |
| C0h | SGXPLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.31 |
| C4h | SGXPLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.32 |
| C8h | SGXPLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.33 |
| CCh | SGXPLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.34 |
| D0h | SGXPLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.35 |
| D4h | SGXPLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.36 |
| E0h | HDVICPPLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.37 |
| E4h | HDVICPPLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.38 |
| E8h | HDVICPPLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.39 |
| ECh | HDVICPPLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.40 |
| F0h | HDVICPPLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.41 |
| F4h | HDVICPPLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.42 |
| F8h | HDVICPPLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.43 |
| FCh | HDVICPPLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.44 |
| 100h | HDVICPPLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.45 |
| 104h | HDVICPPLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.46 |
| 110h | L3PLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.47 |
| 114h | L3PLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.48 |
| 118h | L3PLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.49 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-42. PLLSS REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 11Ch | L3PLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.50 |
| 120h | L3PLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.51 |
| 124h | L3PLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.52 |
| 128h | L3PLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.53 |
| 12Ch | L3PLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.54 |
| 130h | L3PLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.55 |
| 134h | L3PLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.56 |
| 140h | ISPPLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.57 |
| 144h | ISPPLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.58 |
| 148h | ISPPLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.59 |
| 14Ch | ISPPLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.60 |
| 150h | ISPPLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.61 |
| 154h | ISPPLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.62 |
| 158h | ISPPLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.63 |
| 15Ch | ISPPLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.64 |
| 160h | ISPPLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.65 |
| 164h | ISPPLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.66 |
| 170h | DSSPLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.67 |
| 174h | DSSPLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.68 |
| 178h | DSSPLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.69 |
| 17Ch | DSSPLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.70 |
| 180h | DSSPLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.71 |
| 184h | DSSPLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.72 |
| 188h | DSSPLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.73 |
| 18Ch | DSSPLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.74 |
| 190h | DSSPLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.75 |
| 194h | DSSPLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.76 |
| 1A0h | VIDEO0PLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.77 |
| 1A4h | VIDEO0PLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.78 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-42. PLLSS REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 1A8h | VIDEO0PLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.79 |
| 1ACh | VIDEO0PLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.80 |
| 1B0h | VIDEO0PLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.81 |
| 1B4h | VIDEO0PLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.82 |
| 1B8h | VIDEO0PLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.83 |
| 1BCh | VIDEO0PLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.84 |
| 1C0h | VIDEO0PLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.85 |
| 1C4h | VIDEO0PLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.86 |
| 1D0h | VIDEO1PLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.87 |
| 1D4h | VIDEO1PLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.88 |
| 1D8h | VIDEO1PLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.89 |
| 1DCh | VIDEO1PLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.90 |
| 1E0h | VIDEO1PLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.91 |
| 1E4h | VIDEO1PLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.92 |
| 1E8h | VIDEO1PLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.93 |
| 1ECh | VIDEO1PLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.94 |
| 1F0h | VIDEO1PLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.95 |
| 1F4h | VIDEO1PLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.96 |
| 200h | HDMIPLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.97 |
| 204h | HDMIPLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.98 |
| 208h | HDMIPLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.99 |
| 20Ch | HDMIPLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.100 |
| 210h | HDMIPLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.101 |
| 214h | HDMIPLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.102 |
| 218h | HDMIPLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.103 |
| 21Ch | HDMIPLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.104 |
| 220h | HDMIPLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.105 |
| 224h | HDMIPLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.106 |
| 230h | AUDIOPLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.107 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-42. PLLSS REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 234h | AUDIOPLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.108 |
| 238h | AUDIOPLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.109 |
| 23Ch | AUDIOPLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.110 |
| 240h | AUDIOPLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.111 |
| 244h | AUDIOPLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.112 |
| 248h | AUDIOPLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.113 |
| 24Ch | AUDIOPLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.114 |
| 250h | AUDIOPLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.115 |
| 254h | AUDIOPLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.116 |
| 260h | USBPLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.117 |
| 264h | USBPLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.118 |
| 268h | USBPLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.119 |
| 26Ch | USBPLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.120 |
| 270h | USBPLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.121 |
| 274h | USBPLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.122 |
| 278h | USBPLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.123 |
| 27Ch | USBPLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.124 |
| 280h | USBPLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.125 |
| 284h | USBPLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.126 |
| 290h | DDRPLL_PWRCTRL | The PLL Power Control register is used to control the power state of therespective PLL. | Section 2.10.1.127 |
| 294h | DDRPLL_CLKCTRL | The PLL Clock Control register is used to control the different clock control state of the respective PLL | Section 2.10.1.128 |
| 298h | DDRPLL_TENABLE | Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ. | Section 2.10.1.129 |
| 29Ch | DDRPLL_TENABLEDIV | The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL | Section 2.10.1.130 |
| 2A0h | DDRPLL_M2NDIV | PLL M2NDIV register is for programming M2 and N values of respective PLL | Section 2.10.1.131 |
| 2A4h | DDRPLL_MN2DIV | PLL MN2DIV register is for programming M and N2 values of respective PLL | Section 2.10.1.132 |
| 2A8h | DDRPLL_FRACDIV | The PLL FRACDIV register is for controlling the fractional values of respective PLL | Section 2.10.1.133 |
| 2ACh | DDRPLL_BWCTRL | The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL | Section 2.10.1.134 |
| 2B0h | DDRPLL_FRACCTRL | Controlls the fractional portion of respective ADPLLLJ. | Section 2.10.1.135 |
| 2B4h | DDRPLL_STATUS | The PLL Status register is for viewing the status of the respective PLL. | Section 2.10.1.136 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-42. PLLSS REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 2C0h | OSC_SRC | The following groups of PLL needs a selection of clock source of OSC0 or OSC1 o DSPPLL/SGXPLL/HDVICPPLL/ISPPLL/DSSPLL o VIDEO0PLL o VIDEO1PLL o HDMIPLL o AUDIOPLL o For TEST reasons, the following additional PLLs also have OSC0 vs OSC1 selection: GEM, SGX, HDVICP, ISP, DSS, USB, DDR | Section 2.10.1.137 |
| 2C4h | MPU_CLKSRC | Clock source for ARM PLL | Section 2.10.1.138 |
| 2C8h | VIDEO_PLL_CLKSRC | Select the clock source for VIDEO PLLs, VENC_G_CLK and audio PLL CLK2 input of PRCM | Section 2.10.1.139 |
| 2CCh | MLB_ATL_CLKSRC | MLB and ATL clock source | Section 2.10.1.140 |
| 2D0h | McASP345_AUX_CLKSRC | Select the AUX clock source for McASP3/4/5 | Section 2.10.1.141 |
| 2D4h | McASP_AHCLK_CLKSRC | Select the source of McASP AHCLKX/R | Section 2.10.1.142 |
| 2D8h | McBSP_UART_CLKSRC | Select the clock source for UART3/4/5 and MCBSP_CLKS. | Section 2.10.1.143 |
| 2DCh | HDMI_I2S_CLKSRC | Select the clock source for HDMI_I2S. | Section 2.10.1.144 |
| 2E0h | DMTIMER_CLKSRC | Select the clock source for DMTIMERs 1..8 | Section 2.10.1.145 |
| 2E4h | CLKOUT_MUX | Control the options for CLKOUT multiplexing. The clock outputs are respectively CLKOUT0 and CLKOUT1. | Section 2.10.1.146 |
| 2E8h | RMII_REFCLK_SRC | Control the Cpts_RFT_CLK source mux and RMII_Ref_CLK source mux. | Section 2.10.1.147 |
| 2F0h | SYSCLK18_CLKSRC | Select the clock source of SYSCLK18. | Section 2.10.1.148 |
| 2F4h | WDT0_CLKSRC | Select the clock source of WDT0. | Section 2.10.1.149 |
| 320h | DMTIMER_CLK_CHANGE | This register controls the IDLE REQUEST and logs the IDLE STATUS for DMTIMERs1..8. | Section 2.10.1.150 |
| 324h | DEEPSLEEP_CTRL | The DEEPSLEEP control register is not supported on this SoC. | Section 2.10.1.151 |
| 328h | DEEPSLEEP_STATUS | The DEEPSLEEP status register logs the internal deepsleep counter value. | Section 2.10.1.152 |

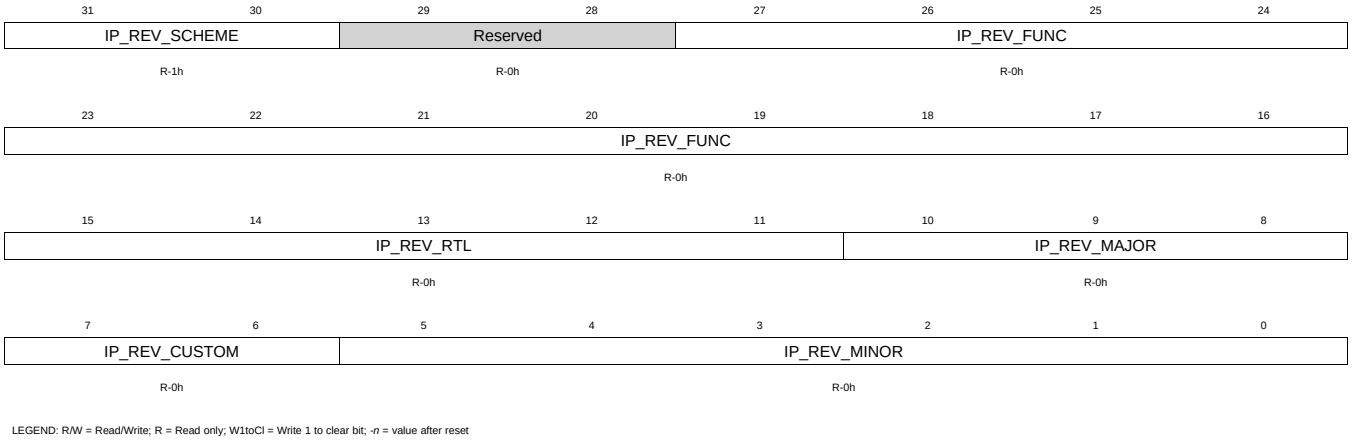

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.1 CONTROL_REVISION Register (offset = 0h) [reset = 40000000h]

CONTROL_REVISION is shown in Figure 3-2 and described in Table 3-4.

The CONTROL_REVISION register is a read only register containing the revision number of the Control module. A write to this register has no effect.

**Figure 2-27. CONTROL_REVISION Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-43. CONTROL_REVISION Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | IP_REV_SCHEME | R | 1h | Used to distinguish register scheme<br>0x0: Legacy (ASP or WTBU) scheme<br>0x1: Highlander 0.8 scheme |
| 29-28 | Reserved | R | 0h | |
| 27-16 | IP_REV_FUNC | R | 0h | Function indicates a software compatible module family. If there is no level of software compatibility a new Func number (and hence REVISION) should be assigned. |
| 15-11 | IP_REV_RTL | R | 0h | RTL Version (R), maintained by IP design owner. |
| 10-8 | IP_REV_MAJOR | R | 0h | Major Revision (X), maintained by IP specification owner. |
| 7-6 | IP_REV_CUSTOM | R | 0h | Indicates a special version for a particular device. Consequence of use may avoid use of standard Chip Support Library (CSL) / Drivers |
| 5-0 | IP_REV_MINOR | R | 0h | Minor Revision (Y), maintained by IP specification owner. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.2 CONTROL_HWINFO Register (offset = 4h) [reset = 0h]

CONTROL_HWINFO is shown in Figure 3-3 and described in Table 3-5.

The CONTROL_HWINFO register contains information about the module hardware configuration (typically HDL generics, if any).

**Figure 2-28. CONTROL_HWINFO Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IP_HWINFO | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-44. CONTROL_HWINFO Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | IP_HWINFO | R | 0h | IP Module dependent |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.3 CONTROL_SYSCONFIG Register (offset = 10h) [reset = 2Ah]

CONTROL_SYSCONFIG is shown in Figure 3-4 and described in Table 3-6.

The CONTROL_SYSCONFIG register implements only a reduced set of the register defined in the OCP Design Guidelines.

**Figure 2-29. CONTROL_SYSCONFIG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | STANDBY | | IDLEMODE | | FREEEMU | Reserved |
| R-0h | | R-2h | | R/W-2h | | R-1h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-45. CONTROL_SYSCONFIG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-4 | STANDBY | R | 2h | Configure local initiator state management.Reserved in Control Module since it has no local initiator.<br>0x0 = 00: Force Standby<br>0x1 = 01: No Standby Mode<br>0x2 = 10: Smart Standby<br>0x3 = 11: Smart Standby wakeup capable |
| 3-2 | IDLEMODE | R/W | 2h | Configure local target state management<br>0x0 = 00: Force Idle<br>0x1 = 01: No Idle<br>0x2 = 10: Smart Idle<br>0x3 = 11: Smart Idle wakeup capable |
| 1 | FREEEMU | R | 1h | Sensitivity to Emulation suspend input.<br>0x0: Module is sensitive to EMU suspend<br>0x1: Module not sensitive to EMU suspend |
| 0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.4 PLLSS_MMR_LOCK Register (offset = 40h) [reset = 1F125B64h]

PLLSS_MMR_LOCK is shown in Figure 2-30 and described in Table 2-46.

The PLLSS_MMR_LOCK Register is used allow/block writes to the registers in address region Start_Addess_x -End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x -End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of PLLSS_MMR_LOCK registers but they should follow the underlying security specified in each section, e.g. Firewall setting, Non Secure Supervisior etc.

**Figure 2-30. PLLSS_MMR_LOCK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MMR_LOCK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-1F125B64h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-46. PLLSS_MMR_LOCK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | MMR_LOCK | R/W | 1F125B64h | Lock/Unlock register for region 0x0008 - 0x0FFF<br>517622845 = UNLOCKED : All writes to 0x0008 - 0x0FFF is allowed.<br>521296740 = LOCKED : All writes to 0x0008 - 0x0FFF is blocked. |

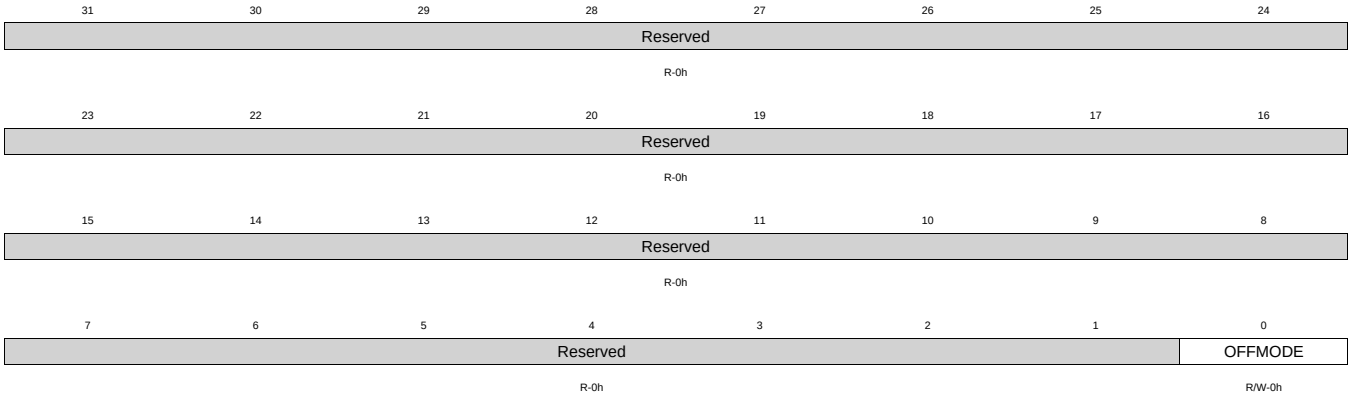

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.5 MPUPLL_PWRCTRL Register (offset = 48h) [reset = 0h]

MPUPLL_PWRCTRL is shown in Figure 2-31 and described in Table 2-47.

The MPU PLL Power Control register is used to control the power state of the MPU _PLL.

**Figure 2-31. MPUPLL_PWRCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-47. MPUPLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.6 MPUPLL_CLKCTRL Register (offset = 4Ch) [reset = 91E818h]

MPUPLL_CLKCTRL is shown in Figure 2-32 and described in Table 2-48.

The MPU PLL Clock Control register is used to control the different clock control state of the MPU_PLL

**Figure 2-32. MPUPLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | Reserved | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | Reserved | STBYRET | CLKOUTEN | Reserved | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| M3PWDNZ | STOPMODE | LOWCURRSTDBY | LPMODE | DRIFTGUARDEN | REGM4XEN | Reserved | RELAXED_LOCK |
| R/W-1h | R/W-1h | R/W-1h | R/W-0h | R/W-1h | R/W-0h | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-Ch | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-48. MPUPLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | Reserved | R | 0h | |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | Reserved | R | 0h | |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0: synchronously disables CLKOUT<br>0x1: synchronously enables CLKOUT |
| 19 | Reserved | R | 0h | |
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0: When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1: When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0: Asynchronous power down for M2 divider<br>0x1: M2 divider is functional |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-48. MPUPLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | M3PWDNZ | R/W | 1h | M3 divider power down mode<br>0x0 : Asynchronous power down for M3 divider<br>0x1 : M3 divider is functional |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | LOWCURRSTDBY | R/W | 1h | When in Lossclk/Stbyret/Idle<br>0x0 : Fast relock<br>0x1 : Slow relock |
| 12 | LPMODE | R/W | 0h | Can be set to 1 in cases with CLKINP/(N+1) less than or equal 1MHz and CLKINP*M/(N+1) less than or equal 100MHz |
| 11 | DRIFTGUARDEN | R/W | 1h | When RECAL status flag is asserted.1 Enables recalibration |
| 10 | REGM4XEN | R/W | 0h | Enable REGM*4 (Active High). Can be set to 1 only for CLKOUT*M2 greater than 150MHz when LPMODE = 0 and CLKOUT*M2 greater than 60MHz when LPMODE = 1 . This bit should not be changed on the fly in locked condition. INITIALIZATION should follow change of this bit. |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | Ch | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |

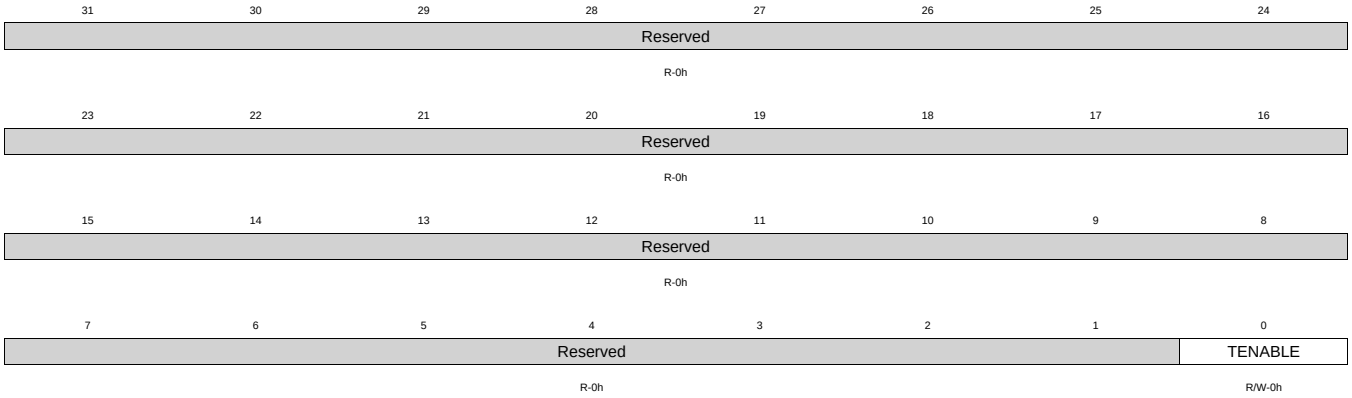

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.7 MPUPLL_TENABLE Register (offset = 50h) [reset = 0h]

MPUPLL_TENABLE is shown in Figure 2-33 and described in Table 2-49.

The MPU PLL TENABLE register is used to load the M, N dividers of MPU_PLL

**Figure 2-33. MPUPLL_TENABLE Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-49. MPUPLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N latch (active rise edge) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.8 MPUPLL_TENABLEDIV Register (offset = 54h) [reset = 0h]

MPUPLL_TENABLEDIV is shown in Figure 2-34 and described in Table 2-50.

The MPU PLL TENABLEDIV register is used to load the M2,N2 dividers of MPU_PLL

**Figure 2-34. MPUPLL_TENABLEDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | TENABLEDIV |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-50. MPUPLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.9 MPUPLL_M2NDIV Register (offset = 58h) [reset = 50013h]

MPUPLL_M2NDIV is shown in Figure 2-35 and described in Table 2-51.

MPU PLL M2NDIV register is for programming M2 and N values of MPU _PLL

**Figure 2-35. MPUPLL_M2NDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | M2 | | | | |
| R-0h | | | R/W-5h | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | N | | | | | | |
| R-0h | R/W-13h | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-51. MPUPLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-7 | Reserved | R | 0h | |
| 6-0 | N | R/W | 13h | Pre-divider is REGN+1 |

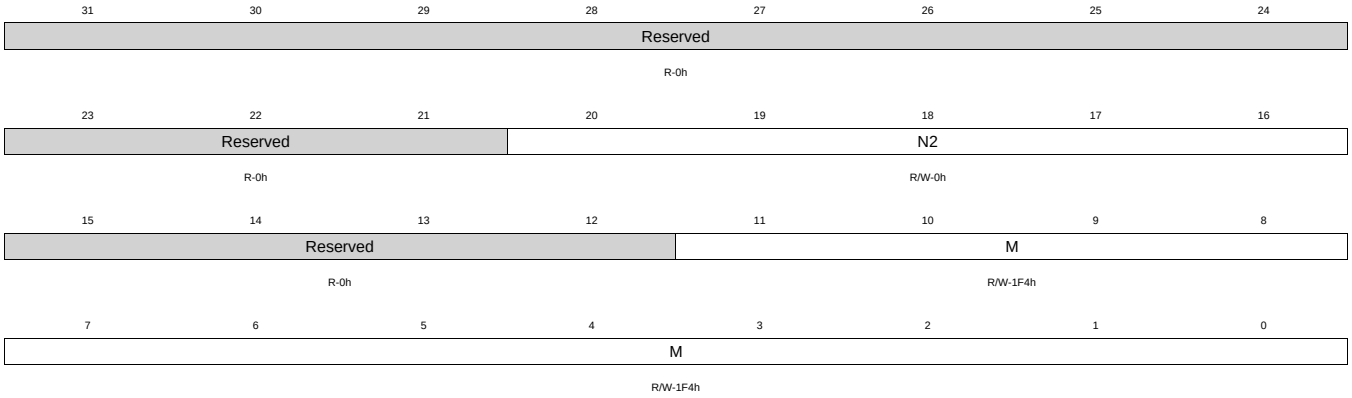

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.10 MPUPLL_MN2DIV Register (offset = 5Ch) [reset = 1F4h]

MPUPLL_MN2DIV is shown in Figure 2-36 and described in Table 2-52.

MPU PLL MN2DIV register is for programming M and N2 values of MPU _PLL

**Figure 2-36. MPUPLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-52. MPUPLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 1F4h | Feedback multiplier is REGM |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.11 MPUPLL_FRACDIV Register (offset = 60h) [reset = 0h]

MPUPLL_FRACDIV is shown in Figure 2-37 and described in Table 2-53.

The MPU PLL FRACDIV register is for controlling the fractional values of MPU _PLL

**Figure 2-37. MPUPLL_FRACDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | FRACTIONALM | |
| R-0h | | | | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-53. MPUPLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.12 MPUPLL_BWCTRL Register (offset = 64h) [reset = 0h]

MPUPLL_BWCTRL is shown in Figure 2-38 and described in Table 2-54.

The MPU PLL_BWCTRL register is for controlling the loop bandwidth of MPU _PLL

**Figure 2-38. MPUPLL_BWCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | BWCONTROL | | Reserved |
| R-0h | | | | | R/W-0h | | R-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-54. MPUPLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.13 MPUPLL_FRACCTRL Register (offset = 68h) [reset = 0h]

MPUPLL_FRACCTRL is shown in Figure 2-39 and described in Table 2-55.

Controlls the fractional portion of Modena PLL

**Figure 2-39. MPUPLL_FRACCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOWNSPREAD | ModFreqDividerExponent | | | ModFreqDividerMantissa | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ModFreqDividerMantissa | | | Reserved | DeltaMStepInteger | | DeltaMStepFraction | |
| R/W-0h | | | R-0h | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-55. MPUPLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20 | Reserved | R | 0h | |
| 19-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.14 MPUPLL_STATUS Register (offset = 6Ch) [reset = 121h]

MPUPLL_STATUS is shown in Figure 2-40 and described in Table 2-56.

The MPU PLL Status register is for viewing the status of the MPU PLL.

**Figure 2-40. MPUPLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | SSACK | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | LIMP | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-56. MPUPLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Reserved | R | 0h | |
| 30 | SSACK | R | 0h | PODPLL_MPU_SSACK |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0: indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1: indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | Indicates the enable/disable condition of CLKOUTEN<br>0x0 = CLKOUT gating completed<br>0x1 = CLKOUT enabling completed |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | Acknowledge for change to M2 divider. Toggles from 1-0 or 0-1 (depending on current value) once CLKOUT frequency change has completed. |
| 2 | LIMP | R | 0h | 0x1 In LIMP mode<br>0x0 In Stop Mode |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-56. MPUPLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |

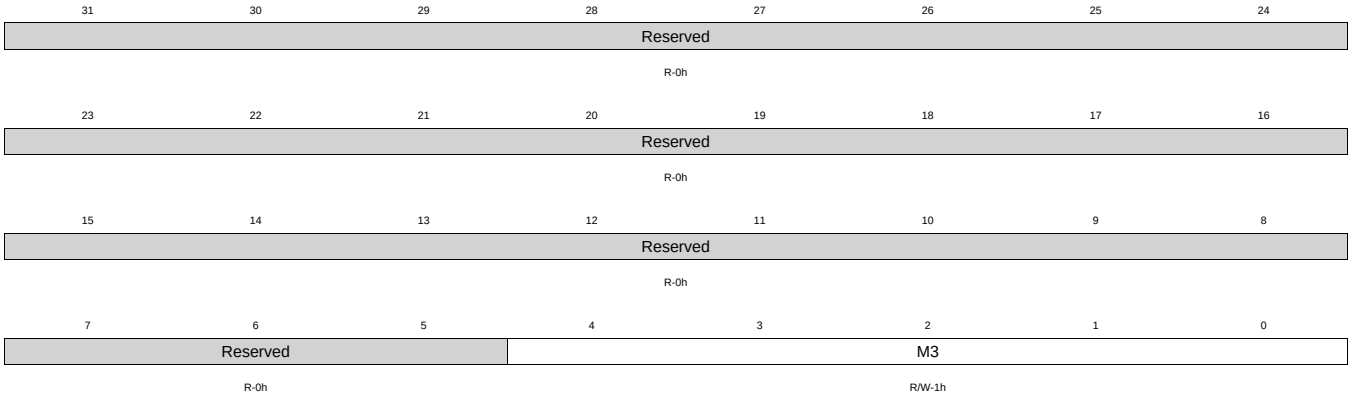

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.15 MPUPLL_M3DIV Register (offset = 70h) [reset = 1h]

MPUPLL_M3DIV is shown in Figure 2-41 and described in Table 2-57.

The MPU PLL_M3DIV register is for controlling the M3 divider of MPU _PLL

**Figure 2-41. MPUPLL_M3DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-57. MPUPLL_M3DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | Reserved | R | 0h | |
| 4-0 | M3 | R/W | 1h | Post Divider Reg M3 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.16 MPUPLL_RAMPCTRL Register (offset = 74h) [reset = 0h]

MPUPLL_RAMPCTRL is shown in Figure 2-42 and described in Table 2-58.

The MPUPLL_RAMPCTRL register is for controlling the ramping of MPU _PLL

**Figure 2-42. MPUPLL_RAMPCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CLKRAMPLEVEL | | CLKRAMPRATE | | |
| R-0h | | | R/W-0h | | R/W-0h | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | RELOCK_RAMP_EN |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-58. MPUPLL_RAMPCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20-19 | CLKRAMPLEVEL | R/W | 0h | Controls the ramp sequence.<br>0x0 = 00 : No ramping<br>0x1 = 01 : Bypass clk /Fout/8 / Fout/4 / Fout/2 / Fout<br>0x2 = 10 : Bypass clk / Fout/4 / Fout/2 /Fout/1.5/Fout<br>0x3 = 11 : Reserved |
| 18-16 | CLKRAMPRATE | R/W | 0h | Controls the time spent on each ramp step<br>0x0 = 000 : 2 REFCLKs<br>0x1 = 001 : 4 REFCLKs<br>0x2 = 010 : 8 REFCLKs<br>0x3 = 011 : 16 REFCLKs<br>0x4 = 100 : 32 REFCLKs<br>0x5 = 101 : 64 REFCLKs<br>0x6 = 110 : 128 REFCLKs<br>0x7 = 111 : 512 REFCLKs |
| 15-1 | Reserved | R | 0h | |
| 0 | RELOCK_RAMP_EN | R/W | 0h | 0 Clock ramping happens only during initial lock as controlled by CLKRAMPLEVEL bits. No clock ramping during relock 1 Clock ramping happens both during initial lock and relock |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.17 DSPPLL_PWRCTRL Register (offset = 80h) [reset = 30h]

DSPPLL_PWRCTRL is shown in Figure 2-43 and described in Table 2-59.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-43. DSPPLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-59. DSPPLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins. For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.18 DSPPLL_CLKCTRL Register (offset = 84h) [reset = 914824h]

DSPPLL_CLKCTRL is shown in Figure 2-44 and described in Table 2-60.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-44. DSPPLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-60. DSPPLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-60. DSPPLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0: When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1: When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0: Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0: Reserved<br>0x2: HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3: Reserved<br>0x4: HS1: DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5: Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0: FREQLOCK asserted when DC frequency error less than 1%<br>0x1: FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.19 DSPPLL_TENABLE Register (offset = 88h) [reset = 0h]

DSPPLL_TENABLE is shown in Figure 2-45 and described in Table 2-61.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-45. DSPPLL_TENABLE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | TENABLE |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-61. DSPPLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |

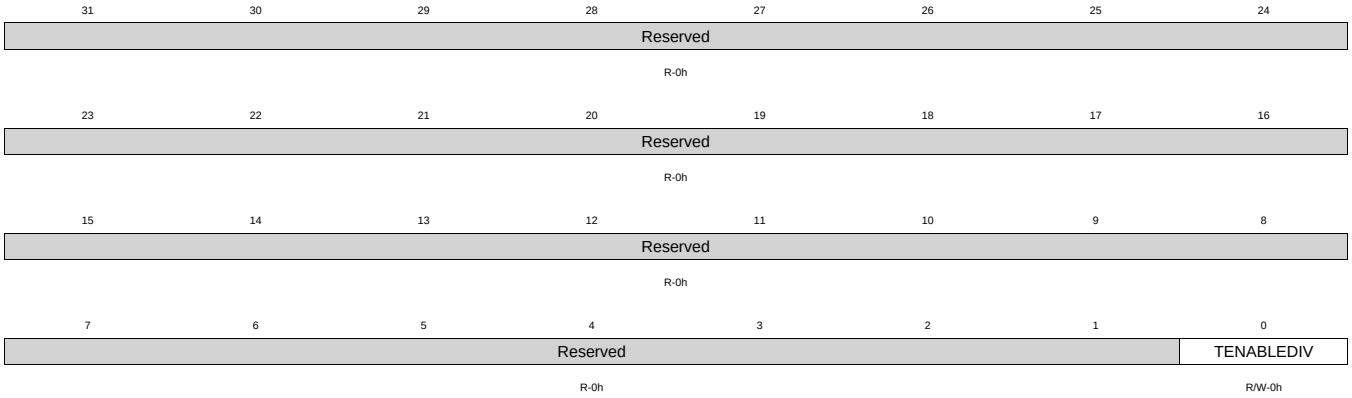

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.20 DSPPLL_TENABLEDIV Register (offset = 8Ch) [reset = 0h]

DSPPLL_TENABLEDIV is shown in Figure 2-46 and described in Table 2-62.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-46. DSPPLL_TENABLEDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-62. DSPPLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.21 DSPPLL_M2NDIV Register (offset = 90h) [reset = 50013h]

DSPPLL_M2NDIV is shown in Figure 2-47 and described in Table 2-63.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-47. DSPPLL_M2NDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | M2 | | | | | | |
| R-0h | R/W-5h | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| N | | | | | | | |
| R/W-13h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-63. DSPPLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |

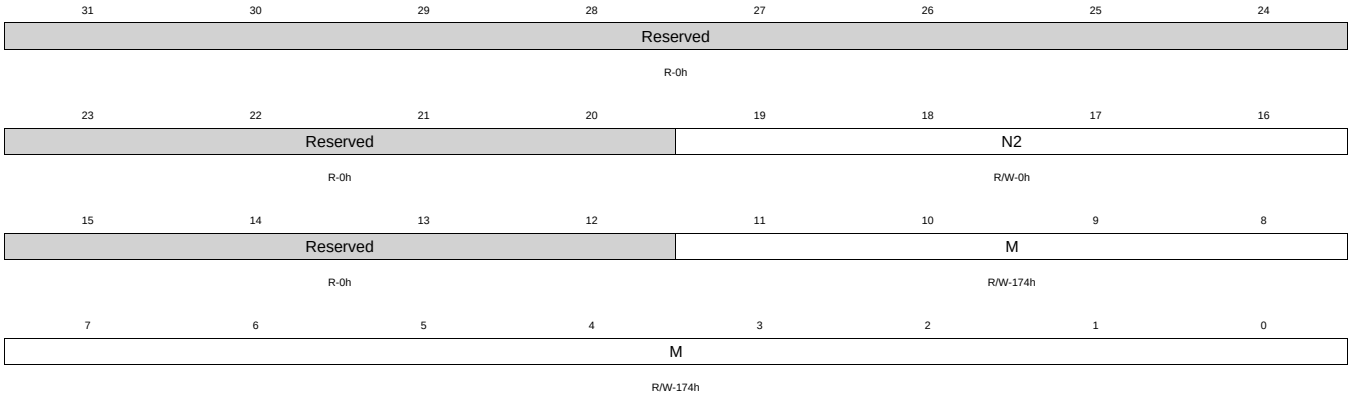

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.22 DSPPLL_MN2DIV Register (offset = 94h) [reset = 174h]

DSPPLL_MN2DIV is shown in Figure 2-48 and described in Table 2-64.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-48. DSPPLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-64. DSPPLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



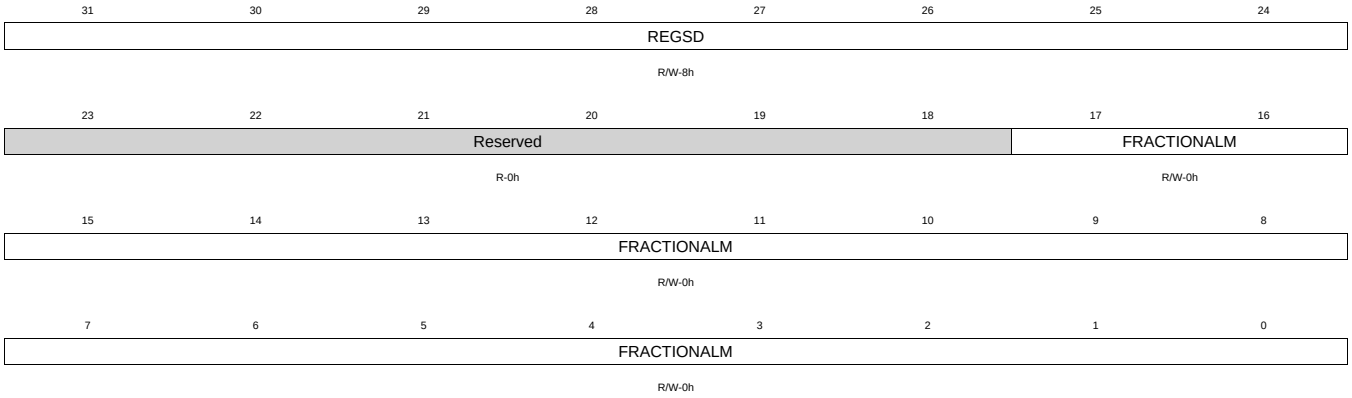
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.23 DSPPLL_FRACDIV Register (offset = 98h) [reset = 8000000h]

DSPPLL_FRACDIV is shown in Figure 2-49 and described in Table 2-65.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-49. DSPPLL_FRACDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-65. DSPPLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.24 DSPPLL_BWCTRL Register (offset = 9Ch) [reset = 0h]

DSPPLL_BWCTRL is shown in Figure 2-50 and described in Table 2-66.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-50. DSPPLL_BWCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | BWCONTROL | | BW_INCR_DECRZ |
| R-0h | | | | | R/W-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-66. DSPPLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0 : decrease BW<br>0x1 : increase BW |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.25 DSPPLL_FRACCTRL Register (offset = A0h) [reset = 0h]

DSPPLL_FRACCTRL is shown in Figure 2-51 and described in Table 2-67.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-51. DSPPLL_FRACCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOWNSPREAD | ModFreqDividerExponent | | | ModFreqDividerMantissa | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ModFreqDividerMantissa | | | DeltaMStepInteger | | | DeltaMStepFraction | |
| R/W-0h | | | R/W-0h | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-67. DSPPLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.26 DSPPLL_STATUS Register (offset = A4h) [reset = C0000121h]

DSPPLL_STATUS is shown in Figure 2-52 and described in Table 2-68.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-52. DSPPLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-68. DSPPLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0: indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1: indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | Indicates the enable/disable condition of CLKOUTEN<br>0x0 = CLKOUT gating completed<br>0x1 = CLKOUT enabling completed |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | Acknowledge for change to M2 divider. Toggles from 1-0 or 0-1 (depending on current value) once CLKOUT frequency change has completed. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-68. DSPPLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.27 SGXPLLL_PWRCTRL Register (offset = B0h) [reset = 30h]

SGXPLLL_PWRCTRL is shown in Figure 2-53 and described in Table 2-69.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-53. SGXPLLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-69. SGXPLLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.28 SGXPLL_CLKCTRL Register (offset = B4h) [reset = 914824h]

SGXPLL_CLKCTRL is shown in Figure 2-54 and described in Table 2-70.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-54. SGXPLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-70. SGXPLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-70. SGXPLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>1x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 : Reserved<br>0x2 : HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 : Reserved<br>0x4 : HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 : Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |

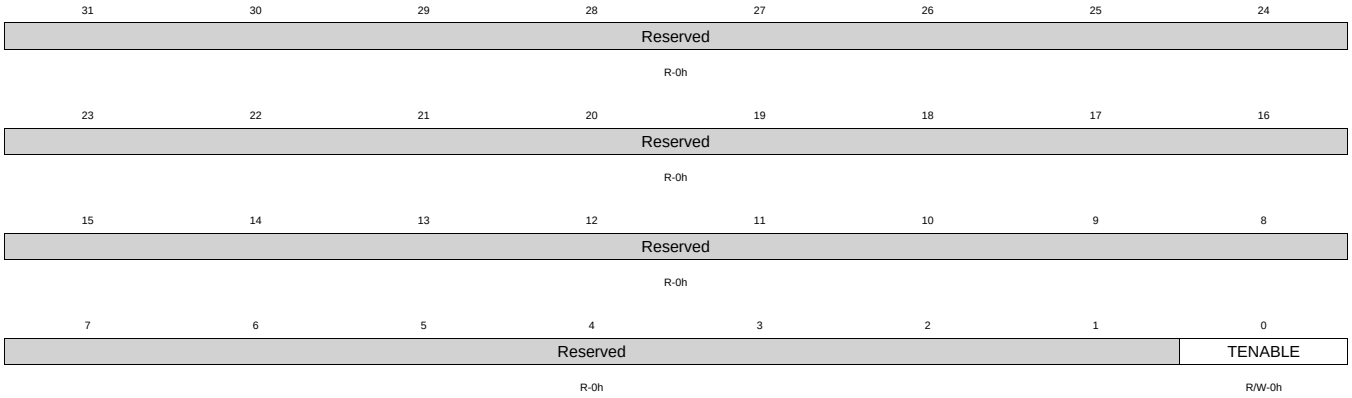

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.29 SGXPLLL_TENABLE Register (offset = B8h) [reset = 0h]

SGXPLLL_TENABLE is shown in Figure 2-55 and described in Table 2-71.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-55. SGXPLLL_TENABLE Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-71. SGXPLLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.30 SGXPLL_TENABLEDIV Register (offset = BCh) [reset = 0h]

SGXPLL_TENABLEDIV is shown in Figure 2-56 and described in Table 2-72.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-56. SGXPLL_TENABLEDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | TENABLEDIV |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-72. SGXPLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |



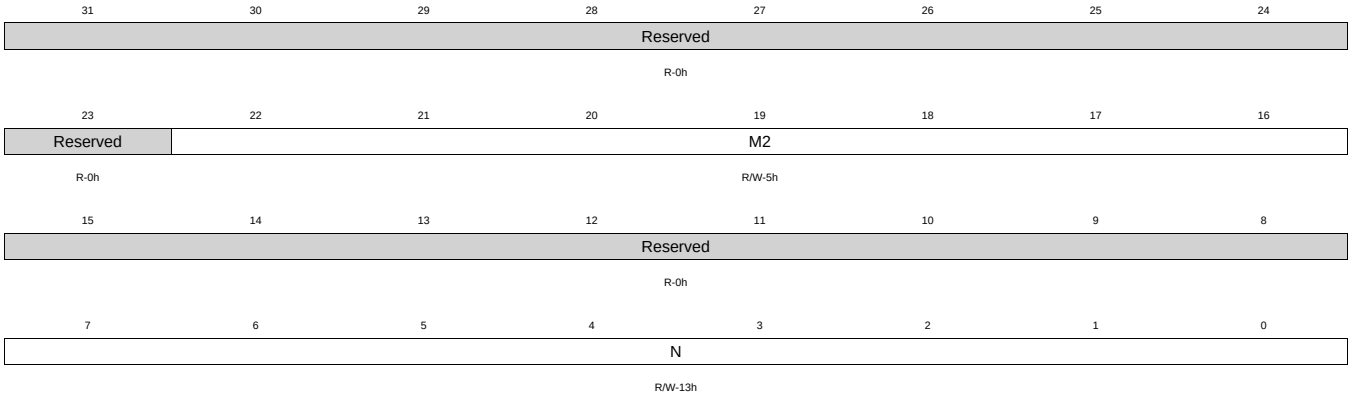
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.31 SGXPLL_M2NDIV Register (offset = C0h) [reset = 50013h]

SGXPLL_M2NDIV is shown in Figure 2-57 and described in Table 2-73.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-57. SGXPLL_M2NDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-73. SGXPLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |

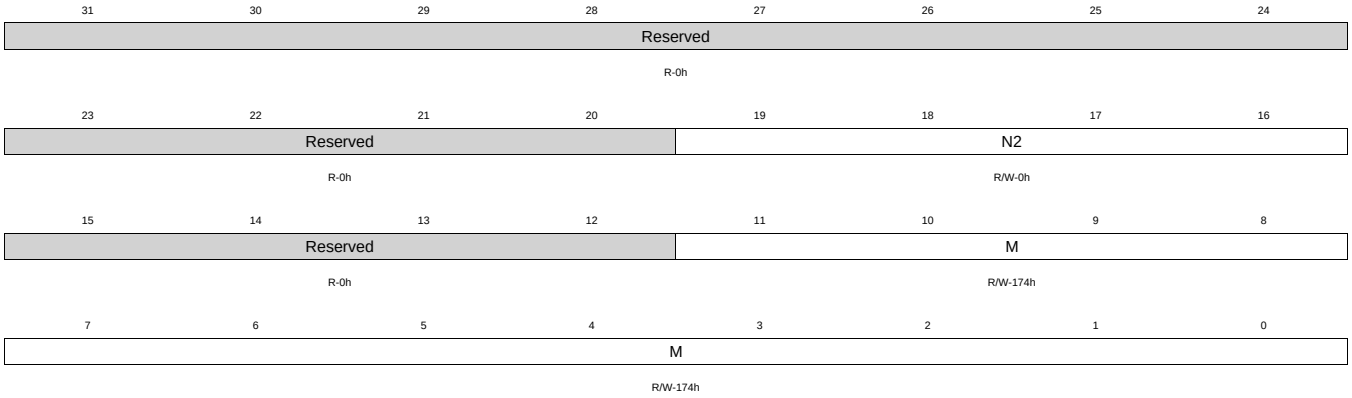

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.32 SGXPLL_MN2DIV Register (offset = C4h) [reset = 174h]

SGXPLL_MN2DIV is shown in Figure 2-58 and described in Table 2-74.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-58. SGXPLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-74. SGXPLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



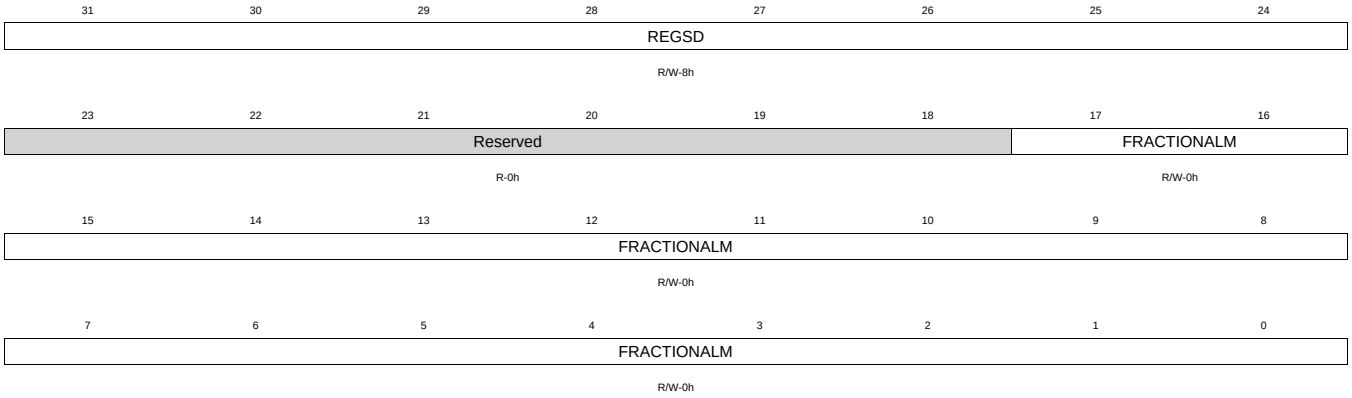
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.33 SGXPLL_FRACDIV Register (offset = C8h) [reset = 8000000h]

SGXPLL_FRACDIV is shown in Figure 2-59 and described in Table 2-75.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-59. SGXPLL_FRACDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-75. SGXPLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.34 SGXPLL_BWCTRL Register (offset = CCh) [reset = 0h]

SGXPLL_BWCTRL is shown in Figure 2-60 and described in Table 2-76.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-60. SGXPLL_BWCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | BWCONTROL | | BW_INCR_DECRZ |
| R-0h | | | | | R/W-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-76. SGXPLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0: decrease BW<br>0x1: increase BW |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.35 SGXPLL_FRACCTRL Register (offset = D0h) [reset = 0h]

SGXPLL_FRACCTRL is shown in Figure 2-61 and described in Table 2-77.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-61. SGXPLL_FRACCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOWNSPREAD | ModFreqDividerExponent | | | ModFreqDividerMantissa | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ModFreqDividerMantissa | | | DeltaMStepInteger | | | DeltaMStepFraction | |
| R/W-0h | | | R/W-0h | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-77. SGXPLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0: enables both side frequency spread about the programmed frequency.<br>0x1: enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.36 SGXPLL_STATUS Register (offset = D4h) [reset = C0000121h]

SGXPLL_STATUS is shown in Figure 2-62 and described in Table 2-78.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-62. SGXPLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-78. SGXPLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1/0 indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-78. SGXPLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.37 HDVICPPLL_PWRCTRL Register (offset = E0h) [reset = 30h]

HDVICPPLL_PWRCTRL is shown in Figure 2-63 and described in Table 2-79.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-63. HDVICPPLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-79. HDVICPPLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.38 HDVICPPLL_CLKCTRL Register (offset = E4h) [reset = 914824h]

HDVICPPLL_CLKCTRL is shown in Figure 2-64 and described in Table 2-80.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-64. HDVICPPLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-80. HDVICPPLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that dont have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-80. HDVICPPLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 : Reserved<br>0x2 : HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 : Reserved<br>0x4 : HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 : Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |

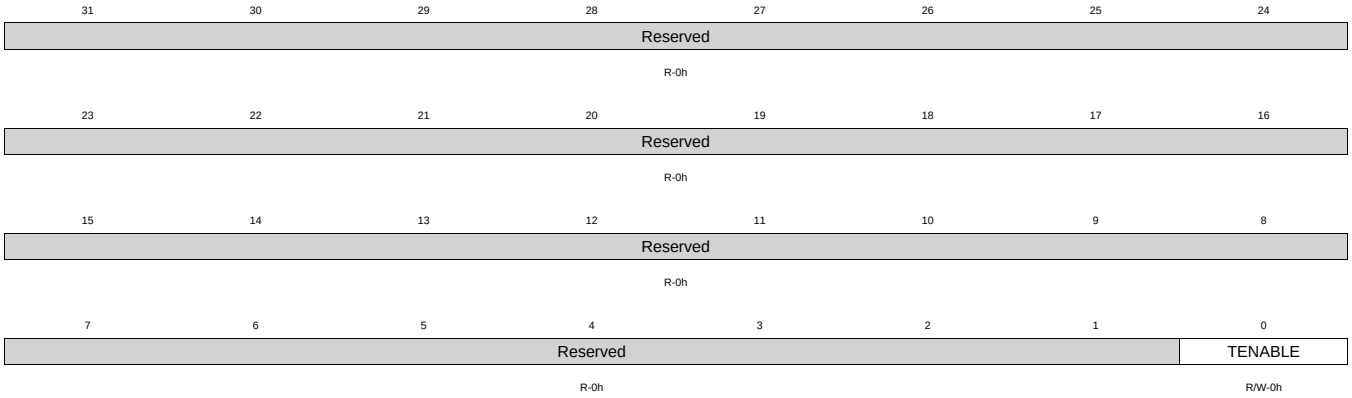

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.39 HDVICPPLL_TENABLE Register (offset = E8h) [reset = 0h]

HDVICPPLL_TENABLE is shown in Figure 2-65 and described in Table 2-81.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-65. HDVICPPLL_TENABLE Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-81. HDVICPPLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |

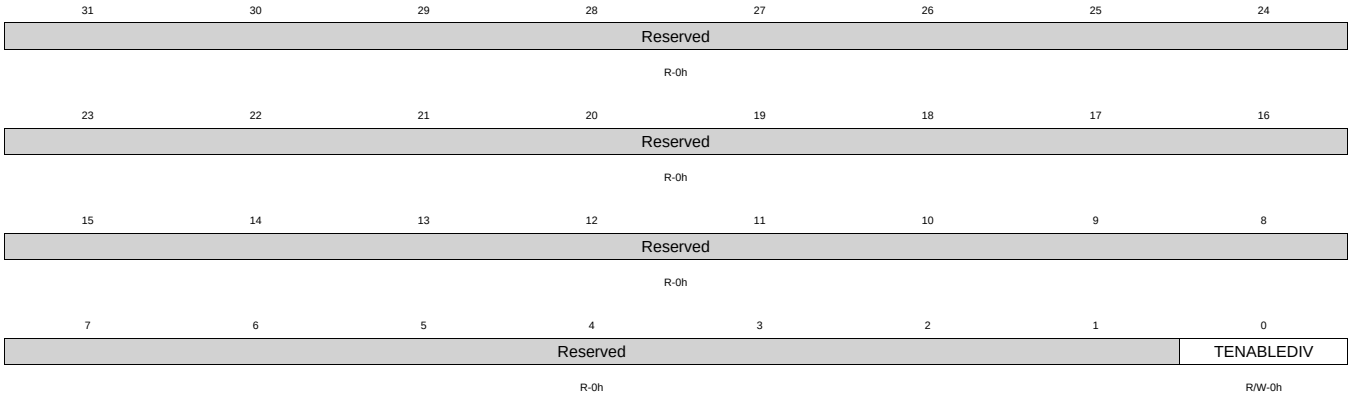

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.40 HDVICPPLL_TENABLEDIV Register (offset = ECh) [reset = 0h]

HDVICPPLL_TENABLEDIV is shown in Figure 2-66 and described in Table 2-82.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-66. HDVICPPLL_TENABLEDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-82. HDVICPPLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |

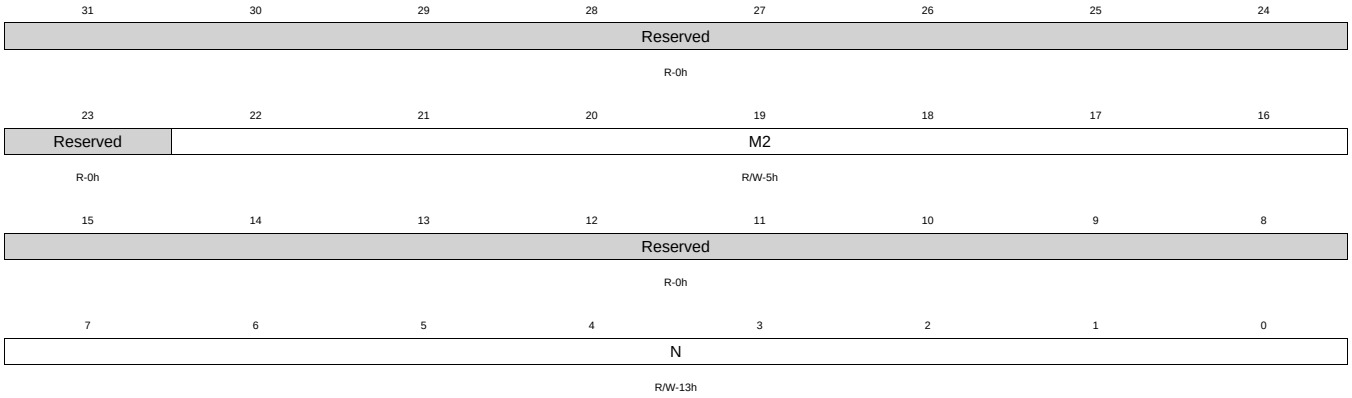

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.41 HDVICPPLL_M2NDIV Register (offset = F0h) [reset = 50013h]

HDVICPPLL_M2NDIV is shown in Figure 2-67 and described in Table 2-83.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-67. HDVICPPLL_M2NDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-83. HDVICPPLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |

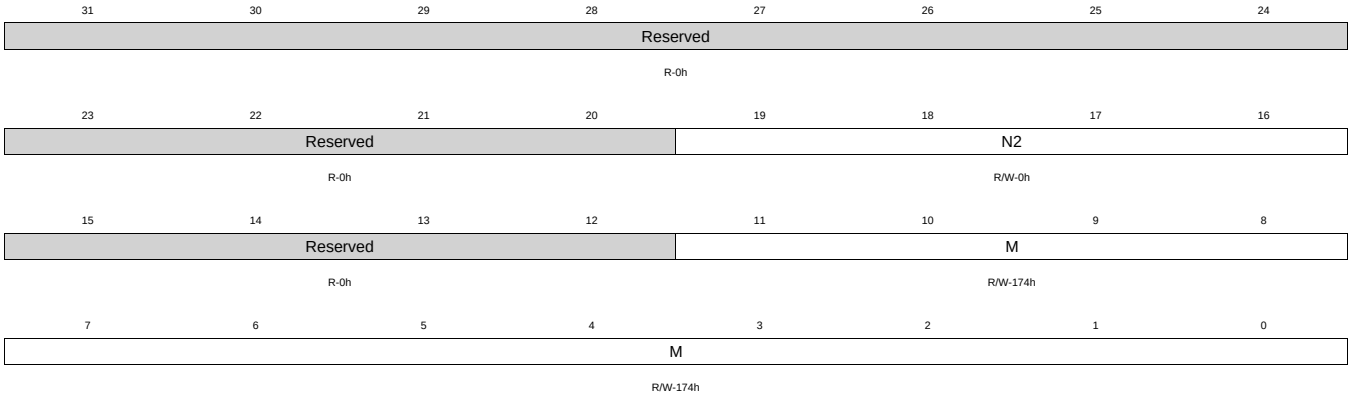

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.42 HDVICPPLL_MN2DIV Register (offset = F4h) [reset = 174h]

HDVICPPLL_MN2DIV is shown in Figure 2-68 and described in Table 2-84.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-68. HDVICPPLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-84. HDVICPPLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



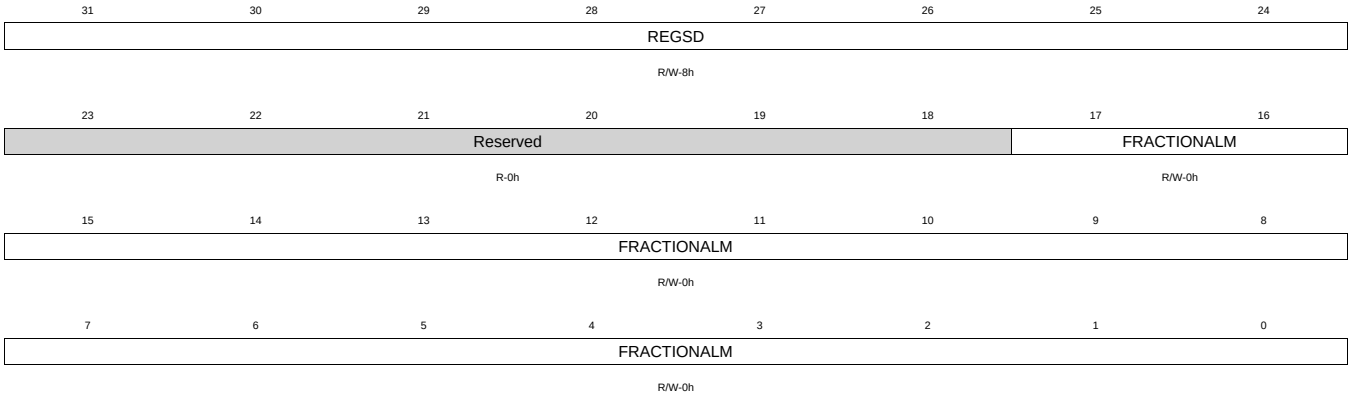
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.43 HDVICPPLL_FRACDIV Register (offset = F8h) [reset = 8000000h]

HDVICPPLL_FRACDIV is shown in Figure 2-69 and described in Table 2-85.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-69. HDVICPPLL_FRACDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-85. HDVICPPLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |

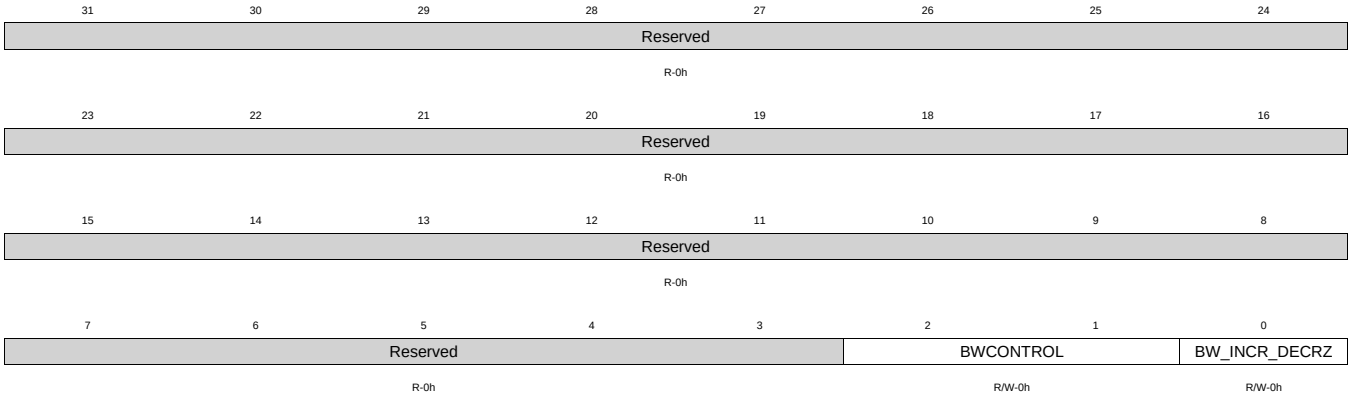

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.44 HDVICPPLL_BWCTRL Register (offset = FCh) [reset = 0h]

HDVICPPLL_BWCTRL is shown in Figure 2-70 and described in Table 2-86.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-70. HDVICPPLL_BWCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-86. HDVICPPLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0 : decrease BW<br>0x1 : increase BW |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.45 HDVICPPLL_FRACCTRL Register (offset = 100h) [reset = 0h]

HDVICPPLL_FRACCTRL is shown in Figure 2-71 and described in Table 2-87.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-71. HDVICPPLL_FRACCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOWNSPREAD | ModFreqDividerExponent | | | ModFreqDividerMantissa | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ModFreqDividerMantissa | | | DeltaMStepInteger | | | DeltaMStepFraction | |
| R/W-0h | | | R/W-0h | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-87. HDVICPPLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.46 HDVICPPLL_STATUS Register (offset = 104h) [reset = C0000121h]

HDVICPPLL_STATUS is shown in Figure 2-72 and described in Table 2-88.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-72. HDVICPPLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-88. HDVICPPLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1/0 indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-88. HDVICPPLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.47 L3PLL_PWRCTRL Register (offset = 110h) [reset = 30h]

L3PLL_PWRCTRL is shown in Figure 2-73 and described in Table 2-89.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-73. L3PLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-89. L3PLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.48 L3PLL_CLKCTRL Register (offset = 114h) [reset = 914824h]

L3PLL_CLKCTRL is shown in Figure 2-74 and described in Table 2-90.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-74. L3PLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-90. L3PLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-90. L3PLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 : Reserved<br>0x2 : HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 : Reserved<br>0x4 : HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 : Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |

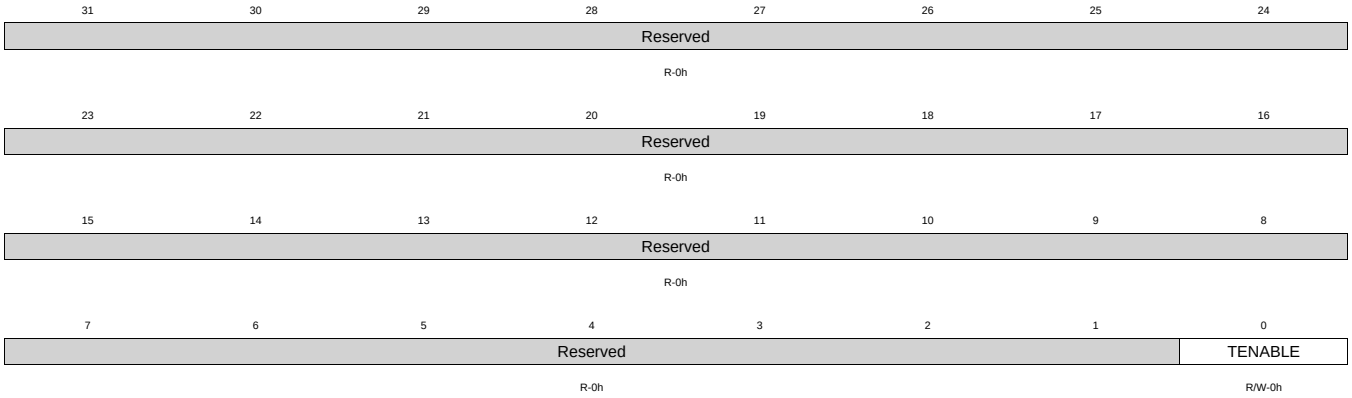

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.49 L3PLL_TENABLE Register (offset = 118h) [reset = 0h]

L3PLL_TENABLE is shown in Figure 2-75 and described in Table 2-91.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-75. L3PLL_TENABLE Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-91. L3PLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |

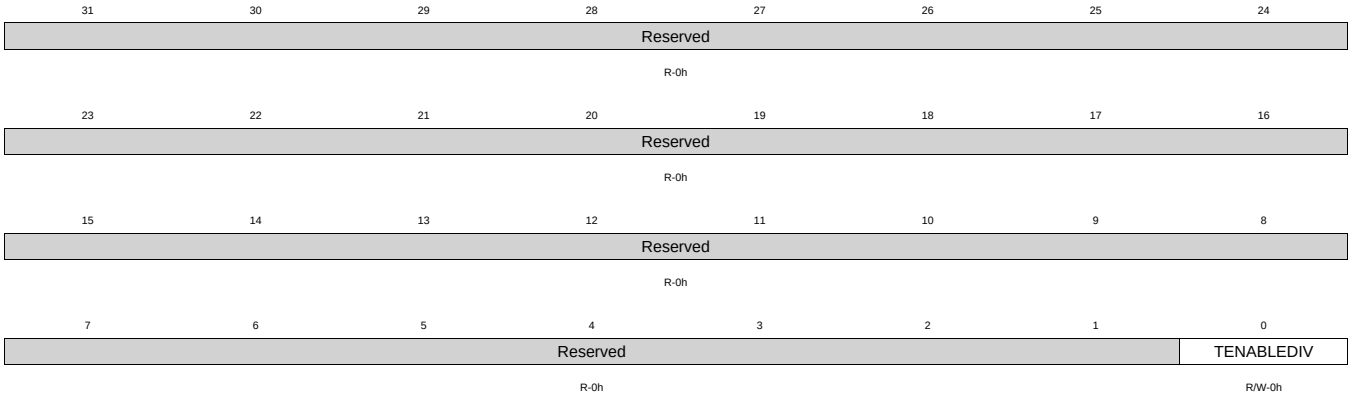

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.50 L3PLL_TENABLEDIV Register (offset = 11Ch) [reset = 0h]

L3PLL_TENABLEDIV is shown in Figure 2-76 and described in Table 2-92.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-76. L3PLL_TENABLEDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-92. L3PLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |

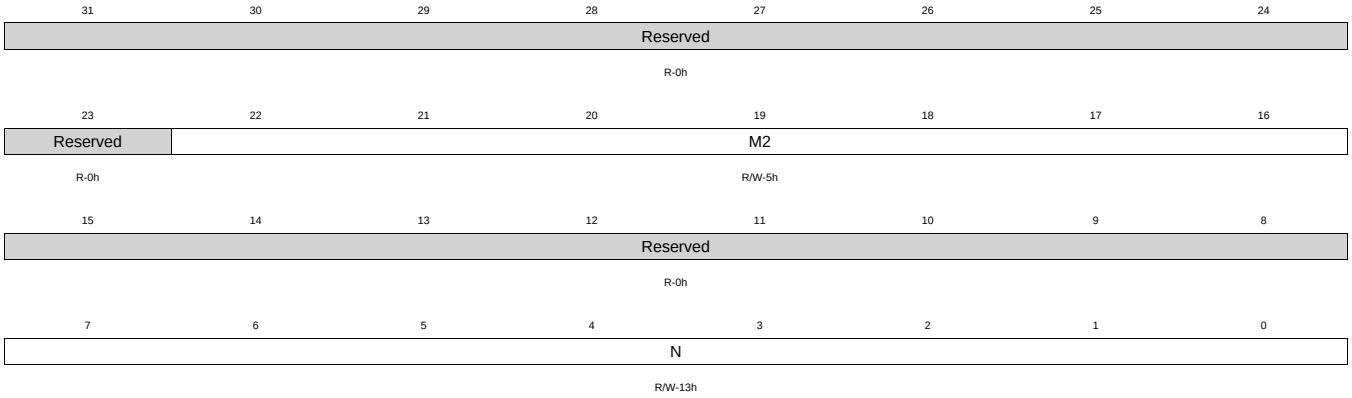

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.51 L3PLL_M2NDIV Register (offset = 120h) [reset = 50013h]

L3PLL_M2NDIV is shown in Figure 2-77 and described in Table 2-93.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-77. L3PLL_M2NDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-93. L3PLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |

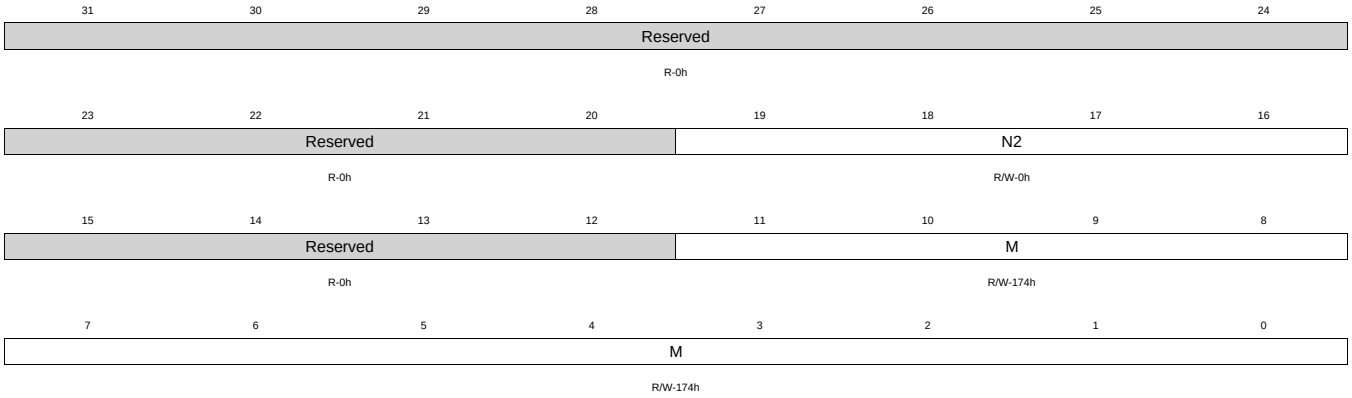

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.52 L3PLL_MN2DIV Register (offset = 124h) [reset = 174h]

L3PLL_MN2DIV is shown in Figure 2-78 and described in Table 2-94.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-78. L3PLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-94. L3PLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.53 L3PLL_FRACDIV Register (offset = 128h) [reset = 8000000h]

L3PLL_FRACDIV is shown in Figure 2-79 and described in Table 2-95.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-79. L3PLL_FRACDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REGSD | | | | | | | |
| R/W-8h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | FRACTIONALM | |
| R-0h | | | | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-95. L3PLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.54 L3PLL_BWCTRL Register (offset = 12Ch) [reset = 0h]

L3PLL_BWCTRL is shown in Figure 2-80 and described in Table 2-96.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-80. L3PLL_BWCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | BWCONTROL | | BW_INCR_DECRZ |
| R-0h | | | | | R/W-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-96. L3PLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0 : decrease BW<br>0x1 : increase BW |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.55 L3PLL_FRACCTRL Register (offset = 130h) [reset = 0h]

L3PLL_FRACCTRL is shown in Figure 2-81 and described in Table 2-97.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-81. L3PLL_FRACCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOWNSPREAD | ModFreqDividerExponent | | | ModFreqDividerMantissa | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ModFreqDividerMantissa | | | DeltaMStepInteger | | | DeltaMStepFraction | |
| R/W-0h | | | R/W-0h | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-97. L3PLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.56 L3PLL_STATUS Register (offset = 134h) [reset = C0000121h]

L3PLL_STATUS is shown in Figure 2-82 and described in Table 2-98.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-82. L3PLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-98. L3PLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1/0 indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-98. L3PLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.57 ISPPLL_PWRCTRL Register (offset = 140h) [reset = 30h]

ISPPLL_PWRCTRL is shown in Figure 2-83 and described in Table 2-99.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-83. ISPPLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-99. ISPPLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.58 ISPPLL_CLKCTRL Register (offset = 144h) [reset = 914824h]

ISPPLL_CLKCTRL is shown in Figure 2-84 and described in Table 2-100.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-84. ISPPLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-100. ISPPLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-100. ISPPLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 : Reserved<br>0x2 : HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 : Reserved<br>0x4 : HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 : Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.59 ISPPLL_TENABLE Register (offset = 148h) [reset = 0h]

ISPPLL_TENABLE is shown in Figure 2-85 and described in Table 2-101.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-85. ISPPLL_TENABLE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | TENABLE |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-101. ISPPLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |

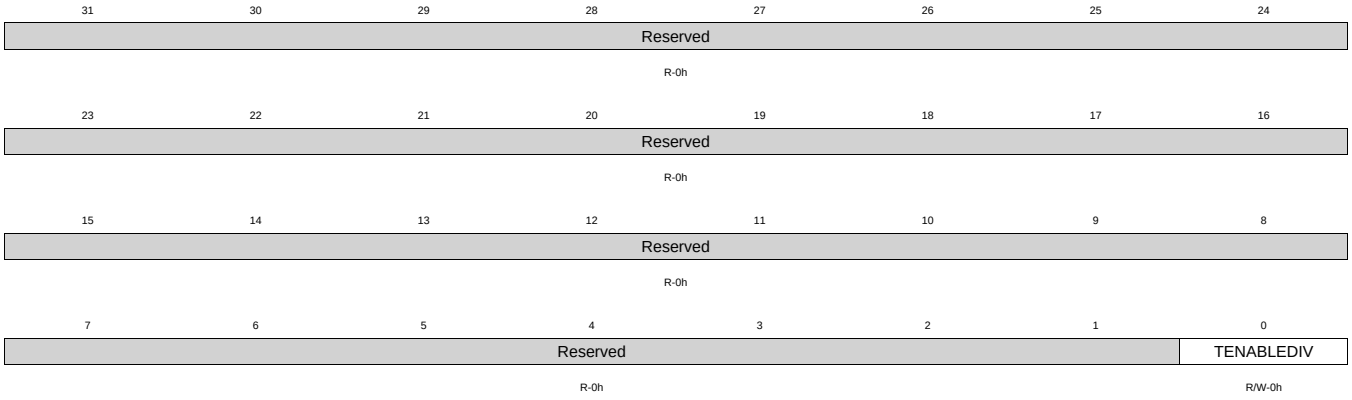

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.60 ISPPLL_TENABLEDIV Register (offset = 14Ch) [reset = 0h]

ISPPLL_TENABLEDIV is shown in Figure 2-86 and described in Table 2-102.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-86. ISPPLL_TENABLEDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-102. ISPPLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.61 ISPPLL_M2NDIV Register (offset = 150h) [reset = 50013h]

ISPPLL_M2NDIV is shown in Figure 2-87 and described in Table 2-103.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-87. ISPPLL_M2NDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | M2 | | | | | | |
| R-0h | R/W-5h | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| N | | | | | | | |
| R/W-13h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-103. ISPPLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |

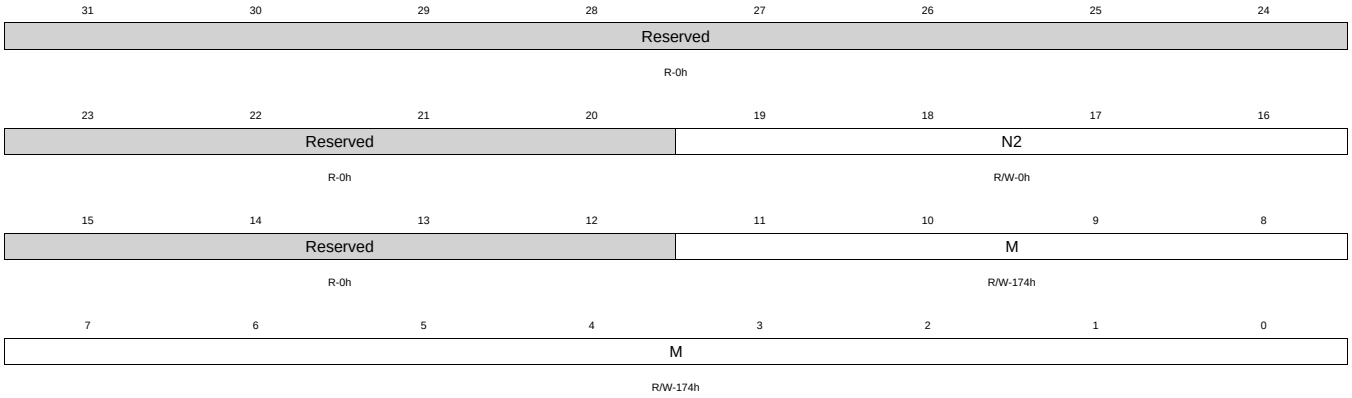

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.62 ISPPLL_MN2DIV Register (offset = 154h) [reset = 174h]

ISPPLL_MN2DIV is shown in Figure 2-88 and described in Table 2-104.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-88. ISPPLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-104. ISPPLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



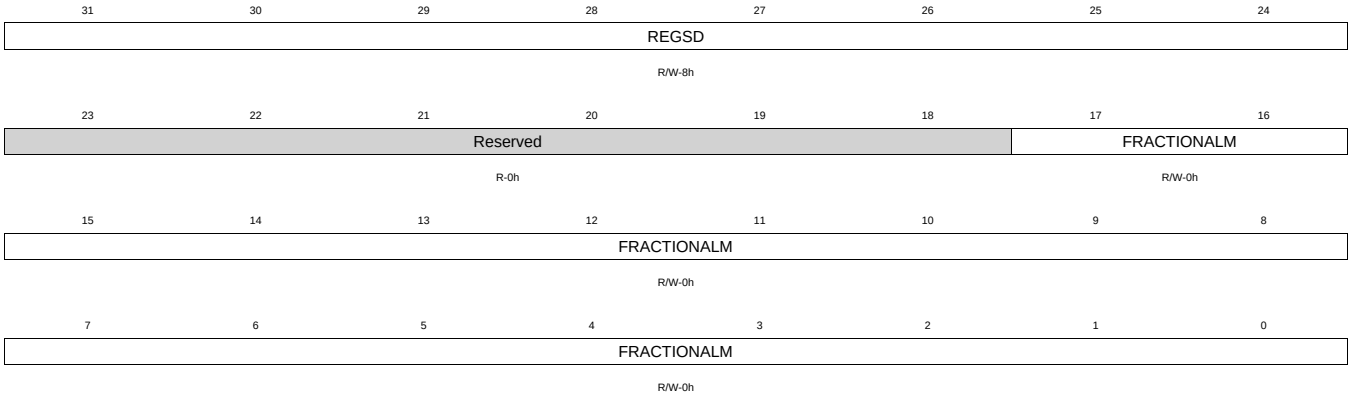
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.63 ISPPLL_FRACDIV Register (offset = 158h) [reset = 8000000h]

ISPPLL_FRACDIV is shown in Figure 2-89 and described in Table 2-105.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-89. ISPPLL_FRACDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-105. ISPPLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.64 ISPPLL_BWCTRL Register (offset = 15Ch) [reset = 0h]

ISPPLL_BWCTRL is shown in Figure 2-90 and described in Table 2-106.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-90. ISPPLL_BWCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | BWCONTROL | | BW_INCR_DECRZ |
| R-0h | | | | | R/W-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-106. ISPPLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0 : decrease BW<br>0x1 : increase BW |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.65 ISPPLL_FRACCTRL Register (offset = 160h) [reset = 0h]

ISPPLL_FRACCTRL is shown in Figure 2-91 and described in Table 2-107.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-91. ISPPLL_FRACCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOWNSPREAD | ModFreqDividerExponent | | | ModFreqDividerMantissa | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ModFreqDividerMantissa | | | DeltaMStepInteger | | | DeltaMStepFraction | |
| R/W-0h | | | R/W-0h | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-107. ISPPLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.66 ISPPLL_STATUS Register (offset = 164h) [reset = C0000121h]

ISPPLL_STATUS is shown in Figure 2-92 and described in Table 2-108.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-92. ISPPLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-108. ISPPLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1/0 indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-108. ISPPLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.67 DSSPLL_PWRCTRL Register (offset = 170h) [reset = 30h]

DSSPLL_PWRCTRL is shown in Figure 2-93 and described in Table 2-109.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-93. DSSPLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-109. DSSPLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.68 DSSPLL_CLKCTRL Register (offset = 174h) [reset = 914824h]

DSSPLL_CLKCTRL is shown in Figure 2-94 and described in Table 2-110.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-94. DSSPLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-110. DSSPLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-110. DSSPLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 Reserved<br>0x2 HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 Reserved<br>0x4 HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 Reserved2 |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |

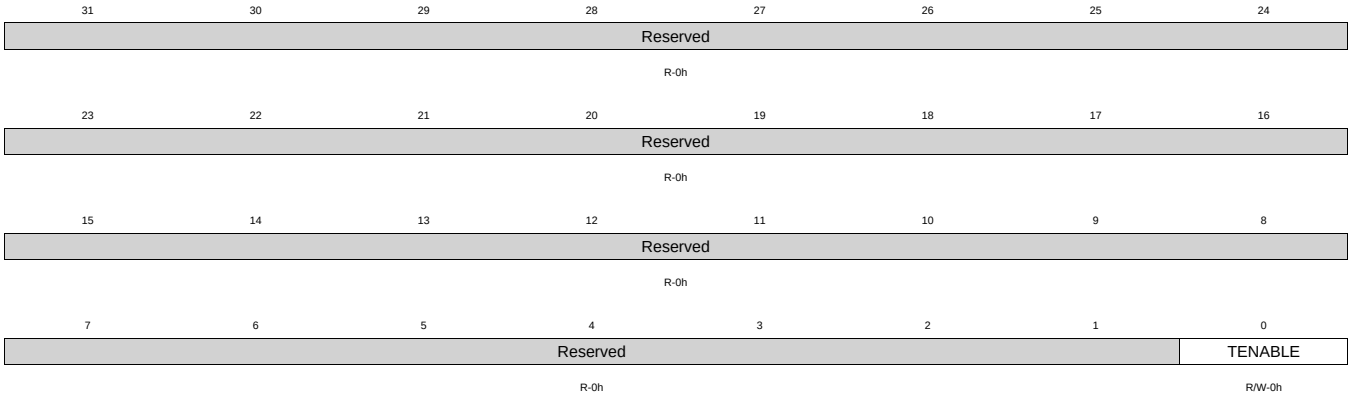

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.69 DSSPLL_TENABLE Register (offset = 178h) [reset = 0h]

DSSPLL_TENABLE is shown in Figure 2-95 and described in Table 2-111.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-95. DSSPLL_TENABLE Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-111. DSSPLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |

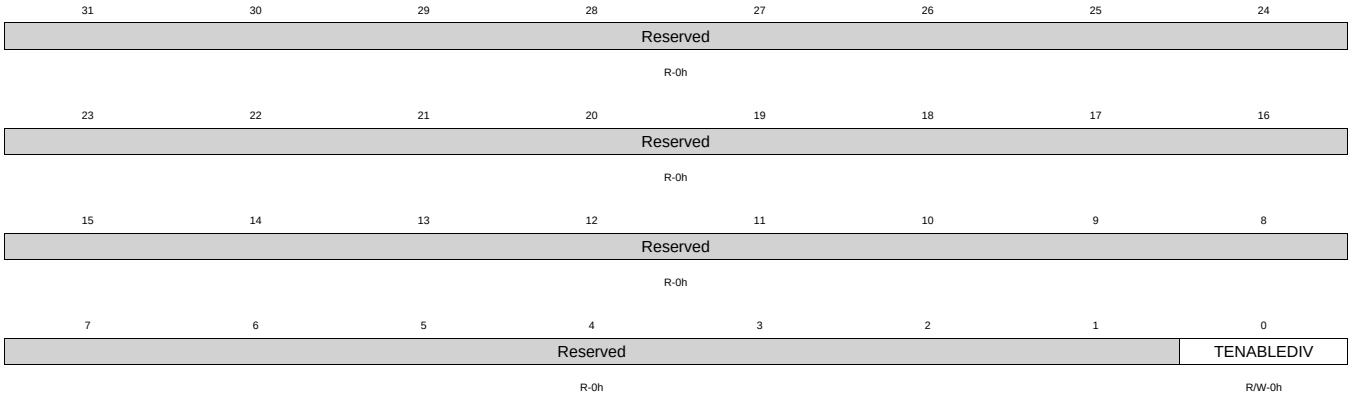

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.70 DSSPLL_TENABLEDIV Register (offset = 17Ch) [reset = 0h]

DSSPLL_TENABLEDIV is shown in Figure 2-96 and described in Table 2-112.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-96. DSSPLL_TENABLEDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-112. DSSPLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.71 DSSPLL_M2NDIV Register (offset = 180h) [reset = 50013h]

DSSPLL_M2NDIV is shown in Figure 2-97 and described in Table 2-113.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-97. DSSPLL_M2NDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | M2 | | | | | | |
| R-0h | R/W-5h | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| N | | | | | | | |
| R/W-13h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-113. DSSPLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.72 DSSPLL_MN2DIV Register (offset = 184h) [reset = 174h]

DSSPLL_MN2DIV is shown in Figure 2-98 and described in Table 2-114.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-98. DSSPLL_MN2DIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | N2 | | | |
| R-0h | | | | R/W-0h | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | M | | | |
| R-0h | | | | R/W-174h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| M | | | | | | | |
| R/W-174h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-114. DSSPLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



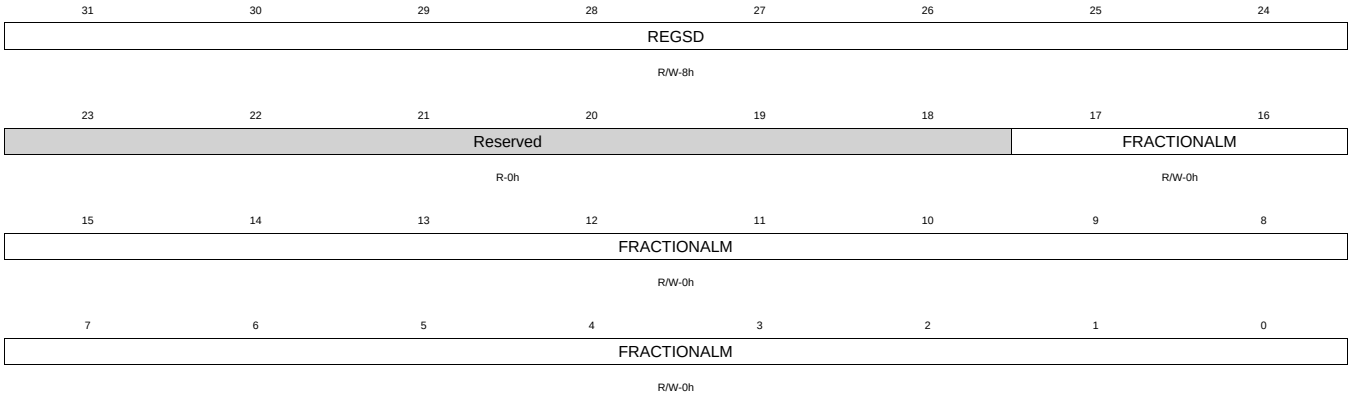
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.73 DSSPLL_FRACDIV Register (offset = 188h) [reset = 8000000h]

DSSPLL_FRACDIV is shown in Figure 2-99 and described in Table 2-115.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-99. DSSPLL_FRACDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-115. DSSPLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |

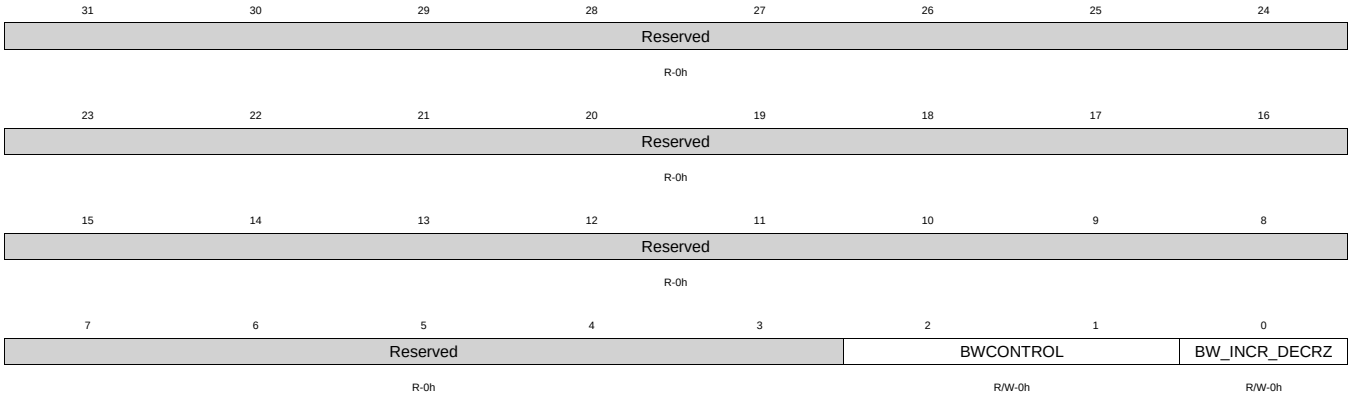

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.74 DSSPLL_BWCTRL Register (offset = 18Ch) [reset = 0h]

DSSPLL_BWCTRL is shown in Figure 2-100 and described in Table 2-116.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-100. DSSPLL_BWCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-116. DSSPLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0 : decrease BW<br>0x1 : increase BW |



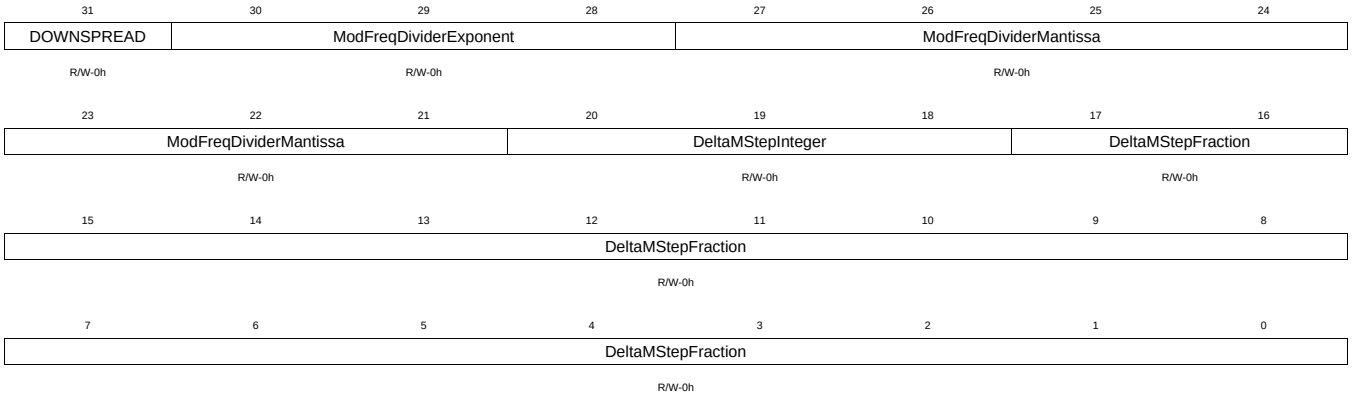
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.75 DSSPLL_FRACCTRL Register (offset = 190h) [reset = 0h]

DSSPLL_FRACCTRL is shown in Figure 2-101 and described in Table 2-117.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-101. DSSPLL_FRACCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-117. DSSPLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.76 DSSPLL_STATUS Register (offset = 194h) [reset = C0000121h]

DSSPLL_STATUS is shown in Figure 2-102 and described in Table 2-118.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-102. DSSPLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-118. DSSPLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1/0 indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-118. DSSPLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.77 VIDEO0PLL_PWRCTRL Register (offset = 1A0h) [reset = 30h]

VIDEO0PLL_PWRCTRL is shown in Figure 2-103 and described in Table 2-119.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-103. VIDEO0PLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-119. VIDEO0PLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.78 VIDEO0PLL_CLKCTRL Register (offset = 1A4h) [reset = 914824h]

VIDEO0PLL_CLKCTRL is shown in Figure 2-104 and described in Table 2-120.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-104. VIDEO0PLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-120. VIDEO0PLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-120. VIDEO0PLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 : Reserved<br>0x2 : HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 : Reserved<br>0x4 : HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 : Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.79 VIDEO0PLL_TENABLE Register (offset = 1A8h) [reset = 0h]

VIDEO0PLL_TENABLE is shown in Figure 2-105 and described in Table 2-121.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-105. VIDEO0PLL_TENABLE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | TENABLE |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-121. VIDEO0PLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |

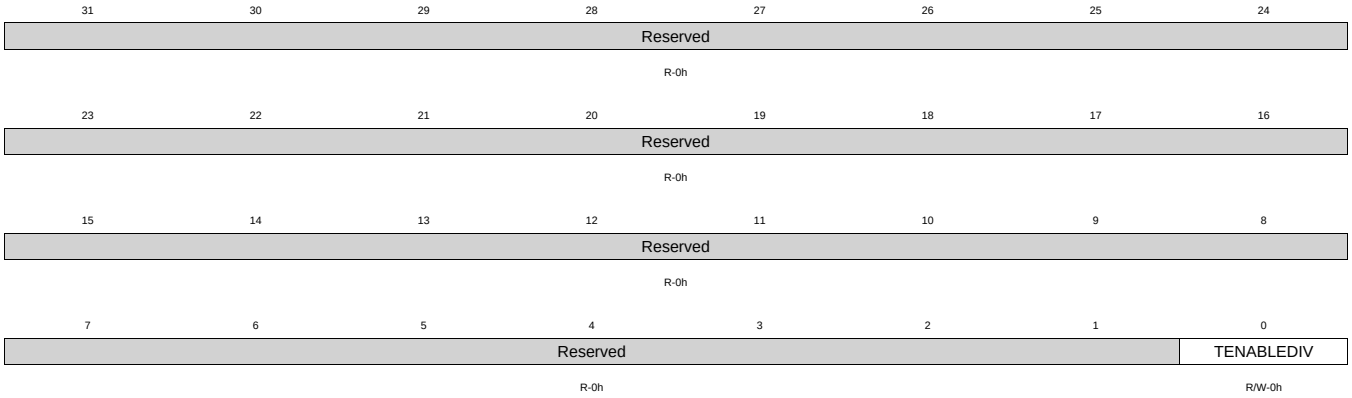

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.80 VIDEO0PLL_TENABLEDIV Register (offset = 1ACh) [reset = 0h]

VIDEO0PLL_TENABLEDIV is shown in Figure 2-106 and described in Table 2-122.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-106. VIDEO0PLL_TENABLEDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-122. VIDEO0PLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.81 VIDEO0PLL_M2NDIV Register (offset = 1B0h) [reset = 50013h]

VIDEO0PLL_M2NDIV is shown in Figure 2-107 and described in Table 2-123.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-107. VIDEO0PLL_M2NDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | M2 | | | | | | |
| R-0h | R/W-5h | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| N | | | | | | | |
| R/W-13h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-123. VIDEO0PLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |

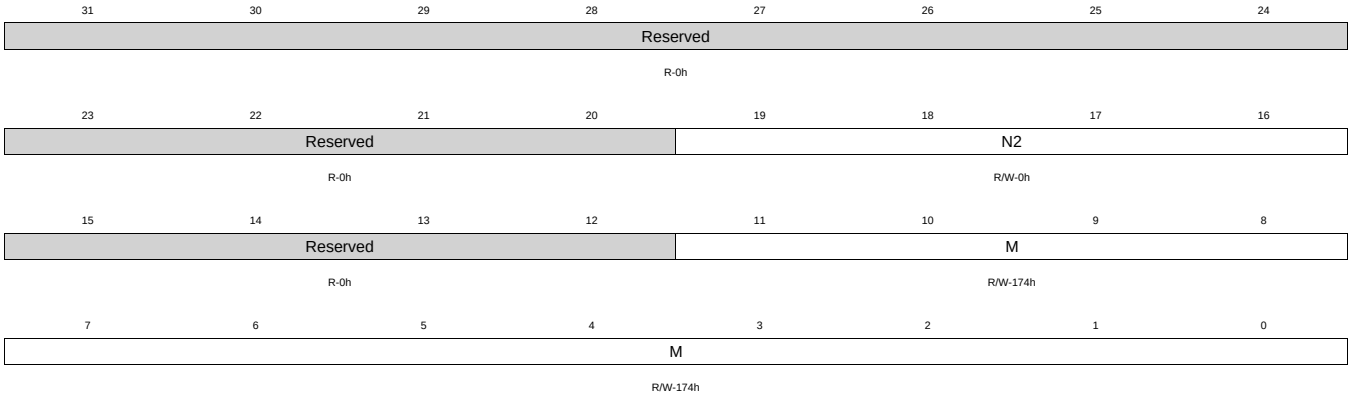

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.82 VIDEO0PLL_MN2DIV Register (offset = 1B4h) [reset = 174h]

VIDEO0PLL_MN2DIV is shown in Figure 2-108 and described in Table 2-124.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-108. VIDEO0PLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-124. VIDEO0PLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.83 VIDEO0PLL_FRACDIV Register (offset = 1B8h) [reset = 8000000h]

VIDEO0PLL_FRACDIV is shown in Figure 2-109 and described in Table 2-125.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-109. VIDEO0PLL_FRACDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REGSD | | | | | | | |
| R/W-8h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | FRACTIONALM | |
| R-0h | | | | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-125. VIDEO0PLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.84 VIDEO0PLL_BWCTRL Register (offset = 1BCh) [reset = 0h]

VIDEO0PLL_BWCTRL is shown in Figure 2-110 and described in Table 2-126.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-110. VIDEO0PLL_BWCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | BWCONTROL | | BW_INCR_DECRZ |
| R-0h | | | | | R/W-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-126. VIDEO0PLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0 : decrease BW<br>0x1 : increase BW |



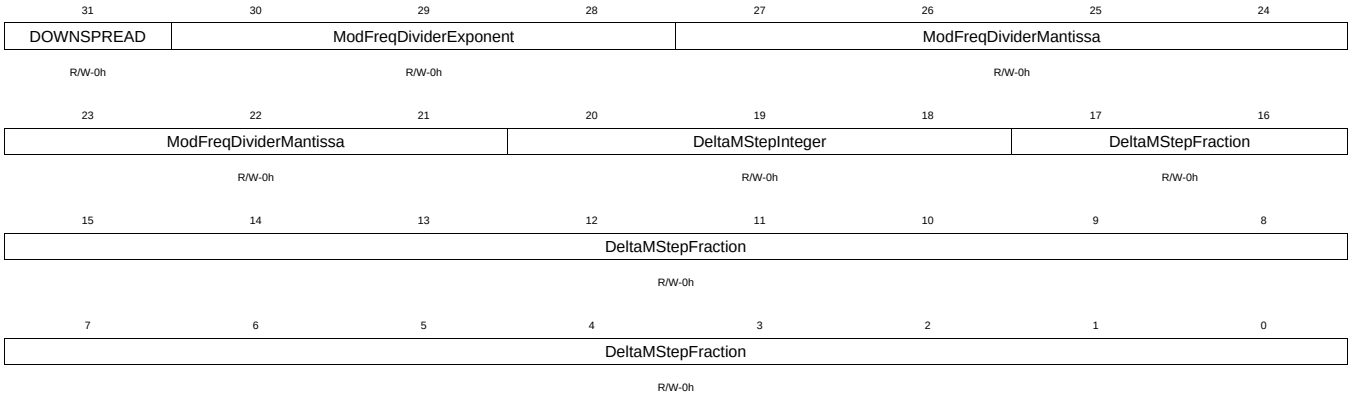
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.85 VIDEO0PLL_FRACCTRL Register (offset = 1C0h) [reset = 0h]

VIDEO0PLL_FRACCTRL is shown in Figure 2-111 and described in Table 2-127.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-111. VIDEO0PLL_FRACCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-127. VIDEO0PLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.86 VIDEO0PLL_STATUS Register (offset = 1C4h) [reset = C0000121h]

VIDEO0PLL_STATUS is shown in Figure 2-112 and described in Table 2-128.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-112. VIDEO0PLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-128. VIDEO0PLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1 /0indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-128. VIDEO0PLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.87 VIDEO1PLL_PWRCTRL Register (offset = 1D0h) [reset = 30h]

VIDEO1PLL_PWRCTRL is shown in Figure 2-113 and described in Table 2-129.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-113. VIDEO1PLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-129. VIDEO1PLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.88 VIDEO1PLL_CLKCTRL Register (offset = 1D4h) [reset = 914824h]

VIDEO1PLL_CLKCTRL is shown in Figure 2-114 and described in Table 2-130.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-114. VIDEO1PLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-130. VIDEO1PLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-130. VIDEO1PLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 : Reserved<br>0x2 : HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 : Reserved<br>0x4 : HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 : Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.89 VIDEO1PLL_TENABLE Register (offset = 1D8h) [reset = 0h]

VIDEO1PLL_TENABLE is shown in Figure 2-115 and described in Table 2-131.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-115. VIDEO1PLL_TENABLE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | TENABLE |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-131. VIDEO1PLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.90 VIDEO1PLL_TENABLEDIV Register (offset = 1DCh) [reset = 0h]

VIDEO1PLL_TENABLEDIV is shown in Figure 2-116 and described in Table 2-132.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-116. VIDEO1PLL_TENABLEDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TENABLEDIV |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-132. VIDEO1PLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |



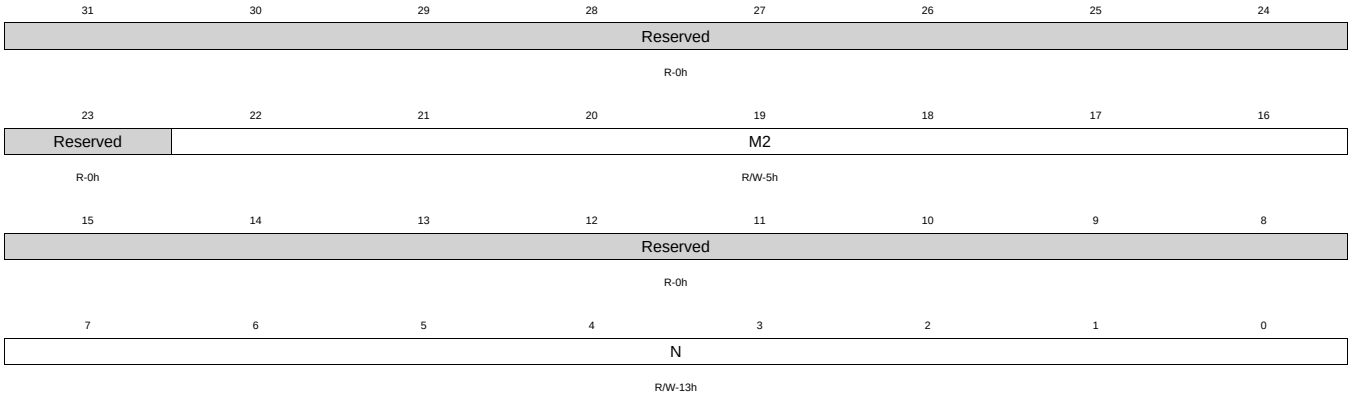
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.91 VIDEO1PLL_M2NDIV Register (offset = 1E0h) [reset = 50013h]

VIDEO1PLL_M2NDIV is shown in Figure 2-117 and described in Table 2-133.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-117. VIDEO1PLL_M2NDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-133. VIDEO1PLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |

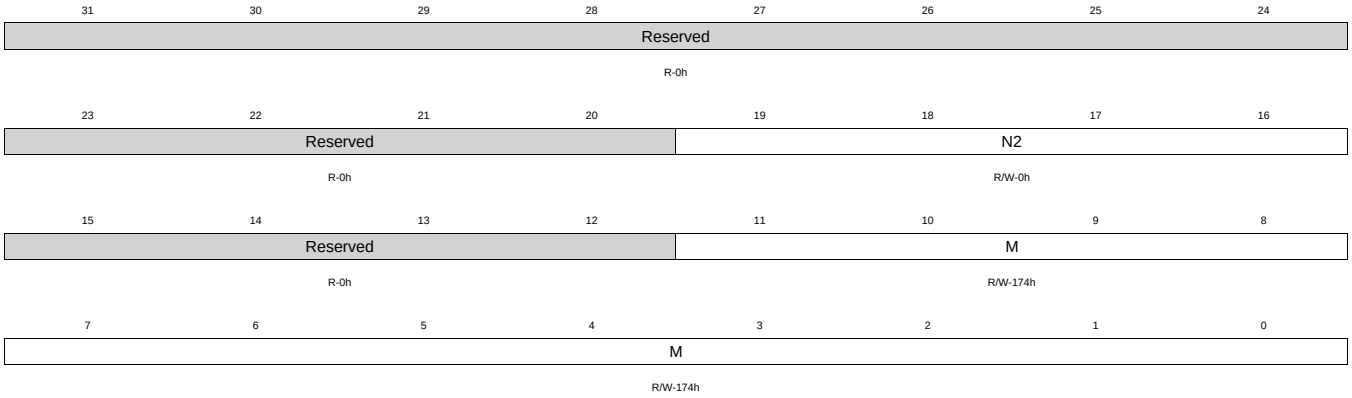

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.92 VIDEO1PLL_MN2DIV Register (offset = 1E4h) [reset = 174h]

VIDEO1PLL_MN2DIV is shown in Figure 2-118 and described in Table 2-134.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-118. VIDEO1PLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-134. VIDEO1PLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



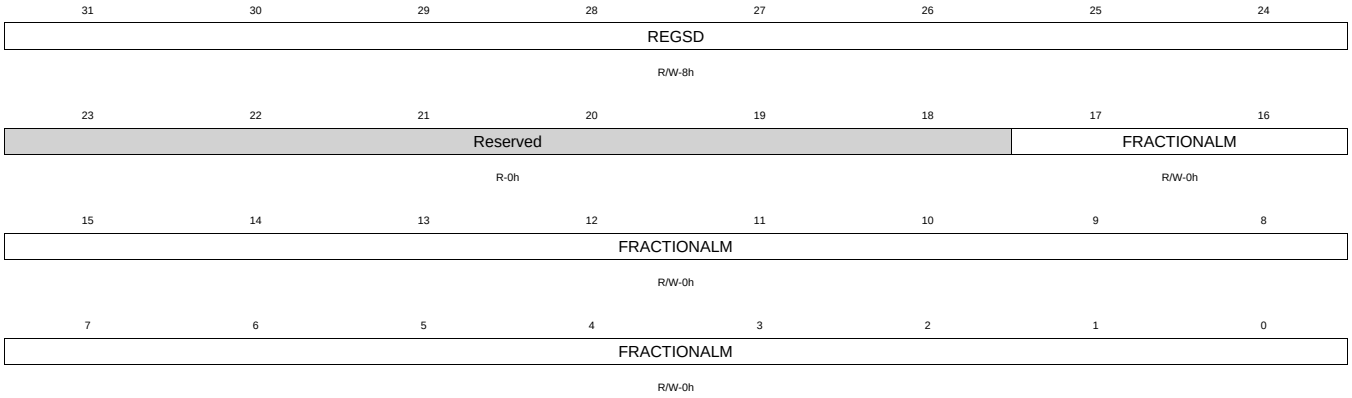
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.93 VIDEO1PLL_FRACDIV Register (offset = 1E8h) [reset = 8000000h]

VIDEO1PLL_FRACDIV is shown in Figure 2-119 and described in Table 2-135.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-119. VIDEO1PLL_FRACDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-135. VIDEO1PLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.94 VIDEO1PLL_BWCTRL Register (offset = 1ECh) [reset = 0h]

VIDEO1PLL_BWCTRL is shown in Figure 2-120 and described in Table 2-136.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-120. VIDEO1PLL_BWCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | BWCONTROL | | BW_INCR_DECRZ |
| R-0h | | | | | R/W-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-136. VIDEO1PLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0 : decrease BW<br>0x1 : increase BW |



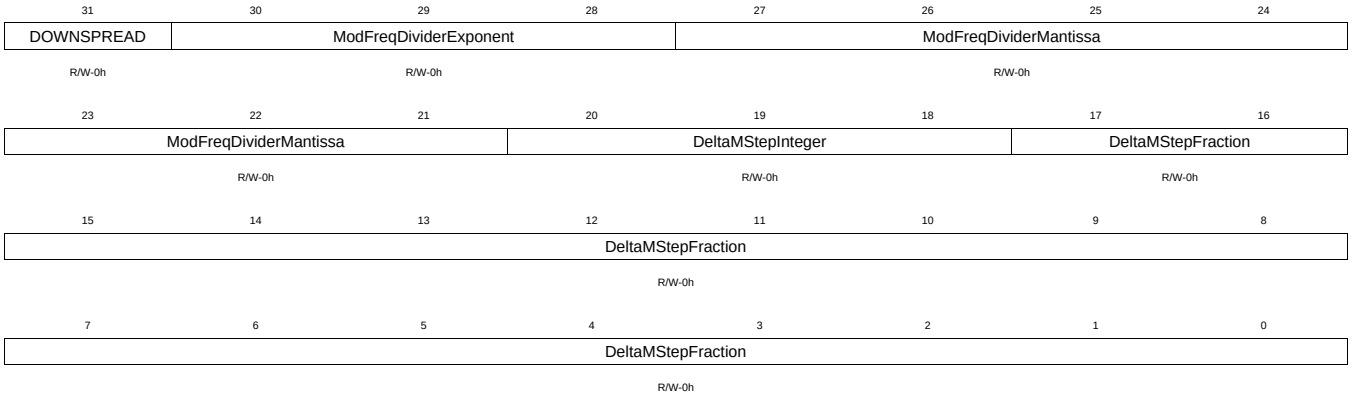
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.95 VIDEO1PLL_FRACCTRL Register (offset = 1F0h) [reset = 0h]

VIDEO1PLL_FRACCTRL is shown in Figure 2-121 and described in Table 2-137.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-121. VIDEO1PLL_FRACCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-137. VIDEO1PLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.96 VIDEO1PLL_STATUS Register (offset = 1F4h) [reset = C0000121h]

VIDEO1PLL_STATUS is shown in Figure 2-122 and described in Table 2-138.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-122. VIDEO1PLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-138. VIDEO1PLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1/0 indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-138. VIDEO1PLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.97 HDMIPLL_PWRCTRL Register (offset = 200h) [reset = 30h]

HDMIPLL_PWRCTRL is shown in Figure 2-123 and described in Table 2-139.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-123. HDMIPLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-139. HDMIPLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.98 HDMIPLL_CLKCTRL Register (offset = 204h) [reset = 914824h]

HDMIPLL_CLKCTRL is shown in Figure 2-124 and described in Table 2-140.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-124. HDMIPLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-140. HDMIPLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-140. HDMIPLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 = Reserved<br>0x2 = HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 = Reserved<br>0x4 = HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 = Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |

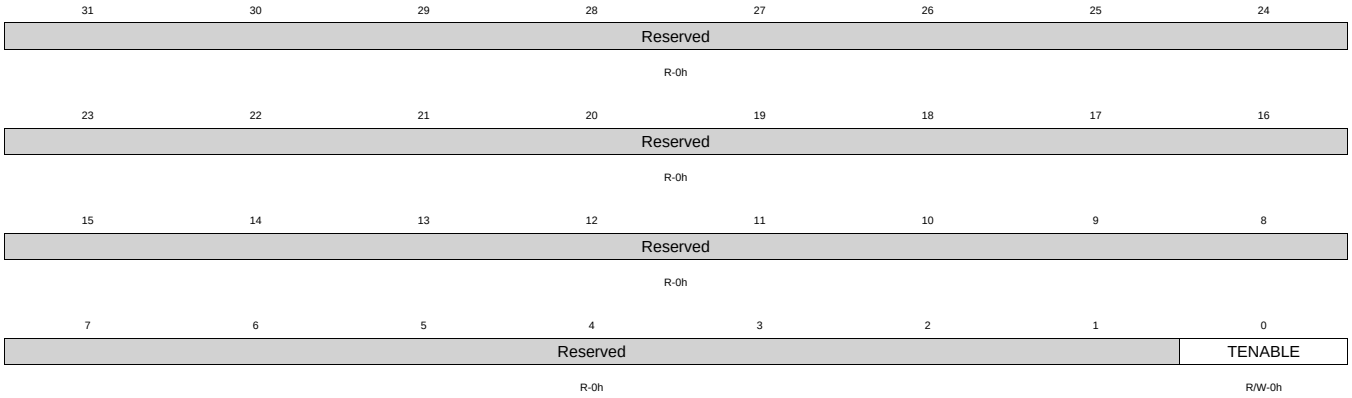

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.99 HDMIPLL_TENABLE Register (offset = 208h) [reset = 0h]

HDMIPLL_TENABLE is shown in Figure 2-125 and described in Table 2-141.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-125. HDMIPLL_TENABLE Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-141. HDMIPLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |

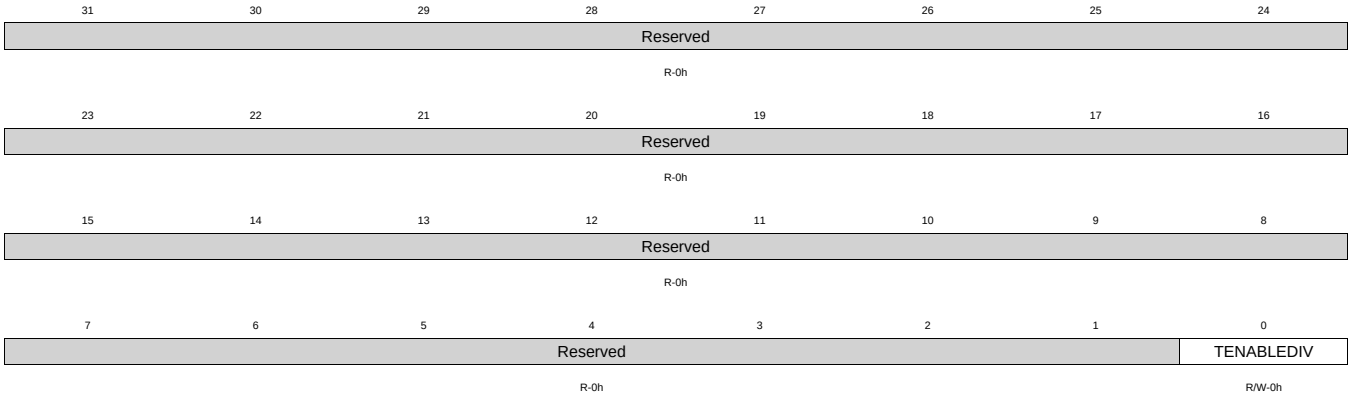

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.100 HDMIPLL_TENABLEDIV Register (offset = 20Ch) [reset = 0h]

HDMIPLL_TENABLEDIV is shown in Figure 2-126 and described in Table 2-142.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-126. HDMIPLL_TENABLEDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-142. HDMIPLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |



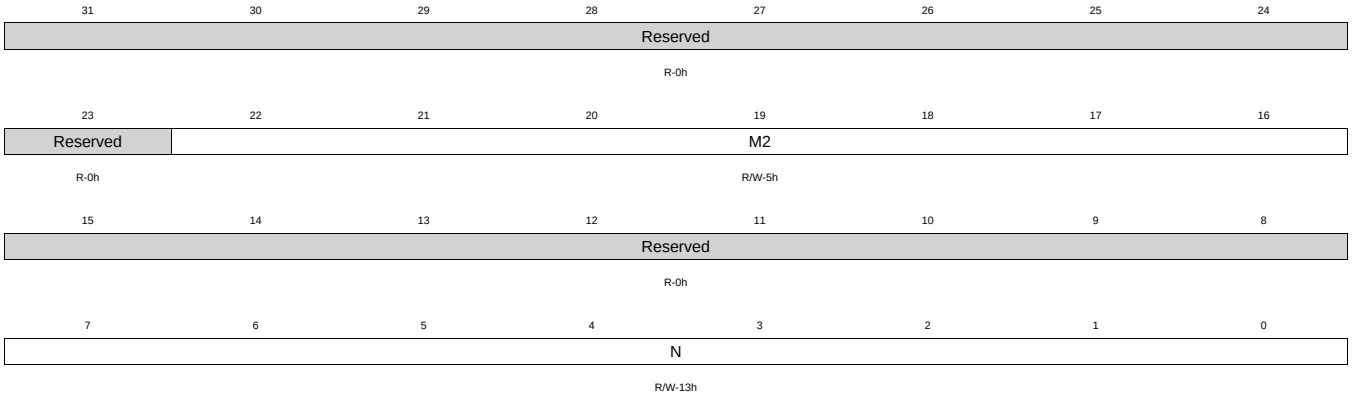
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.101 HDMIPLL_M2NDIV Register (offset = 210h) [reset = 50013h]

HDMIPLL_M2NDIV is shown in Figure 2-127 and described in Table 2-143.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-127. HDMIPLL_M2NDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-143. HDMIPLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.102 HDMIPLL_MN2DIV Register (offset = 214h) [reset = 174h]

HDMIPLL_MN2DIV is shown in Figure 2-128 and described in Table 2-144.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-128. HDMIPLL_MN2DIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | N2 | | | |
| R-0h | | | | R/W-0h | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | M | | | |
| R-0h | | | | R/W-174h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| M | | | | | | | |
| R/W-174h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-144. HDMIPLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.103 HDMIPLL_FRACDIV Register (offset = 218h) [reset = 8000000h]

HDMIPLL_FRACDIV is shown in Figure 2-129 and described in Table 2-145.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-129. HDMIPLL_FRACDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REGSD | | | | | | | |
| R/W-8h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | FRACTIONALM | |
| R-0h | | | | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-145. HDMIPLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.104 HDMIPLL_BWCTRL Register (offset = 21Ch) [reset = 0h]

HDMIPLL_BWCTRL is shown in Figure 2-130 and described in Table 2-146.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-130. HDMIPLL_BWCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | BWCONTROL | | BW_INCR_DECRZ |
| R-0h | | | | | R/W-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-146. HDMIPLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0 : decrease BW<br>0x1 : increase BW |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.105 HDMIPLL_FRACCTRL Register (offset = 220h) [reset = 0h]

HDMIPLL_FRACCTRL is shown in Figure 2-131 and described in Table 2-147.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-131. HDMIPLL_FRACCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOWNSPREAD | ModFreqDividerExponent | | | ModFreqDividerMantissa | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ModFreqDividerMantissa | | | DeltaMStepInteger | | | DeltaMStepFraction | |
| R/W-0h | | | R/W-0h | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-147. HDMIPLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.106 HDMIPLL_STATUS Register (offset = 224h) [reset = C0000121h]

HDMIPLL_STATUS is shown in Figure 2-132 and described in Table 2-148.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-132. HDMIPLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-148. HDMIPLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1/0 indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-148. HDMIPLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.107 AUDIOPLL_PWRCTRL Register (offset = 230h) [reset = 30h]

AUDIOPLL_PWRCTRL is shown in Figure 2-133 and described in Table 2-149.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-133. AUDIOPLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-149. AUDIOPLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.108 AUDIOPLL_CLKCTRL Register (offset = 234h) [reset = 914824h]

AUDIOPLL_CLKCTRL is shown in Figure 2-134 and described in Table 2-150.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-134. AUDIOPLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-150. AUDIOPLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-150. AUDIOPLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 : Reserved<br>0x2 : HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 : Reserved<br>0x4 : HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 : Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |

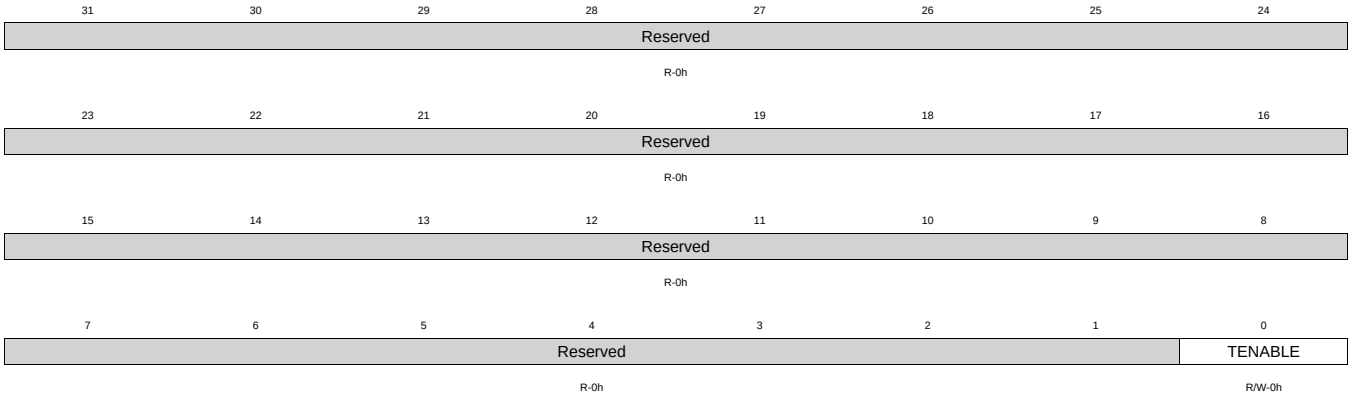

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.109 AUDIOPLL_TENABLE Register (offset = 238h) [reset = 0h]

AUDIOPLL_TENABLE is shown in Figure 2-135 and described in Table 2-151.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-135. AUDIOPLL_TENABLE Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-151. AUDIOPLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |

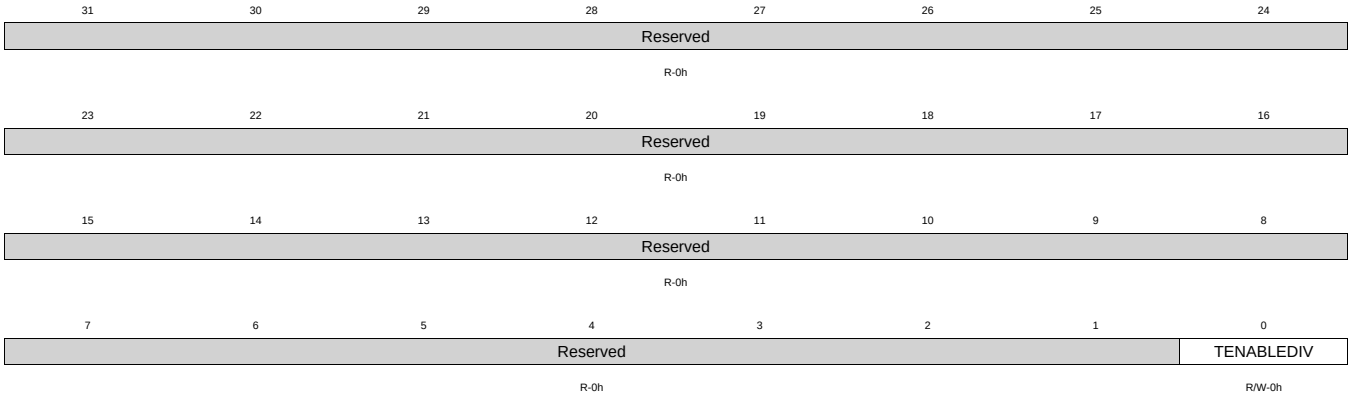

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.110 AUDIOPLL_TENABLEDIV Register (offset = 23Ch) [reset = 0h]

AUDIOPLL_TENABLEDIV is shown in Figure 2-136 and described in Table 2-152.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-136. AUDIOPLL_TENABLEDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-152. AUDIOPLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |

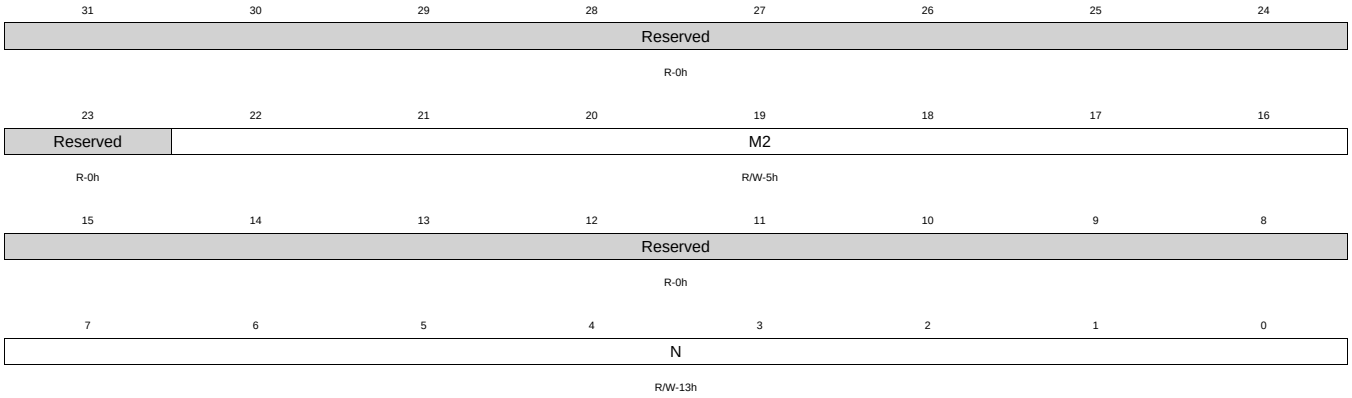

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.111 AUDIOPLL_M2NDIV Register (offset = 240h) [reset = 50013h]

AUDIOPLL_M2NDIV is shown in Figure 2-137 and described in Table 2-153.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-137. AUDIOPLL_M2NDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-153. AUDIOPLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |

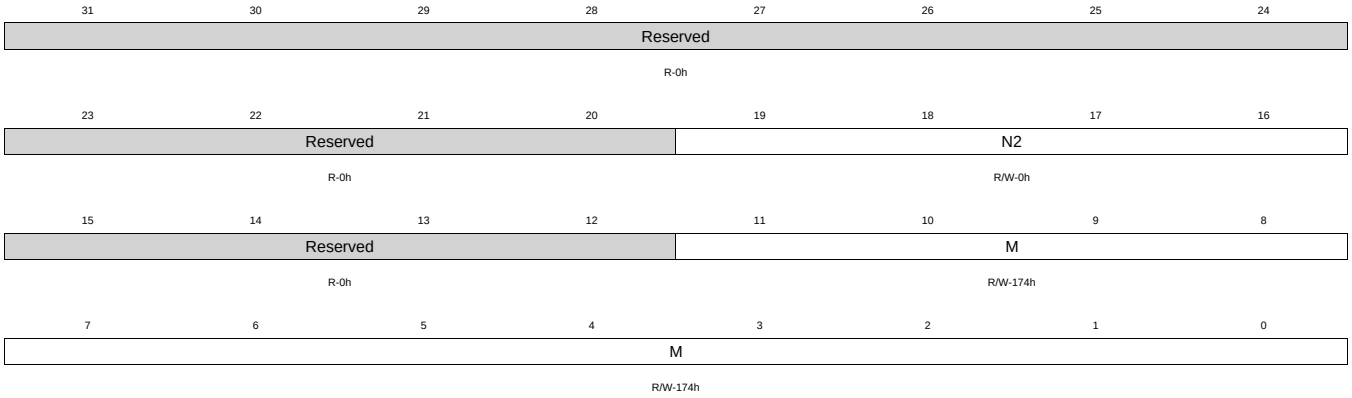

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.112 AUDIOPLL_MN2DIV Register (offset = 244h) [reset = 174h]

AUDIOPLL_MN2DIV is shown in Figure 2-138 and described in Table 2-154.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-138. AUDIOPLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-154. AUDIOPLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.113 AUDIOPLL_FRACDIV Register (offset = 248h) [reset = 8000000h]

AUDIOPLL_FRACDIV is shown in Figure 2-139 and described in Table 2-155.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-139. AUDIOPLL_FRACDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REGSD | | | | | | | |
| R/W-8h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | FRACTIONALM | |
| R-0h | | | | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-155. AUDIOPLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |

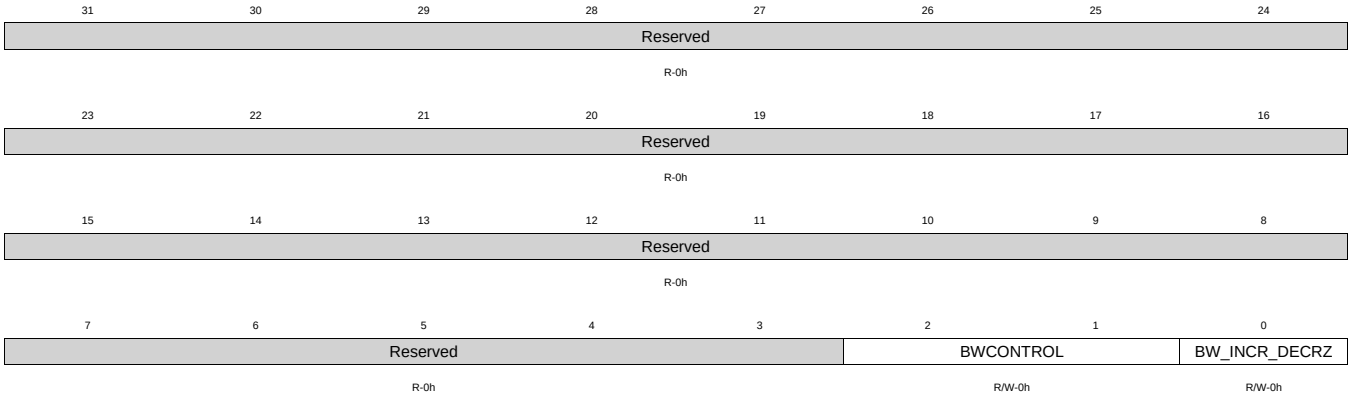

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.114 AUDIOPLL_BWCTRL Register (offset = 24Ch) [reset = 0h]

AUDIOPLL_BWCTRL is shown in Figure 2-140 and described in Table 2-156.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-140. AUDIOPLL_BWCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-156. AUDIOPLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0 : decrease BW<br>0x1 : increase BW |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.115 AUDIOPLL_FRACCTRL Register (offset = 250h) [reset = 0h]

AUDIOPLL_FRACCTRL is shown in Figure 2-141 and described in Table 2-157.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-141. AUDIOPLL_FRACCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOWNSPREAD | ModFreqDividerExponent | | | ModFreqDividerMantissa | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ModFreqDividerMantissa | | | DeltaMStepInteger | | | DeltaMStepFraction | |
| R/W-0h | | | R/W-0h | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-157. AUDIOPLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.116 AUDIOPLL_STATUS Register (offset = 254h) [reset = C0000121h]

AUDIOPLL_STATUS is shown in Figure 2-142 and described in Table 2-158.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-142. AUDIOPLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-158. AUDIOPLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1/0 indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-158. AUDIOPLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.117 USBPLL_PWRCTRL Register (offset = 260h) [reset = 30h]

USBPLL_PWRCTRL is shown in Figure 2-143 and described in Table 2-159.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-143. USBPLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-159. USBPLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.118 USBPLL_CLKCTRL Register (offset = 264h) [reset = 914824h]

USBPLL_CLKCTRL is shown in Figure 2-144 and described in Table 2-160.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-144. USBPLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-160. USBPLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-160. USBPLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 : Reserved<br>0x2 : HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 : Reserved<br>0x4 : HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 : Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |

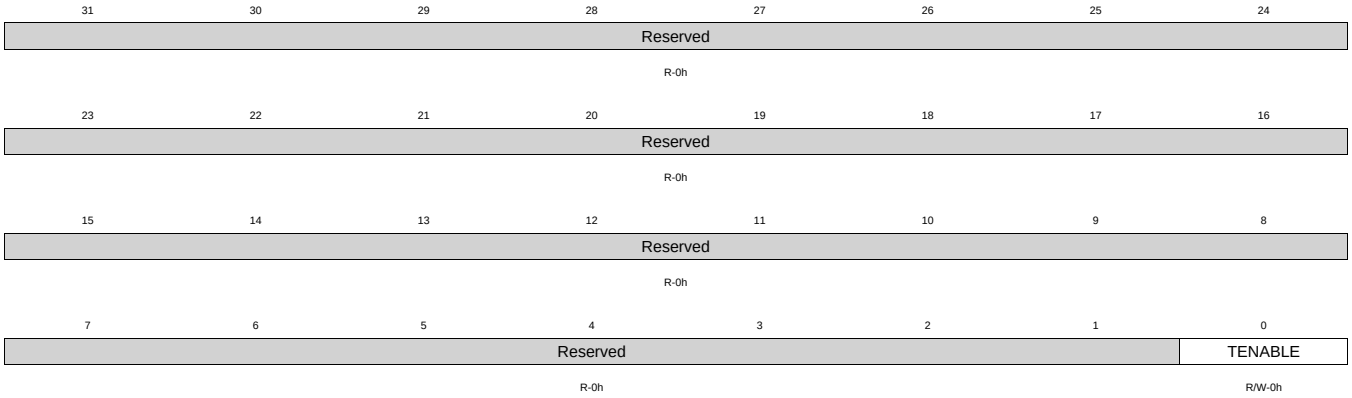

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.119 USBPLL_TENABLE Register (offset = 268h) [reset = 0h]

USBPLL_TENABLE is shown in Figure 2-145 and described in Table 2-161.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-145. USBPLL_TENABLE Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-161. USBPLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.120 USBPLL_TENABLEDIV Register (offset = 26Ch) [reset = 0h]

USBPLL_TENABLEDIV is shown in Figure 2-146 and described in Table 2-162.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-146. USBPLL_TENABLEDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | TENABLEDIV |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-162. USBPLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |



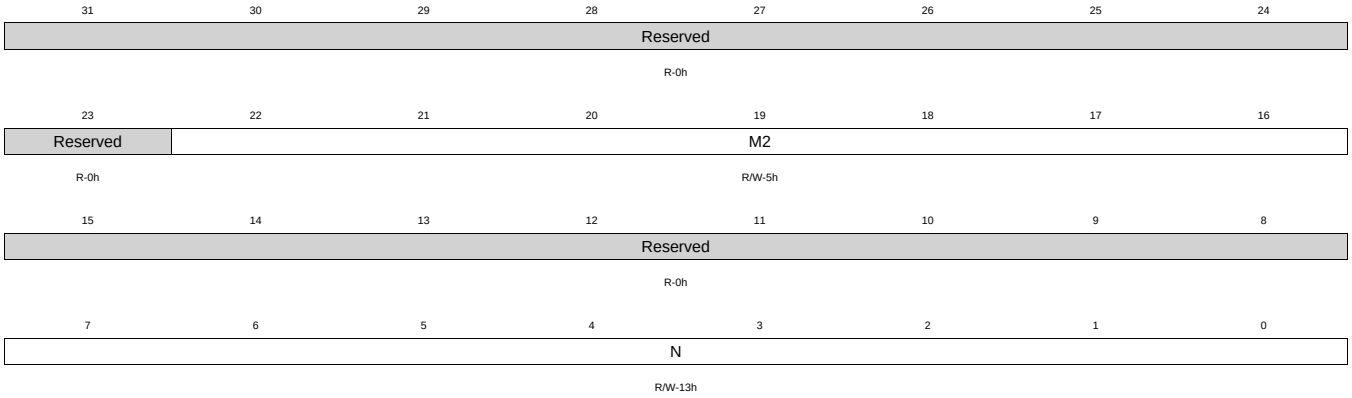
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.121 USBPLL_M2NDIV Register (offset = 270h) [reset = 50013h]

USBPLL_M2NDIV is shown in Figure 2-147 and described in Table 2-163.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-147. USBPLL_M2NDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-163. USBPLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |

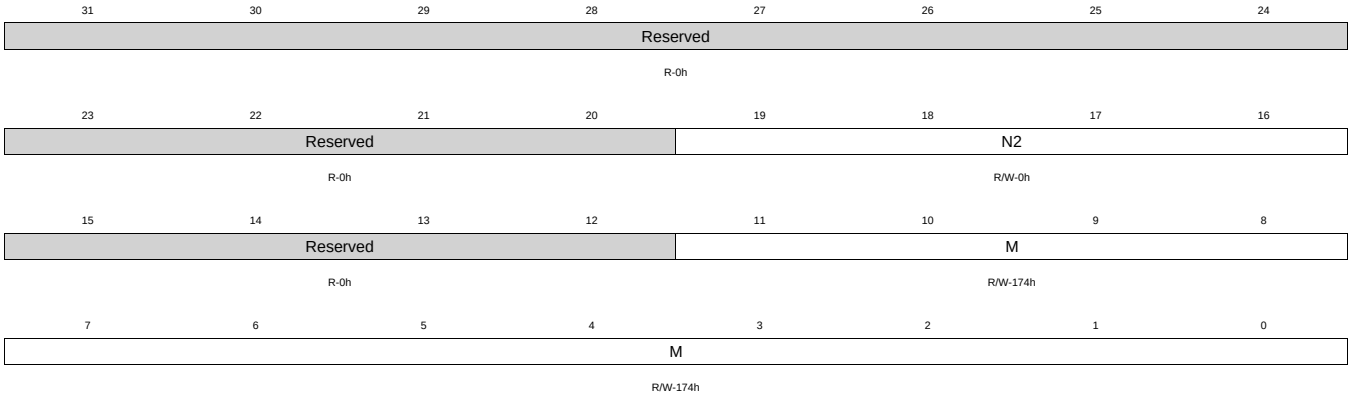

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.122 USBPLL_MN2DIV Register (offset = 274h) [reset = 174h]

USBPLL_MN2DIV is shown in Figure 2-148 and described in Table 2-164.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-148. USBPLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-164. USBPLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.123 USBPLL_FRACDIV Register (offset = 278h) [reset = 8000000h]

USBPLL_FRACDIV is shown in Figure 2-149 and described in Table 2-165.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-149. USBPLL_FRACDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REGSD | | | | | | | |
| R/W-8h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | FRACTIONALM | |
| R-0h | | | | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FRACTIONALM | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-165. USBPLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |

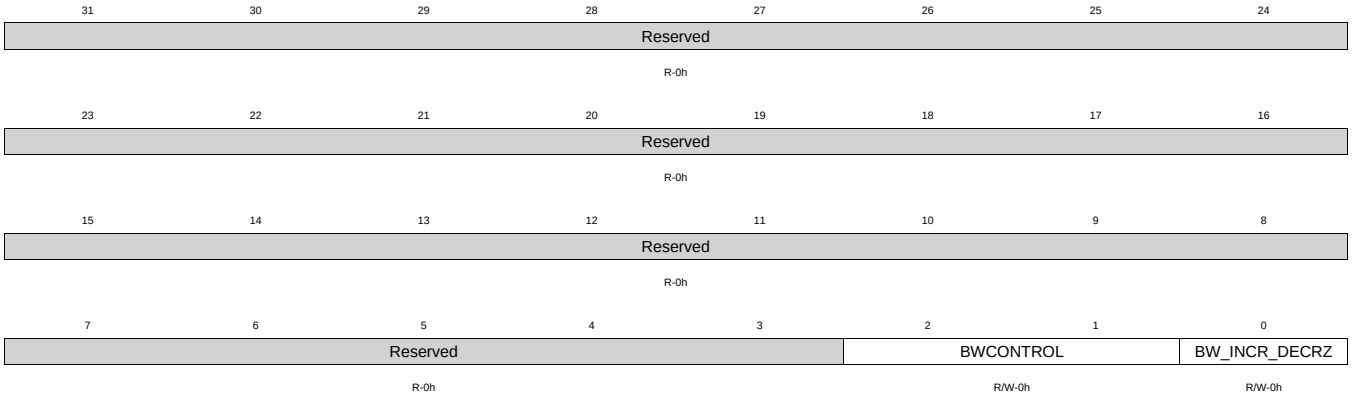

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.124 USBPLL_BWCTRL Register (offset = 27Ch) [reset = 0h]

USBPLL_BWCTRL is shown in Figure 2-150 and described in Table 2-166.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-150. USBPLL_BWCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-166. USBPLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0 = 0 : decrease BW<br>1 = 1 : increase BW |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.125 USBPLL_FRACCTRL Register (offset = 280h) [reset = 0h]

USBPLL_FRACCTRL is shown in Figure 2-151 and described in Table 2-167.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-151. USBPLL_FRACCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOWNSPREAD | ModFreqDividerExponent | | | ModFreqDividerMantissa | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ModFreqDividerMantissa | | | DeltaMStepInteger | | | DeltaMStepFraction | |
| R/W-0h | | | R/W-0h | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-167. USBPLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.126 USBPLL_STATUS Register (offset = 284h) [reset = C0000121h]

USBPLL_STATUS is shown in Figure 2-152 and described in Table 2-168.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-152. USBPLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-168. USBPLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1 /0indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-168. USBPLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.127 DDRPLL_PWRCTRL Register (offset = 290h) [reset = 30h]

DDRPLL_PWRCTRL is shown in Figure 2-153 and described in Table 2-169.

The PLL Power Control register is used to control the power state of therespective PLL.

**Figure 2-153. DDRPLL_PWRCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PONIN | PGOODIN | RET | ISORET | ISOSCAN | OFFMODE |
| R-0h | | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-169. DDRPLL_PWRCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | PONIN | R/W | 1h | ON/OFF control of the weak power switch digital. For functional mode it should be 1 |
| 4 | PGOODIN | R/W | 1h | ON/OFF control of the strong power switch digital. For functional mode it should be 1 |
| 3 | RET | R/W | 0h | Save/Restore control for Retention mode. For functional mode it should be 0 |
| 2 | ISORET | R/W | 0h | Save/Restore control for Isolation of output pins For functional mode it should be 0 |
| 1 | ISOSCAN | R/W | 0h | Save/Restore control for Isolation of the Scanout pins For functional mode it should be 0 |
| 0 | OFFMODE | R/W | 0h | Used to switch OFF the logic on VDDA. For functional mode it should be 0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.128 DDRPLL_CLKCTRL Register (offset = 294h) [reset = 914824h]

DDRPLL_CLKCTRL is shown in Figure 2-154 and described in Table 2-170.

The PLL Clock Control register is used to control the different clock control state of the respective PLL

**Figure 2-154. DDRPLL_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CYCLESLIPEN | ENSSC | CLKDCOLDOEN | NWELLTRIM | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IDLE | BYPASSACKZ | STBYRET | CLKOUTEN | CLKOUTLDOEN | ULOWCLKEN | CLKDCOLDOPWDNZ | M2PWDNZ |
| R/W-1h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h | R/W-0h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | STOPMODE | Reserved | SELFREQDCO | | | Reserved | RELAXED_LOCK |
| R-0h | R/W-1h | R-0h | R/W-2h | | | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | TINITZ |
| R-12h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-170. DDRPLL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CYCLESLIPEN | R/W | 0h | FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK. |
| 30 | ENSSC | R/W | 0h | Controls Clock Spreading<br>SSC is not supported. Should be set to 0x0 to disable clock spreading. |
| 29 | CLKDCOLDOEN | R/W | 0h | Synchronously enables/disables CLKDCOLDO<br>0x0 : synchronously disables CLKDCOLDO<br>0x1 : synchronously enables CLKDCOLDO |
| 28-24 | NWELLTRIM | R/W | 0h | Trim values for the PLL |
| 23 | IDLE | R/W | 1h | Sets PLL to Idle mode<br>0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked<br>0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power |
| 22 | BYPASSACKZ | R/W | 0h | BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module. |
| 21 | STBYRET | R/W | 0h | Standby retention control<br>0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks.<br>0x1 : prepares ADPLLLJ for retention by gating all the internal clocks. |
| 20 | CLKOUTEN | R/W | 1h | CLKOUT enable or disable<br>0x0 : synchronously disables CLKOUT<br>0x1 : synchronously enables CLKOUT |
| 19 | CLKOUTLDOEN | R | 0h | Synchronously enables/disables CLKOUTLDO<br>0x0 : synchronously disables CLKOUTLDO<br>0x1 : synchronously enables CLKOUTLDO |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-170. DDRPLL_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | ULOWCLKEN | R/W | 0h | Select CLKOUT source in bypass<br>0x0 : When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1)<br>0x1 : When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW. |
| 17 | CLKDCOLDOPWDNZ | R/W | 0h | 0 Asynchronous power down for CLKDCOLDO o/p. |
| 16 | M2PWDNZ | R/W | 1h | M2 divider power down mode<br>0x0 : Asynchronous power down for M2 divider<br>0x1 : M2 divider is functional |
| 15 | Reserved | R | 0h | |
| 14 | STOPMODE | R/W | 1h | When in Lossclk/Stbyret<br>0x0 : Limp mode<br>0x1 : Stopmode |
| 13 | Reserved | R | 0h | |
| 12-10 | SELFREQDCO | R/W | 2h | DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector.<br>0x0 : Reserved<br>0x2 : HS2 : DCOCLK range is from 500 MHz to 1000 MHz<br>0x3 : Reserved<br>0x4 : HS1 : DCOCLK range is from 1000 MHz to 2000 MHz<br>0x5 : Reserved |
| 9 | Reserved | R | 0h | |
| 8 | RELAXED_LOCK | R/W | 0h | Decides when FREQLOCK asserted<br>0x0 : FREQLOCK asserted when DC frequency error less than 1%<br>0x1 : FREQLOCK asserted when DC frequency error less than 2% |
| 7-1 | Reserved | R | 12h | |
| 0 | TINITZ | R/W | 0h | PLL core soft reset |

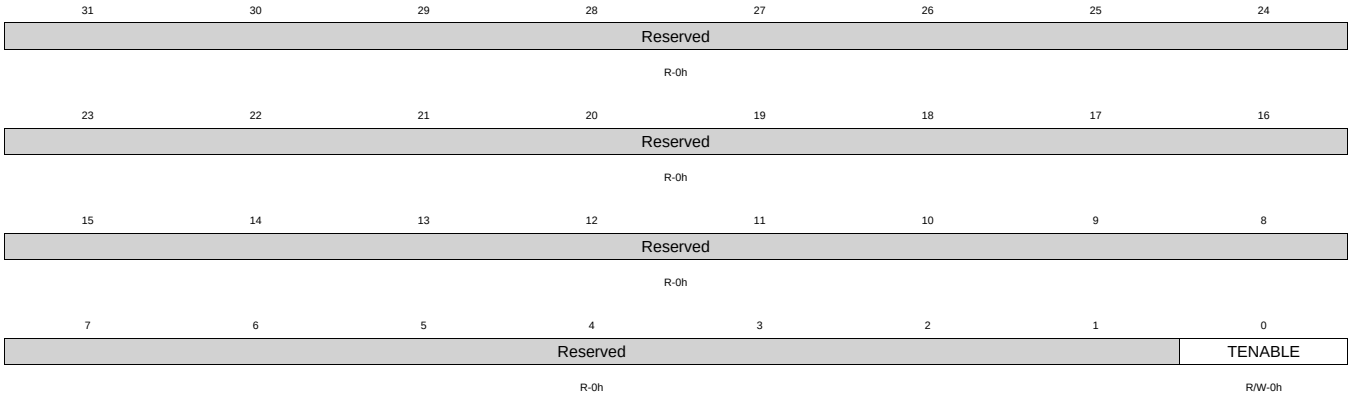

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.129 DDRPLL_TENABLE Register (offset = 298h) [reset = 0h]

DDRPLL_TENABLE is shown in Figure 2-155 and described in Table 2-171.

Load the M, N, SD and SELFREQDCO dividers of the particular ADPLLLJ.

**Figure 2-155. DDRPLL_TENABLE Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-171. DDRPLL_TENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLE | R/W | 0h | M, N. SD and SELFREQDCO latch (active rise edge) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.130 DDRPLL_TENABLEDIV Register (offset = 29Ch) [reset = 0h]

DDRPLL_TENABLEDIV is shown in Figure 2-156 and described in Table 2-172.

The PLL TENABLEDIV register is used to load the M2,N2 dividers of respective PLL

**Figure 2-156. DDRPLL_TENABLEDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | TENABLEDIV |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-172. DDRPLL_TENABLEDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TENABLEDIV | R/W | 0h | M2 and N2 latch (active rise edge) |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.131 DDRPLL_M2NDIV Register (offset = 2A0h) [reset = 50013h]

DDRPLL_M2NDIV is shown in Figure 2-157 and described in Table 2-173.

PLL M2NDIV register is for programming M2 and N values of respective PLL

**Figure 2-157. DDRPLL_M2NDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | M2 | | | | | | |
| R-0h | R/W-5h | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| N | | | | | | | |
| R/W-13h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-173. DDRPLL_M2NDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-16 | M2 | R/W | 5h | Post-divider is REGM2 |
| 15-8 | Reserved | R | 0h | |
| 7-0 | N | R/W | 13h | Pre-divider is REGN+1 |

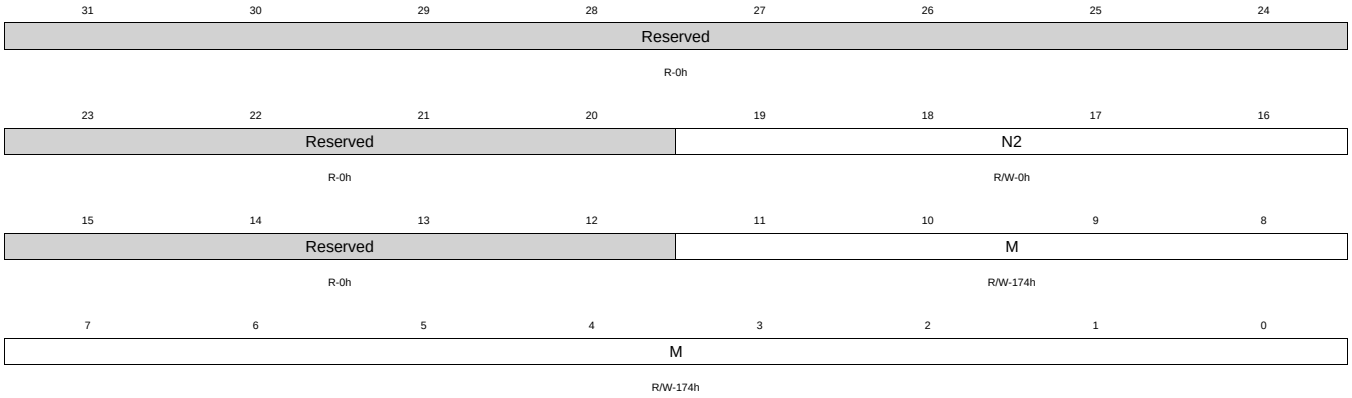

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.132 DDRPLL_MN2DIV Register (offset = 2A4h) [reset = 174h]

DDRPLL_MN2DIV is shown in Figure 2-158 and described in Table 2-174.

PLL MN2DIV register is for programming M and N2 values of respective PLL

**Figure 2-158. DDRPLL_MN2DIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-174. DDRPLL_MN2DIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | N2 | R/W | 0h | Bypass divider is REGN2+1 |
| 15-12 | Reserved | R | 0h | |
| 11-0 | M | R/W | 174h | Feedback multiplier is REGM |



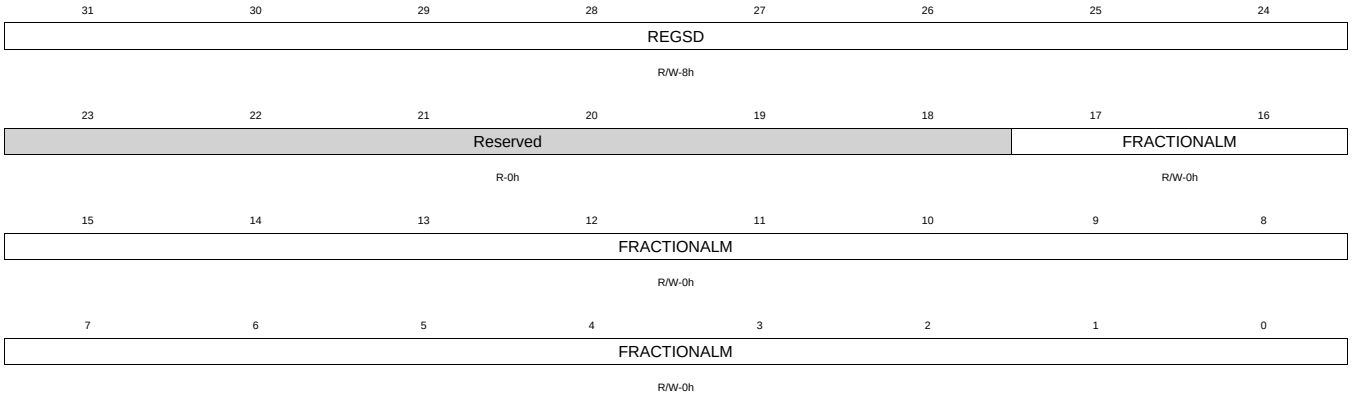
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.133 DDRPLL_FRACDIV Register (offset = 2A8h) [reset = 8000000h]

DDRPLL_FRACDIV is shown in Figure 2-159 and described in Table 2-175.

The PLL FRACDIV register is for controlling the fractional values of respective PLL

**Figure 2-159. DDRPLL_FRACDIV Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-175. DDRPLL_FRACDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REGSD | R/W | 8h | Sigma-Delta Divider<br>Should be set by s/w to provide optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. |
| 23-18 | Reserved | R | 0h | |
| 17-0 | FRACTIONALM | R/W | 0h | Fractional part of the M divider. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.134 DDRPLL_BWCTRL Register (offset = 2ACh) [reset = 0h]

DDRPLL_BWCTRL is shown in Figure 2-160 and described in Table 2-176.

The PLL_BWCTRL register is for controlling the loop bandwidth of respective PLL

**Figure 2-160. DDRPLL_BWCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | BWCONTROL | | BW_INCR_DECRZ |
| R-0h | | | | | R/W-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-176. DDRPLL_BWCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-1 | BWCONTROL | R/W | 0h | Change Loop Bandwidth |
| 0 | BW_INCR_DECRZ | R/W | 0h | Direction of Loop Bandwidth<br>0x0 : decrease BW<br>0x1 : increase BW |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.135 DDRPLL_FRACCTRL Register (offset = 2B0h) [reset = 0h]

DDRPLL_FRACCTRL is shown in Figure 2-161 and described in Table 2-177.

Controlls the fractional portion of respective ADPLLLJ.

**Figure 2-161. DDRPLL_FRACCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOWNSPREAD | ModFreqDividerExponent | | | ModFreqDividerMantissa | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ModFreqDividerMantissa | | | DeltaMStepInteger | | | DeltaMStepFraction | |
| R/W-0h | | | R/W-0h | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DeltaMStepFraction | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-177. DDRPLL_FRACCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOWNSPREAD | R/W | 0h | Controls frequency spread<br>0x0 : enables both side frequency spread about the programmed frequency.<br>0x1 : enables low frequency spread only |
| 30-28 | ModFreqDividerExponent | R/W | 0h | Exponent of the REFCLK divider to define the modulation frequency. |
| 27-21 | ModFreqDividerMantissa | R/W | 0h | Mantissa of the REFCLK divider to define the modulation frequency |
| 20-18 | DeltaMStepInteger | R/W | 0h | Integer part of Frequency Spread control. |
| 17-0 | DeltaMStepFraction | R/W | 0h | The fraction part of Frequency Spread control |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.136 DDRPLL_STATUS Register (offset = 2B4h) [reset = C0000121h]

DDRPLL_STATUS is shown in Figure 2-162 and described in Table 2-178.

The PLL Status register is for viewing the status of the respective PLL.

**Figure 2-162. DDRPLL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PONOUT | PGOODOUT | LDOPWDN | RECAL_BSTATUS3 | RECAL_OPPIN | Reserved | | |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | PHASELOCK | FREQLOCK | BYPASSACK |
| R-0h | | | | | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STBYRETACK | LOSSREF | CLKOUTENACK | LOCK2 | M2CHANGEACK | SSCACK | HIGHJITTER | BYPASS |
| R-0h | R-0h | R-1h | R-0h | R-0h | R-0h | R-0h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-178. DDRPLL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PONOUT | R | 1h | status of the weak power-switch<br>0x0 : indicates the/OFF status of the weak power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the weak power-switch in digital to SOC. |
| 30 | PGOODOUT | R | 1h | status of the strong power-switch<br>0x0 : indicates the/OFF status of the strong power-switch in digital to SOC.<br>0x1 : ndicates the ON status of the strong power-switch in digital to SOC. |
| 29 | LDOPWDN | R | 0h | 1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition. |
| 28 | RECAL_BSTATUS3 | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 27 | RECAL_OPPIN | R | 0h | Recalibration status flag. 1 ADPLLLJ requires recalibration) |
| 26-11 | Reserved | R | 0h | |
| 10 | PHASELOCK | R | 0h | Status on PHASELOCK output pin |
| 9 | FREQLOCK | R | 0h | Status on FREQLOCK output pin |
| 8 | BYPASSACK | R | 1h | Status of BYPASSACK output pin |
| 7 | STBYRETACK | R | 0h | Standby and retention status<br>0x0 : indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process.<br>0x1 : indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention. |
| 6 | LOSSREF | R | 0h | Reference input loss |
| 5 | CLKOUTENACK | R | 1h | 1 /0indicates enable/disable condition of CLKOUTEN |
| 4 | LOCK2 | R | 0h | ADPLL internal loop lock status |
| 3 | M2CHANGEACK | R | 0h | acknowledge for M2 change |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-178. DDRPLL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | SSCACK | R | 0h | Spread Spectrum status<br>0x0 : Spread-spectrum Clocking is disabled on output clocks<br>0x1 : Spread-spectrum Clocking is enabled on output clocks |
| 1 | HIGHJITTER | R | 0h | 1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%. |
| 0 | BYPASS | R | 1h | Bypass status signal. 1 CLKOUT in bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.137 OSC_SRC Register (offset = 2C0h) [reset = 0h]

OSC_SRC is shown in Figure 2-163 and described in Table 2-179.

The following groups of PLL needs a selection of clock source of OSC0 or OSC1 o DSPPLL/SGXPLL/HDVICPPLL/ISPPLL/DSSPLL o VIDEO0PLL o VIDEO1PLL o HDMIPLL o AUDIOPLL o For TEST reasons, the following additional PLLs also have OSC0 vs OC1 selection: GEM, SGX, HDVICP, ISP, DSS, USB, DDR

**Figure 2-163. OSC_SRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GEM_PLL_SOURCE | SGX_PLL_SOURCE | HDVICP_PLL_SOURCE | ISP_PLL_SOURCE | DSS_PLL_SOURCE | USB_PLL_SOURCE | DDR_PLL_SOURCE | AUDIO_PLL_SOURCE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | HDMI_PLL_SOURCE | VIDEO1_PLL_SOURCE | VIDEO0_PLL_SOURCE |
| R-0h | | | | | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | L3_PLL_SOURCE |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-179. OSC_SRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | GEM_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the GEM ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 30 | SGX_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the SGX ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 29 | HDVICP_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the HDVICP ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 28 | ISP_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the ISP ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 27 | DSS_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the DSS ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 26 | USB_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the USB ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 25 | DDR_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the DDR ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 24 | AUDIO_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the AUDIO ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 23-19 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-179. OSC_SRC Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | HDMI_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the HDMI ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 17 | VIDEO1_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the VIDEO1 ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 16 | VIDEO0_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the VIDEO0 ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |
| 15-1 | Reserved | R | 0h | |
| 0 | L3_PLL_SOURCE | R/W | 0h | Select the source Oscillator of the L3 ADPLLLJ<br>0x0 : Source is OSC0<br>0x1 : Source is OSC1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.138 MPU_CLKSRC Register (offset = 2C4h) [reset = 0h]

MPU_CLKSRC is shown in Figure 2-164 and described in Table 2-180.

Clock source for ARM PLL

**Figure 2-164. MPU_CLKSRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | MPU_SOURCE |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-180. MPU_CLKSRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | MPU_SOURCE | R/W | 0h | Select the source clock of the ARM PLL<br>0x0 : Source is OSC0<br>0x1 : Source is RTC DIVIDER OUTPUT |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.139 VIDEO_PLL_CLKSRC Register (offset = 2C8h) [reset = 0h]

VIDEO_PLL_CLKSRC is shown in Figure 2-165 and described in Table 2-181. Select the clock source for VIDEO PLLs, VENC_G_CLK and audio PLL CLK2 input of PRCM.

**Figure 2-165. VIDEO_PLL_CLKSRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | HD_VENC_G_CLK_SOURCE |
| R-0h | | | | | | | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | TPPSSSTSO_MUX_SOURCE | | Reserved | |
| R-0h | | | | R/W-0h | | R-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIDEO_PLL_OUT_MUX_SOURCE | |
| R-0h | | | | | | R/W-0h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | VIDEO_PLL_CLK2_SOURCE |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-181. VIDEO_PLL_CLKSRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | Reserved | R | 0h | |
| 24 | HD_VENC_G_CLK_SOURCE | R/W | 0h | hd_venc_g_clk selection of DSS<br>0x0 : VIDEO1_PLL_OUT<br>0x1 : VIDEO_PLL_CLK2 from videopll_clk2_mux |
| 23-20 | Reserved | R | 0h | |
| 19-18 | TPPSSSTSO_MUX_SOURCE | R/W | 0h | Select the source of Audio PLL CLK2 input of PRCM<br>0x0 = 00 : Source is ISP_PLL_OUT/2<br>0x1 = 01 : Source is ISP_PLL_OUT<br>0x2 = 10 : Reserved<br>0x3 = 11 : Reserved |
| 17-10 | Reserved | R | 0h | |
| 9-8 | VIDEO_PLL_OUT_MUX_SOURCE | R/W | 0h | Select the source of VIDEO_PLL_OUT mux<br>0x0 = 00 : Source is VIDEO0_PLL_OUT<br>0x1 = 01 : Source is HDMI_PLL_OUT<br>0x2 = 10 : Source is VIDEO1_PLL_OUT<br>0x3 = 11 |
| 7-1 | Reserved | R | 0h | |
| 0 | VIDEO_PLL_CLK2_SOURCE | R/W | 0h | Select the source of VIDEO_PLL_CLK2<br>0x0 : Source is VIDEO_M_PCLK<br>0x1 : Source is HDMI PLL CLKOUT |

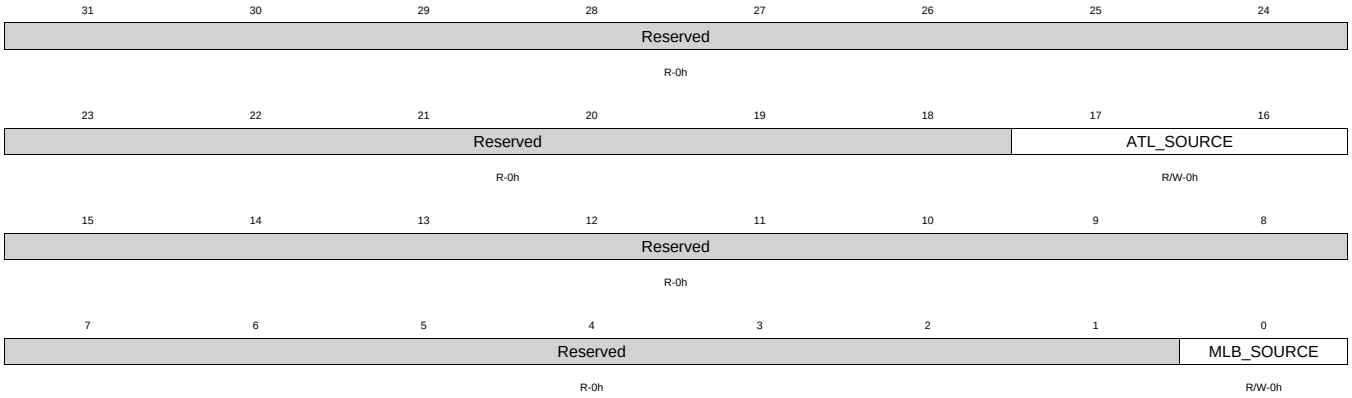

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.140 MLB_ATL_CLKSRC Register (offset = 2CCh) [reset = 0h]

MLB_ATL_CLKSRC is shown in Figure 2-166 and described in Table 2-182.

MLB and ATL clock source

**Figure 2-166. MLB_ATL_CLKSRC Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-182. MLB_ATL_CLKSRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | ATL_SOURCE | R/W | 0h | Select the source clock of the ATL<br>0x0 = 00 : Source is SYSCLK19<br>0x1 = 01 : Source is AUDIO_PLL_OUT<br>0x2 = 10 : Source is VIDEO_PLL_OUT_MUX<br>0x3 = 11 : Reserved |
| 15-1 | Reserved | R | 0h | |
| 0 | MLB_SOURCE | R/W | 0h | Select the source clock of the MLB<br>0x0 : Source is MLB_CLK_INPUT pin<br>0x1 : Source is MLBP_CLK_INPUT pin |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.141 McASP345_AUX_CLKSRC Register (offset = 2D0h) [reset = 0h]

McASP345_AUX_CLKSRC is shown in Figure 2-167 and described in Table 2-183.

Select the AUX clock source for MCASP3/4/5 modules.

**Figure 2-167. McASP345_AUX_CLKSRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | McASP5_AUX_SOURCE | | |
| R-0h | | | | | R/W-0h | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | McASP4_AUX_SOURCE | | |
| R-0h | | | | | R/W-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | McASP3_AUX_SOURCE | | |
| R-0h | | | | | R/W-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-183. McASP345_AUX_CLKSRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18-16 | McASP5_AUX_SOURCE | R/W | 0h | Select the source clock of the MCASP5 AUX clock<br>0x0 = 000 : Source is OUTPUT OF MLB MUX<br>0x1 = 001 : Source is AUDIO_PLL_OUT<br>0x2 = 010 : Source is is VIDEO_PLL_OUT_MUX<br>0x3 = 011 : Source is AUD_CLKIN0<br>0x4 = 100 : Source is AUD_CLKIN1<br>0x5 = 101 : Source is AUD_CLKIN2<br>0x6 = 110 : Source is AUXOSC_XI<br>0x7 = 111 : Reserved |
| 15-11 | Reserved | R | 0h | |
| 10-8 | McASP4_AUX_SOURCE | R/W | 0h | Select the source clock of the MCASP4 AUX clock<br>0x0 = 000 : Source is OUTPUT OF MLB MUX<br>0x1 = 001 : Source is AUDIO_PLL_OUT<br>0x2 = 010 : Source is is VIDEO_PLL_OUT_MUX<br>0x3 = 011 : Source is AUD_CLKIN0<br>0x4 = 100 : Source is AUD_CLKIN1<br>0x5 = 101 : Source is AUD_CLKIN2<br>0x6 = 110 : Source is AUXOSC_XI<br>0x7 = 111 : Reserved |
| 7-3 | Reserved | R | 0h | |
| 2-0 | McASP3_AUX_SOURCE | R/W | 0h | Select the source clock of the MCASP3 AUX clock<br>0x0 = 000 : Source is OUTPUT OF MLB MUX<br>0x1 = 001 : Source is AUDIO_PLL_OUT<br>0x2 = 010 : Source is is VIDEO_PLL_OUT_MUX<br>0x3 = 011 : Source is AUD_CLKIN0<br>0x4 = 100 : Source is AUD_CLKIN1<br>0x5 = 101 : Source is AUD_CLKIN2<br>0x6 = 110 : Source is AUXOSC_XI<br>0x7 = 111 : Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.142 McASP_AHCLK_CLKSRC Register (offset = 2D4h) [reset = 0h]

McASP_AHCLK_CLKSRC is shown in Figure 2-168 and described in Table 2-184.

Selects the source of McASP AHCLKX/R

**Figure 2-168. McASP_AHCLK_CLKSRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | McASP5_AHCLKX_SOURCE | | | McASP4_AHCLKX_SOURCE |
| R-0h | | | | R/W-0h | | | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| McASP4_AHCLKX_SOURCE | | McASP3_AHCLKX_SOURCE | | | McASP2_AHCLKX_SOURCE | | |
| R/W-0h | | R/W-0h | | | R/W-0h | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | McASP1_AHCLKR_SOURCE | | | McASP1_AHCLKX_SOURCE |
| R-0h | | | | R/W-0h | | | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| McASP1_AHCLKX_SOURCE | | McASP0_AHCLKR_SOURCE | | | McASP0_AHCLKX_SOURCE | | |
| R/W-0h | | R/W-0h | | | R/W-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-184. McASP_AHCLK_CLKSRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | |
| 27-25 | McASP5_AHCLKX_SOURCE | R/W | 0h | Select the source clock of the MCASP5 AH clock transmit<br>0x0 = 000 : Source is AUD_CLKIN0<br>0x1 = 001 : Source is AUD_CLKIN1<br>0x2 = 010 : Source is AUD_CLKIN2<br>0x3 = 011 : Source is AUXOSC_XI<br>0x4 = 100 : Source is ATCLK0<br>0x5 = 101 : Source is ATCLK1<br>0x6 = 110 : Source is ATCLK2<br>0x7 = 111 : Source is ATCLK3 |
| 24-22 | McASP4_AHCLKX_SOURCE | R/W | 0h | Select the source clock of the MCASP4 AH clock transmit<br>0x0 = 000 : Source is AUD_CLKIN0<br>0x1 = 001 : Source is AUD_CLKIN1<br>0x2 = 010 : Source is AUD_CLKIN2<br>0x3 = 011 : Source is AUXOSC_XI<br>0x4 = 100 : Source is ATCLK0<br>0x5 = 101 : Source is ATCLK1<br>0x6 = 110 : Source is ATCLK2<br>0x7 = 111 : Source is ATCLK3 |
| 21-19 | McASP3_AHCLKX_SOURCE | R/W | 0h | Select the source clock of the MCASP3 AH clock transmit<br>0x0 = 000 : Source is AUD_CLKIN0<br>0x1 = 001 : Source is AUD_CLKIN1<br>0x2 = 010 : Source is AUD_CLKIN2<br>0x3 = 011 : Source is AUXOSC_XI<br>0x4 = 100 : Source is ATCLK0<br>0x5 = 101 : Source is ATCLK1<br>0x6 = 110 : Source is ATCLK2<br>0x7 = 111 : Source is ATCLK3 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-184. McASP_AHCLK_CLKSRC Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18-16 | McASP2_AHCLKX_SOURCE | R/W | 0h | Select the source clock of the MCASP2 AH clock transmit<br>0x0 = 000 : Source is AUD_CLKIN0<br>0x1 = 001 : Source is AUD_CLKIN1<br>0x2 = 010 : Source is AUD_CLKIN2<br>0x3 = 011 : Source is AUXOSC_XI<br>0x4 = 100 : Source is ATCLK0<br>0x5 = 101 : Source is ATCLK1<br>0x6 = 110 : Source is ATCLK2<br>0x7 = 111 : Source is ATCLK3 |
| 15-12 | Reserved | R | 0h | |
| 11-9 | McASP1_AHCLKR_SOURCE | R/W | 0h | Select the source clock of the MCASP1 AH clock receive<br>0x0 = 000 : Source is AUD_CLKIN0<br>0x1 = 001 : Source is AUD_CLKIN1<br>0x2 = 010 : Source is AUD_CLKIN2<br>0x3 = 011 : Source is AUXOSC_XI<br>0x4 = 100 : Source is ATCLK0<br>0x5 = 101 : Source is ATCLK1<br>0x6 = 110 : Source is ATCLK2<br>0x7 = 111 : Source is ATCLK3 |
| 8-6 | McASP1_AHCLKX_SOURCE | R/W | 0h | Select the source clock of the MCASP1 AH clock transmit<br>0x0 = 000 : Source is AUD_CLKIN0<br>0x1 = 001 : Source is AUD_CLKIN1<br>0x2 = 010 : Source is AUD_CLKIN2<br>0x3 = 011 : Source is AUXOSC_XI<br>0x4 = 100 : Source is ATCLK0<br>0x5 = 101 : Source is ATCLK1<br>0x6 = 110 : Source is ATCLK2<br>0x7 = 111 : Source is ATCLK3 |
| 5-3 | McASP0_AHCLKR_SOURCE | R/W | 0h | Select the source clock of the MCASP0 AH clock receive<br>0x0 = 000 : Source is AUD_CLKIN0<br>0x1 = 001 : Source is AUD_CLKIN1<br>0x2 = 010 : Source is AUD_CLKIN2<br>0x3 = 011 : Source is AUXOSC_XI<br>0x4 = 100 : Source is ATCLK0<br>0x5 = 101 : Source is ATCLK1<br>0x6 = 110 : Source is ATCLK2<br>0x7 = 111 : Source is ATCLK3 |
| 2-0 | McASP0_AHCLKX_SOURCE | R/W | 0h | Select the source clock of the MCASP0 AH clock transmit<br>0x0 = 000 : Source is AUD_CLKIN0<br>0x1 = 001 : Source is AUD_CLKIN1<br>0x2 = 010 : Source is AUD_CLKIN2<br>0x3 = 011 : Source is AUXOSC_XI<br>0x4 = 100 : Source is ATCLK0<br>0x5 = 101 : Source is ATCLK1<br>0x6 = 110 : Source is ATCLK2<br>0x7 = 111 : Source is ATCLK3 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.143 McBSP_UART_CLKSRC Register (offset = 2D8h) [reset = 0h]

McBSP_UART_CLKSRC is shown in Figure 2-169 and described in Table 2-185.

Select the clock source for UART3/4/5 and MCBSP_CLKS.

**Figure 2-169. McBSP_UART_CLKSRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | UART5_CLK_SOURCE |
| R-0h | | | | | | | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| UART5_CLK_SOURCE | UART4_CLK_SOURCE | | UART3_CLK_SOURCE | | McBSP_CLKS_SOURCE | | |
| R/W-0h | R/W-0h | | R/W-0h | | R/W-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-185. McBSP_UART_CLKSRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | Reserved | R | 0h | |
| 8-7 | UART5_CLK_SOURCE | R/W | 0h | Select the clock source for UART5<br>0x0 = 00 : SYSCLK8<br>0x1 = 01 : SYSCLK10<br>0x2 = 10 : SYSCLK6<br>0x3 = Reserved |
| 6-5 | UART4_CLK_SOURCE | R/W | 0h | Select the clock source for UART4<br>0x0 = 00 : SYSCLK8<br>0x1 = 01 : SYSCLK10<br>0x2 = 10 : SYSCLK6<br>0x3 = Reserved |
| 4-3 | UART3_CLK_SOURCE | R/W | 0h | Select the clock source for UART3<br>0x0 = 00 : SYSCLK8<br>0x1 = 01 : SYSCLK10<br>0x2 = 10 : SYSCLK6<br>0x3 = Reserved |
| 2-0 | McBSP_CLKS_SOURCE | R/W | 0h | Select the clock source for McBSP_CLKS<br>0x0 = 000 Source is CM_AUDIOCLK_MCBSP<br>0x1 = 001 Source is AUD_CLKIN0<br>0x2 = 010 Source is AUD_CLKIN1<br>0x3 = 011 Source is AUD_CLKIN2<br>0x4 = 100 Source is AUDOSC_XI<br>0x5 = Reserved |



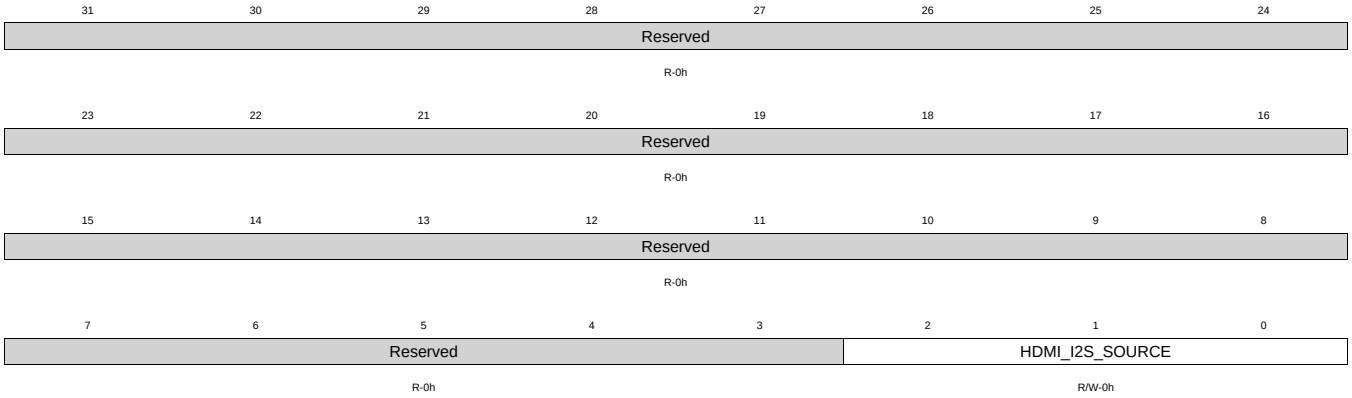
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.144 HDMI_I2S_CLKSRC Register (offset = 2DCh) [reset = 0h]

HDMI_I2S_CLKSRC is shown in Figure 2-170 and described in Table 2-186.

Select the clock source for HDMI_I2S.

**Figure 2-170. HDMI_I2S_CLKSRC Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-186. HDMI_I2S_CLKSRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-0 | HDMI_I2S_SOURCE | R/W | 0h | Select the source clock of the HDMI_I2S clock<br>0x0 = 000 : Source is PRCM out put i2s_mclk<br>0x1 = 001 : Source is AUD_CLKIN0<br>0x2 = 010 : Source is AUD_CLKIN1<br>0x3 = 011 : Source is AUD_CLKIN2<br>0x4 = 100 : Source is AUXOSC_XI<br>0x5 = Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.145 DMTIMER_CLKSRC Register (offset = 2E0h) [reset = 9240924h]

DMTIMER_CLKSRC is shown in Figure 2-171 and described in Table 2-187.

Select the clock source for DMTIMERs 1..8.

**Figure 2-171. DMTIMER_CLKSRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | DMTIMER7_SOURCE | | | DMTIMER6_SOURCE |
| R-0h | | | | R/W-4h | | | R/W-4h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| DMTIMER6_SOURCE | | DMTIMER5_SOURCE | | | DMTIMER4_SOURCE | | |
| R/W-4h | | R/W-4h | | | R/W-4h | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | DMTIMER3_SOURCE | | | DMTIMER2_SOURCE |
| R-0h | | | | R/W-4h | | | R/W-4h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DMTIMER2_SOURCE | | DMTIMER1_SOURCE | | | DMTIMER8_SOURCE | | |
| R/W-4h | | R/W-4h | | | R/W-4h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-187. DMTIMER_CLKSRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | |
| 27-25 | DMTIMER7_SOURCE | R/W | 4h | Select the source clock of the DMTIMER7 clock<br>0x0 = 000 : Source is SYSCLK18<br>0x1 = 001 : Source is AUD_CLKIN0<br>0x2 = 010 : Source is AUD_CLKIN1<br>0x3 = 011 : Source is AUD_CLKIN2<br>0x4 = 100 : Source is OSC0 output<br>0x5 = 101 : Source is AUXOSC_XI<br>0x6 = 110 : Source is TCLK input pin<br>0x7 = Reserved |
| 24-22 | DMTIMER6_SOURCE | R/W | 4h | Select the source clock of the DMTIMER6 clock<br>0x0 = 000 : Source is SYSCLK18<br>0x1 = 001 : Source is AUD_CLKIN0<br>0x2 = 010 : Source is AUD_CLKIN1<br>0x3 = 011 : Source is AUD_CLKIN2<br>0x4 = 100 : Source is OSC0 output<br>0x5 = 101 : Source is AUXOSC_XI<br>0x6 = 110 : Source is TCLK input pin<br>0x7 = Reserved |
| 21-19 | DMTIMER5_SOURCE | R/W | 4h | Select the source clock of the DMTIMER5 clock<br>0x0 = 000 : Source is SYSCLK18<br>0x1 = 001 : Source is AUD_CLKIN0<br>0x2 = 010 : Source is AUD_CLKIN1<br>0x3 = 011 : Source is AUD_CLKIN2<br>0x4 = 100 : Source is OSC0 output<br>0x5 = 101 : Source is AUXOSC_XI<br>0x6 = 110 : Source is TCLK input pin<br>0x7 = Reserved |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-187. DMTIMER_CLKSRC Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18-16 | DMTIMER4_SOURCE | R/W | 4h | Select the source clock of the DMTIMER4 clock<br>0x0 = 000 : Source is SYSCLK18<br>0x1 = 001 : Source is AUD_CLKIN0<br>0x2 = 010 : Source is AUD_CLKIN1<br>0x3 = 011 : Source is AUD_CLKIN2<br>0x4 = 100 : Source is OSC0 output<br>0x5 = 101 : Source is AUXOSC_XI<br>0x6 = 110 : Source is TCLK input pin<br>0x7 = Reserved |
| 15-12 | Reserved | R | 0h | |
| 11-9 | DMTIMER3_SOURCE | R/W | 4h | Select the source clock of the DMTIMER3 clock<br>0x0 = 000 : Source is SYSCLK18<br>0x1 = 001 : Source is AUD_CLKIN0<br>0x2 = 010 : Source is AUD_CLKIN1<br>0x3 = 011 : Source is AUD_CLKIN2<br>0x4 = 100 : Source is OSC0 output<br>0x5 = 101 : Source is AUXOSC_XI<br>0x6 = 110 : Source is TCLK input pin<br>0x7 = Reserved |
| 8-6 | DMTIMER2_SOURCE | R/W | 4h | Select the source clock of the DMTIMER2 clock<br>0x0 = 000 : Source is SYSCLK18<br>0x1 = 001 : Source is AUD_CLKIN0<br>0x2 = 010 : Source is AUD_CLKIN1<br>0x3 = 011 : Source is AUD_CLKIN2<br>0x4 = 100 : Source is OSC0 output<br>0x5 = 101 : Source is AUXOSC_XI<br>0x6 = 110 : Source is TCLK input pin<br>0x7 = Reserved |
| 5-3 | DMTIMER1_SOURCE | R/W | 4h | Select the source clock of the DMTIMER1 clock<br>0x0 = 000 : Source is SYSCLK18<br>0x1 = 001 : Source is AUD_CLKIN0<br>0x2 = 010 : Source is AUD_CLKIN1<br>0x3 = 011 : Source is AUD_CLKIN2<br>0x4 = 100 : Source is OSC0 output<br>0x5 = 101 : Source is AUXOSC_XI<br>0x6 = 110 : Source is TCLK input pin<br>0x7 = Reserved |
| 2-0 | DMTIMER8_SOURCE | R/W | 4h | Select the source clock of the DMTIMER8 clock<br>0x0 = 000 : Source is SYSCLK18<br>0x1 = 001 : Source is AUD_CLKIN0<br>0x2 = 010 : Source is AUD_CLKIN1<br>0x3 = 011 : Source is AUD_CLKIN2<br>0x4 = 100 : Source is OSC0 output<br>0x5 = 101 : Source is AUXOSC_XI<br>0x6 = 110 : Source is TCLK input pin<br>0x7 = Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.146 CLKOUT_MUX Register (offset = 2E4h) [reset = 0h]

CLKOUT_MUX is shown in Figure 2-172 and described in Table 2-188.

Control the options for CLKOUT multiplexing. The clock outputs are respectively CLKOUT0 and CLKOUT1.

**Figure 2-172. CLKOUT_MUX Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CLKOUT1_MUX | | | |
| R-0h | | | | R/W-0h | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CLKOUT0_MUX | | | |
| R-0h | | | | R/W-0h | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-188. CLKOUT_MUX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | CLKOUT1_MUX | R/W | 0h | Select the source clock of the CLKOUT1<br>0x0 = Source is PRCM_SYSCLK_OUT<br>0x1 = Source is SATA SERDES OBS CLK<br>0x2 = Source is PCIe SERDES OBS CLK<br>0x3 = Source is DSS_DPLL_OUT_DIV2<br>0x4 = Source is ISP_DPLL_OUT_DIV2<br>0x5 = Source is L3_DPLL<br>0x6 = Source is OSC0 OUT<br>0x7 = Source is OSC1 OUT<br>0x8 = DEVOSC clock out<br>0x9 = SGX_DPLL OUT<br>0xA = RCOSC32K OUT<br>0xB = Reserved |
| 15-4 | Reserved | R | 0h | |
| 3-0 | CLKOUT0_MUX | R/W | 0h | Select the source clock of the CLKOUT0<br>0x0 = Source is PRCM_SYSCLK_OUT<br>0x1 = Source is SATA SERDES OBS CLK<br>0x2 = Source is PCIe SERDES OBS CLK<br>0x3 = Source is DSS_DPLL_OUT_DIV2<br>0x4 = Source is ISP_DPLL_OUT_DIV2<br>0x5 = Source is L3_DPLL<br>0x6 = Source is OSC0 OUT<br>0x7 = Source is OSC1 OUT<br>0x8 = DEVOSC clock out<br>0x9 = SGX_DPLL OUT<br>0xA = RCOSC32K OUT<br>0xB = Reserved |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.147 RMII_REFCLK_SRC Register (offset = 2E8h) [reset = 0h]

RMII_REFCLK_SRC is shown in Figure 2-173 and described in Table 2-189.

Control the Cpts_RFT_CLK source mux and RMII_Ref_CLK source mux.

**Figure 2-173. RMII_REFCLK_SRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | CPTS_RFT_CLK | | | REFCLK_SOURCE |
| R-0h | | | | R/W-0h | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-189. RMII_REFCLK_SRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | Reserved | R | 0h | |
| 3-1 | CPTS_RFT_CLK | R/W | 0h | Select the source clock of the CPTS_RFT_CLK<br>0x0 = 000 : Source is VIDEO0_PLL_OUT<br>0x1 = 001 : Source is VIDEO1_PLL_OUT<br>0x2 = 010 : Source is AUDIO_PLL_OUT<br>0x3 = 011 : Source is VIDEO_PLL_CLK2 from videopll_clk2_mx<br>0x4 = 100 : Source is L3_PLL OUT<br>0x5 = Reserved |
| 0 | REFCLK_SOURCE | R/W | 0h | Select the source clock of the SYSCLK8<br>0x0 : Source is 50MHz clock SATA SerDes<br>0x1 : Source is REFCLK PIN |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.148 SYSCLK18_CLKSRC Register (offset = 2F0h) [reset = 0h]

SYSCLK18_CLKSRC is shown in Figure 2-174 and described in Table 2-190.

Select the clock source of SYSCLK18.

**Figure 2-174. SYSCLK18_CLKSRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | SYSCLK18_SOURCE |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-190. SYSCLK18_CLKSRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | SYSCLK18_SOURCE | R/W | 0h | Select the source of SYSCLK18 input to PRCM<br>0x0 : Source is RTCDIVIDER OUT<br>0x1 : Source is CLKIN32 PIN |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.149 WDT0_CLKSRC Register (offset = 2F4h) [reset = 0h]

WDT0_CLKSRC is shown in Figure 2-175 and described in Table 2-191.

Select the clock source of WDT0.

**Figure 2-175. WDT0_CLKSRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | WDT0_SOURCE |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-191. WDT0_CLKSRC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | WDT0_SOURCE | R/W | 0h | Select the source of SYSCLK18 input to PRCM<br>0x0 : Source is RTCDIVIDER OUT<br>0x1 : Source is RCOSC 32K OUT |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.150 DMTIMER_CLK_CHANGE Register (offset = 320h) [reset = 0h]

DMTIMER_CLK_CHANGE is shown in Figure 2-176 and described in Table 2-192. This register controls the IDLE REQUEST and logs the IDLE STATUS for DMTIMERs 1..8.

**Figure 2-176. DMTIMER_CLK_CHANGE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DMTIMER7_IDLESTATUS | DMTIMER6_IDLESTATUS | DMTIMER5_IDLESTATUS | DMTIMER4_IDLESTATUS | DMTIMER3_IDLESTATUS | DMTIMER2_IDLESTATUS | DMTIMER1_IDLESTATUS | DMTIMER8_IDLESTATUS |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DMTIMER7_IDLEREQ | DMTIMER6_IDLEREQ | DMTIMER5_IDLEREQ | DMTIMER4_IDLEREQ | DMTIMER3_IDLEREQ | DMTIMER2_IDLEREQ | DMTIMER1_IDLEREQ | DMTIMER8_IDLEREQ |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-192. DMTIMER_CLK_CHANGE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15 | DMTIMER7_IDLESTATUS | R | 0h | IDLE Status for DMTIMER7<br>0x0 : TIMER not IDLE<br>0x1 : TIMER IDLE |
| 14 | DMTIMER6_IDLESTATUS | R | 0h | IDLE Status for DMTIMER6<br>0x0 : TIMER not IDLE<br>0x1 : TIMER IDLE |
| 13 | DMTIMER5_IDLESTATUS | R | 0h | IDLE Status for DMTIMER5<br>0x0 : TIMER not IDLE<br>0x1 : TIMER IDLE |
| 12 | DMTIMER4_IDLESTATUS | R | 0h | IDLE Status for DMTIMER4<br>0x0 : TIMER not IDLE<br>0x1 : TIMER IDLE |
| 11 | DMTIMER3_IDLESTATUS | R | 0h | IDLE Status for DMTIMER3<br>0x0 : TIMER not IDLE<br>0x1 : TIMER IDLE |
| 10 | DMTIMER2_IDLESTATUS | R | 0h | IDLE Status for DMTIMER2<br>0x0 : TIMER not IDLE<br>0x1 : TIMER IDLE |
| 9 | DMTIMER1_IDLESTATUS | R | 0h | IDLE Status for DMTIMER1<br>0 = 0 : TIMER not IDLE<br>1 = 1 : TIMER IDLE |
| 8 | DMTIMER8_IDLESTATUS | R | 0h | IDLE Status for DMTIMER8<br>0x0 : TIMER not IDLE<br>0x1 : TIMER IDLE |
| 7 | DMTIMER7_IDLEREQ | R/W | 0h | IDLE Request assertion status for DMTIMER7<br>0x0 : IDLE REQ not asserted<br>0x1 : IDLE REQ asserted |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-192. DMTIMER_CLK_CHANGE Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | DMTIMER6_IDLEREQ | R/W | 0h | IDLE Request assertion status for DMTIMER6<br>0x0 : IDLE REQ not asserted<br>0x1 : IDLE REQ asserted |
| 5 | DMTIMER5_IDLEREQ | R/W | 0h | IDLE Request assertion status for DMTIMER5<br>0x0 : IDLE REQ not asserted<br>0x1 : IDLE REQ asserted |
| 4 | DMTIMER4_IDLEREQ | R/W | 0h | IDLE Request assertion status for DMTIMER4<br>0x0 : IDLE REQ not asserted<br>0x1 : IDLE REQ asserted |
| 3 | DMTIMER3_IDLEREQ | R/W | 0h | IDLE Request assertion status for DMTIMER3<br>0x0 : IDLE REQ not asserted<br>0x1 : IDLE REQ asserted |
| 2 | DMTIMER2_IDLEREQ | R/W | 0h | IDLE Request assertion status for DMTIMER2<br>0x0 : IDLE REQ not asserted<br>0x1 : IDLE REQ asserted |
| 1 | DMTIMER1_IDLEREQ | R/W | 0h | IDLE Request assertion status for DMTIMER1<br>0x0 : IDLE REQ not asserted<br>0x1 : IDLE REQ asserted |
| 0 | DMTIMER8_IDLEREQ | R/W | 0h | IDLE Request assertion status for DMTIMER8<br>0x0 : IDLE REQ not asserted<br>0x1 : IDLE REQ asserted |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.151 DEEPSLEEP_CTRL Register (offset = 324h) [reset = 6A75h]

DEEPSLEEP_CTRL is shown in Figure 2-177 and described in Table 2-193.

The DEEPSLEEP control register is used to control deep sleep power mode. Deep sleep is not supported on this SoC.

**Figure 2-177. DEEPSLEEP_CTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | DSENABLE | DSPOLARITY |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DSCOUNT | | | | | | | |
| R/W-6A75h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DSCOUNT | | | | | | | |
| R/W-6A75h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-193. DEEPSLEEP_CTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17 | DSENABLE | R/W | 0h | '1' enables the deep sleep circuitry, and allows the device pin DEEPSLEEPZ(TBD) to have control |
| 16 | DSPOLARITY | R/W | 0h | controls the polarity of DEEPSLEEPZ(TBD) pin. '0' means pin is active low. '1' means pin is active high. |
| 15-0 | DSCOUNT | R/W | 6A75h | programmable count of how many osc clocks it needs to see prior to exiting deep sleep mode |

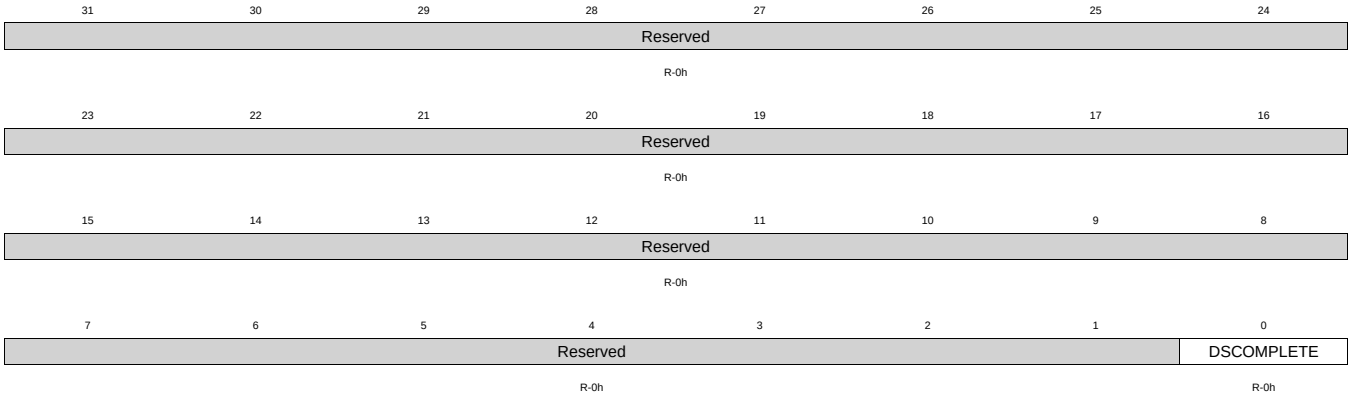

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.1.152 DEEPSLEEP_STATUS Register (offset = 328h) [reset = 0h]

DEEPSLEEP_STATUS is shown in Figure 2-178 and described in Table 2-194.

The DEEPSLEEP control register logs the internal deepsleep counter value.

**Figure 2-178. DEEPSLEEP_STATUS Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-194. DEEPSLEEP_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | DSCOMPLETE | R | 0h | 1' if DSCOUNT value equals the internal deepsleep counter value |

## *2.10.2 PRM_DEVICE Registers*

Table 2-195 lists the memory-mapped registers for the PRM_DEVICE. All register offset addresses not listed in Table 2-195 should be considered as reserved locations and the register contents should not be modified.

**Table 2-195. PRM_DEVICE REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| A0h | PRM_RSTCTRL | Global software cold and warm reset control.<br>This register is auto-cleared.<br>Only write 1 is possible.<br>A read returns 0 only. | Section 2.10.2.1 |
| A4h | PRM_RSTTIME | Reset duration control. | Section 2.10.2.2 |
| A8h | PRM_RSTST | This register logs the global reset sources.<br>Each bit is set upon release of the domain reset signal.<br>Must be cleared by software.<br>[warm reset insensitive] | Section 2.10.2.3 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.2.1 PRM_RSTCTRL Register (offset = A0h) [reset = 0h]

PRM_RSTCTRL is shown in Figure 2-179 and described in Table 2-196.

Global software cold and warm reset control. This register is auto-cleared. Only write 1 is possible. A read returns 0 only.

**Figure 2-179. PRM_RSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | RST_GLOBAL_COLD_SW | RST_GLOBAL_WARM_SW |
| R-0h | | | | | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-196. PRM_RSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0h | |
| 1 | RST_GLOBAL_COLD_SW | R/W | 0h | Global COLD software reset control. This bit is reset only upon a global cold source of reset.<br>0x0: Global COLD software reset is cleared.<br>0x1: Asserts a global COLD software reset. The software must ensure the SDRAM is properly put in sef-refresh mode before applying this reset. |
| 0 | RST_GLOBAL_WARM_SW | R/W | 0h | Global WARM software reset control. This bit is reset upon any global source of reset (warm and cold).<br>0x0: Global warm software reset is cleared.<br>0x1: Asserts a global warm software reset. |

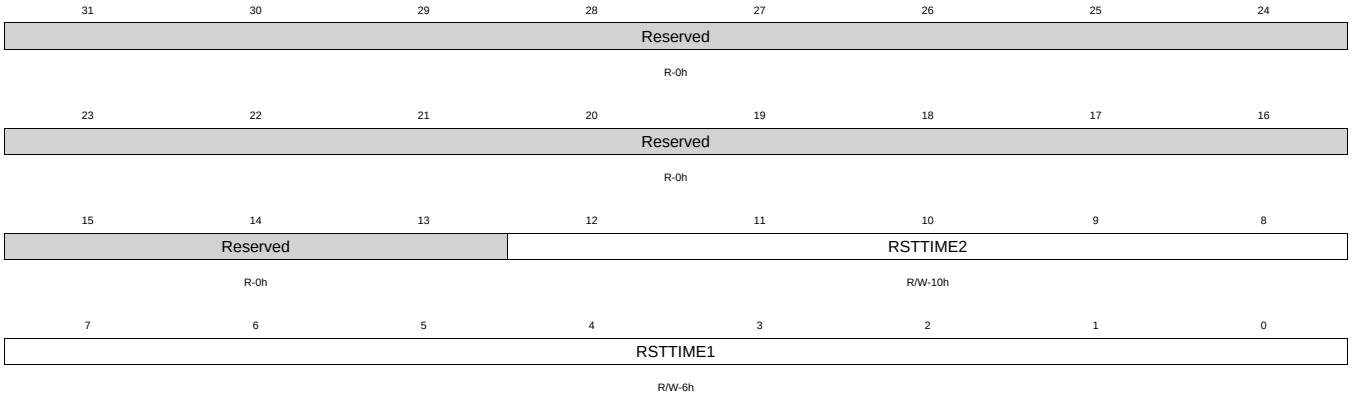

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.2.2 PRM_RSTTIME Register (offset = A4h) [reset = 1006h]

PRM_RSTTIME is shown in Figure 2-180 and described in Table 2-197.

Reset duration control.

**Figure 2-180. PRM_RSTTIME Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-197. PRM_RSTTIME Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | Reserved | R | 0h | |
| 12-8 | RSTTIME2 | R/W | 10h | (Power domain) reset duration 2 (number of SYS_CLK clock cycles) |
| 7-0 | RSTTIME1 | R/W | 6h | (Global) reset duration 1 (number of SYS_CLK clock cycles) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.2.3 PRM_RSTST Register (offset = A8h) [reset = 1h]

PRM_RSTST is shown in Figure 2-181 and described in Table 2-198.

This register logs the global reset sources. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

**Figure 2-181. PRM_RSTST Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | ICEPICK_RST | Reserved |
| R-0h | | | | | | R/W-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | EXTERNAL_WARM_RST | Reserved | MPU_WDT_RST | Reserved | GLOBAL_WARM_SW_RST | GLOBAL_COLD_RST |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-198. PRM_RSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | Reserved | R | 0h | |
| 9 | ICEPICK_RST | R/W | 0h | IcePick reset event. This is a source of global warm reset initiated by the emulation. [warm reset insensitive]<br>0x0: No ICEPICK reset.<br>0x1: IcePick reset has occurred. |
| 8-6 | Reserved | R | 0h | |
| 5 | EXTERNAL_WARM_RST | R/W | 0h | External warm reset event [warm reset insensitive]<br>0x0: No global warm reset.<br>0x1: Global external warm reset has occurred. |
| 4 | Reserved | R/W | 0h | |
| 3 | MPU_WDT_RST | R/W | 0h | MPU Watchdog timer reset event. This is a source of global WARM reset. [warm reset insensitive]<br>0x0: No MPU watchdog reset.<br>0x1: MPU wachtdog reset has occurred. |
| 2 | Reserved | R/W | 0h | |
| 1 | GLOBAL_WARM_SW_RST | R/W | 0h | Global warm software reset event [warm reset insensitive]<br>0x0: No global warm SW reset<br>0x1: Global warm SW reset has occurred. |
| 0 | GLOBAL_COLD_RST | R/W | 1h | Power-on (cold) reset event [warm reset insensitive]<br>0x0: No power-on reset.<br>0x1: Power-on reset has occurred. |

## *2.10.3 CM_DEVICE Registers*

Table 2-199 lists the memory-mapped registers for the CM_DEVICE. All register offset addresses not listed in Table 2-199 should be considered as reserved locations and the register contents should not be modified.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-199. CM_DEVICE REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | CM_CLKOUT_CTRL | This register provides the control over SYS_CLKOUT output | Section 2.10.3.1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.3.1 CM_CLKOUT_CTRL Register (offset = 0h) [reset = 0h]

CM_CLKOUT_CTRL is shown in Figure 2-182 and described in Table 2-200.

This register provides the control over SYS_CLKOUT output

**Figure 2-182. CM_CLKOUT_CTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLKOUTEN | Reserved | CLKOUTDIV | | | Reserved | CLKOUTSOURCE | |
| R/W-0h | R-0h | R/W-0h | | | R-0h | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-200. CM_CLKOUT_CTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | Reserved | R | 0h | |
| 7 | CLKOUTEN | R/W | 0h | This bit controls the external clock activity<br>0x0: DIS : SYS_CLKOUT is disabled<br>0x1: EN : SYS_CLKOUT is enabled |
| 6 | Reserved | R | 0h | |
| 5-3 | CLKOUTDIV | R/W | 0h | This field controls the external clock divison factor of the CLKOUT Second Pin CLKOUT2<br>0x0 = SYS_CLKOUT2/1<br>0x1 = SYS_CLKOUT2/2<br>0x2 = SYS_CLKOUT2/3<br>0x3 = SYS_CLKOUT2/4<br>0x4 = SYS_CLKOUT2/5<br>0x5 = SYS_CLKOUT2/6<br>0x6 = SYS_CLKOUT2/7<br>0x7 = SYS_CLKOUT2/8 |
| 2 | Reserved | R | 0h | |
| 1-0 | CLKOUTSOURCE | R/W | 0h | This field selects the external output clock source<br>0x0: MAIN_PLL_CLK : DEVOSC clock<br>0x1: DPLL_DSP_CLK : USB PLL clock output<br>0x2: VIDEO_PLL_CLK : VIDEO0 PLL clock output<br>0x3: AUDIO_PLL_CLK : RTCDIVIDER output |

## *2.10.4 OCP_SOCKET_PRM Registers*

Table 2-201 lists the memory-mapped registers for the OCP_SOCKET_PRM. All register offset addresses not listed in Table 2-201 should be considered as reserved locations and the register contents should not be modified.

**Table 2-201. OCP_SOCKET_PRM REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | REVISION_PRM | This register contains the IP revision code for the PRCM | Section 2.10.4.1 |



Copyright © 2011–2016, Texas Instruments Incorporated

#### 2.10.4.1 REVISION_PRM Register (offset = 0h) [reset = 10h]

REVISION_PRM is shown in Figure 2-183 and described in Table 2-202.

This register contains the IP revision code for the PRCM

**Figure 2-183. REVISION_PRM Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rev | | | | | | | |
| R-10h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-202. REVISION_PRM Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | Reserved | R | 0h | |
| 7-0 | Rev | R | 10h | IP revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 1.0, 0x21 for 2.1 |

### *2.10.5 CM_DPLL Registers*

Table 2-203 lists the memory-mapped registers for the CM_DPLL. All register offset addresses not listed in Table 2-203 should be considered as reserved locations and the register contents should not be modified.

**Table 2-203. CM_DPLL REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 8h | CM_SYSCLK3_CLKSEL | Selects the divider value for SYSCLK3 | Section 2.10.5.1 |
| 24h | CM_SYSCLK10_CLKSEL | Selects the divider value for SYSCLK10 | Section 2.10.5.2 |
| 40h | CM_VPB3_CLKSEL | Selects the divider value for Video PLL B3 divider | Section 2.10.5.3 |
| 44h | CM_VPC1_CLKSEL | Selects the divider value for Video PLL C1 divider | Section 2.10.5.4 |
| 48h | CM_VPD1_CLKSEL | Selects the divider value for Video PLL D1 divider | Section 2.10.5.5 |
| 4Ch | CM_SYSCLK19_CLKSEL | Selects the divider value for SYSCLK19 clock | Section 2.10.5.6 |
| 50h | CM_SYSCLK20_CLKSEL | Selects the divider value for SYSCLK20 clock | Section 2.10.5.7 |
| 54h | CM_SYSCLK21_CLKSEL | Selects the divider value for SYSCLK21 clock | Section 2.10.5.8 |
| 58h | CM_SYSCLK22_CLKSEL | Selects the divider value for SYSCLK22 clock | Section 2.10.5.9 |
| 5Ch | CM_APA_CLKSEL | Selects the divider value for Audio PLL A divider | Section 2.10.5.10 |
| 70h | CM_SYSCLK14_CLKSEL | Selects the Mux select line for SYSCLK14 clock | Section 2.10.5.11 |
| 74h | CM_SYSCLK16_CLKSEL | Selects the Mux select line for SYSCLK16 clock | Section 2.10.5.12 |
| 78h | CM_SYSCLK18_CLKSEL | Selects the Mux select line for SYSCLK18 clock | Section 2.10.5.13 |
| 7Ch | CM_AUDIOCLK_MCASP0_CLKSEL | Selects the Mux select line for McASP0 audio clock | Section 2.10.5.14 |
| 80h | CM_AUDIOCLK_MCASP1_CLKSEL | Selects the Mux select line for McASP1 audio clock | Section 2.10.5.15 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-203. CM_DPLL REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 84h | CM_AUDIOCLK_MCASP2_CLKSEL | Selects the Mux select line for McASP2 audio clock | Section 2.10.5.16 |
| 88h | CM_AUDIOCLK_MCBSP_CLKSEL | Selects the Mux select line for McBSP audio clock | Section 2.10.5.17 |
| ACh | CM_HDMI_CLKSEL | Selects the Mux select line for HDMI audio clock | Section 2.10.5.18 |
| B0h | CM_SYSCLK23_CLKSEL | Selects the divider value for SYSCLK23 which is the SGX functional clock | Section 2.10.5.19 |

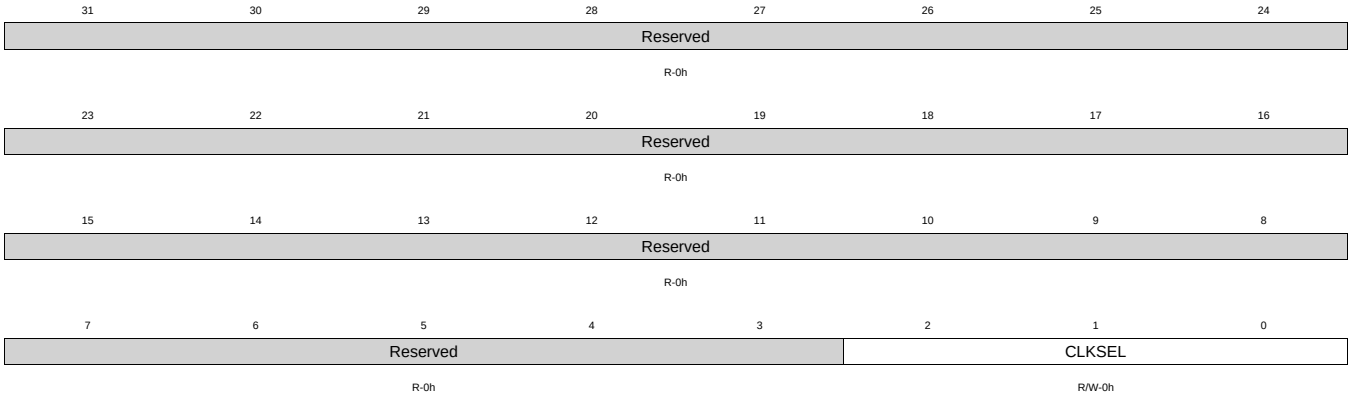

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.1 CM_SYSCLK3_CLKSEL Register (offset = 8h) [reset = 0h]

CM_SYSCLK3_CLKSEL is shown in Figure 2-184 and described in Table 2-204.

Selects the divider value for SYSCLK3

**Figure 2-184. CM_SYSCLK3_CLKSEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-204. CM_SYSCLK3_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2-0 | CLKSEL | R/W | 0h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_3 : Select SYS_CLK divided by 3<br>0x3 = SYSCLK_DIV_4 : Select SYS_CLK divided by 4<br>0x4 = SYSCLK_DIV_5 : Select SYS_CLK divided by 5<br>0x5 = SYSCLK_DIV_6 : Select SYS_CLK divided by 6.<br>0x6 = SYSCLK_DIV_7 : Select SYS_CLK divided by 7.<br>0x7 = SYSCLK_DIV_8 : Select SYS_CLK divided by 8. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.2 CM_SYSCLK10_CLKSEL Register (offset = 24h) [reset = 1h]

CM_SYSCLK10_CLKSEL is shown in Figure 2-185 and described in Table 2-205.

Selects the divider value for SYSCLK10

**Figure 2-185. CM_SYSCLK10_CLKSEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | Reserved | | | CLKSEL | | |
| R-0h | | R-0h | | | R/W-1h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-205. CM_SYSCLK10_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-3 | Reserved | R | 0h | |
| 2-0 | CLKSEL | R/W | 1h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_3 : Select SYS_CLK divided by 3<br>0x3 = SYSCLK_DIV_4 : Select SYS_CLK divided by 4<br>0x4 = SYSCLK_DIV_5 : Select SYS_CLK divided by 5<br>0x5 = SYSCLK_DIV_6 : Select SYS_CLK divided by 6<br>0x6 = SYSCLK_DIV_7 : Select SYS_CLK divided by 7<br>0x7 = SYSCLK_DIV_8 : Select SYS_CLK divided by 8 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.3 CM_VPB3_CLKSEL Register (offset = 40h) [reset = 2h]

CM_VPB3_CLKSEL is shown in Figure 2-186 and described in Table 2-206.

Selects the divider value for Video PLL B3 divider

**Figure 2-186. CM_VPB3_CLKSEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | Reserved | | | Reserved | CLKSEL | |
| R-0h | | R-0h | | | R-0h | R/W-2h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-206. CM_VPB3_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-3 | Reserved | R | 0h | |
| 2 | Reserved | R | 0h | |
| 1-0 | CLKSEL | R/W | 2h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_22 : Select SYS_CLK divided by 22<br>0x3 = Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.4 CM_VPC1_CLKSEL Register (offset = 44h) [reset = 3h]

CM_VPC1_CLKSEL is shown in Figure 2-187 and described in Table 2-207.

Selects the divider value for Video PLL C1 divider

**Figure 2-187. CM_VPC1_CLKSEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | Reserved | | | Reserved | CLKSEL | |
| R-0h | | R-0h | | | R-0h | R/W-3h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-207. CM_VPC1_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-3 | Reserved | R | 0h | |
| 2 | Reserved | R | 0h | |
| 1-0 | CLKSEL | R/W | 3h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_22 : Select SYS_CLK divided by 22<br>0x3 = Reserved |

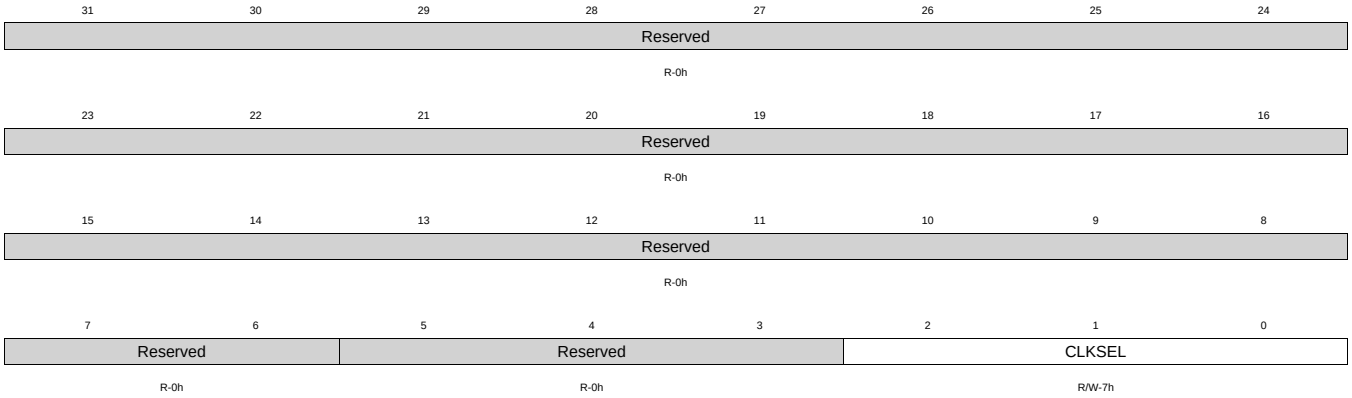

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.5 CM_VPD1_CLKSEL Register (offset = 48h) [reset = 7h]

CM_VPD1_CLKSEL is shown in Figure 2-188 and described in Table 2-208.

Selects the divider value for Video PLL D1 divider

**Figure 2-188. CM_VPD1_CLKSEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-208. CM_VPD1_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-3 | Reserved | R | 0h | |
| 2-0 | CLKSEL | R/W | 7h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_3 : Select SYS_CLK divided by 3<br>0x3 = SYSCLK_DIV_4 : Select SYS_CLK divided by 4<br>0x4 = SYSCLK_DIV_5 : Select SYS_CLK divided by 5<br>0x5 = SYSCLK_DIV_6 : Select SYS_CLK divided by 6<br>0x6 = SYSCLK_DIV_7 : Select SYS_CLK divided by 7<br>0x7 = SYSCLK_DIV_8 : Select SYS_CLK divided by 8 |

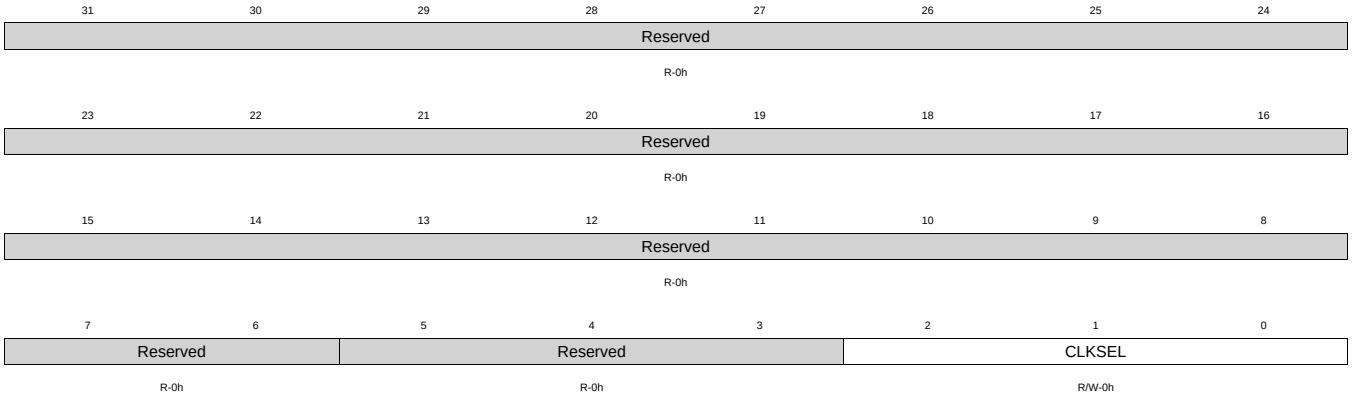

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.6 CM_SYSCLK19_CLKSEL Register (offset = 4Ch) [reset = 0h]

CM_SYSCLK19_CLKSEL is shown in Figure 2-189 and described in Table 2-209.

Selects the divider value for SYSCLK19 clock

**Figure 2-189. CM_SYSCLK19_CLKSEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-209. CM_SYSCLK19_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-3 | Reserved | R | 0h | |
| 2-0 | CLKSEL | R/W | 0h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_3 : Select SYS_CLK divided by 3<br>0x3 = SYSCLK_DIV_4 : Select SYS_CLK divided by 4<br>0x4 = SYSCLK_DIV_5 : Select SYS_CLK divided by 5<br>0x5 = SYSCLK_DIV_6 : Select SYS_CLK divided by 6<br>0x6 = SYSCLK_DIV_7 : Select SYS_CLK divided by 7<br>0x7 = SYSCLK_DIV_8 : Select SYS_CLK divided by 8 |

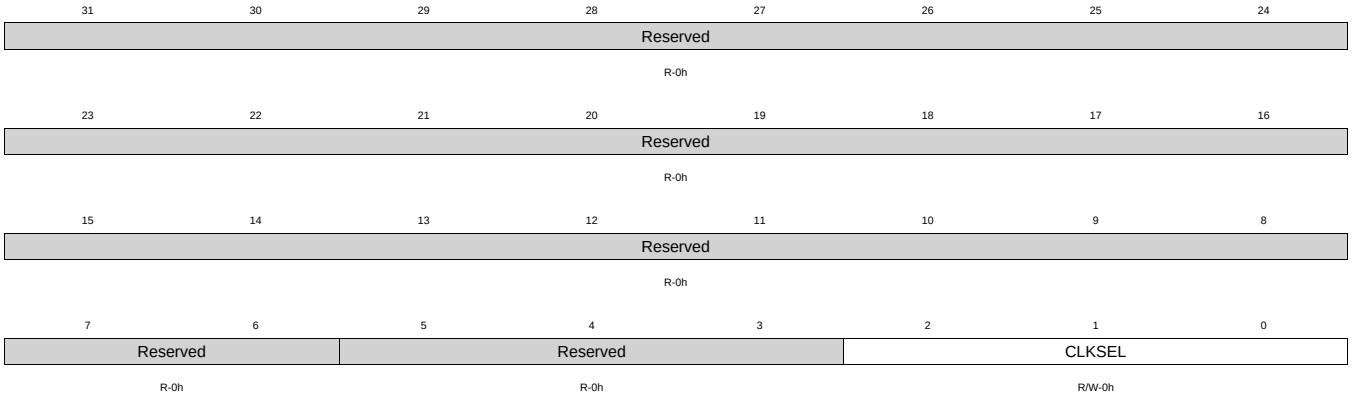

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.7 CM_SYSCLK20_CLKSEL Register (offset = 50h) [reset = 0h]

CM_SYSCLK20_CLKSEL is shown in Figure 2-190 and described in Table 2-210.

Selects the divider value for SYSCLK20 clock

**Figure 2-190. CM_SYSCLK20_CLKSEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-210. CM_SYSCLK20_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-3 | Reserved | R | 0h | |
| 2-0 | CLKSEL | R/W | 0h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_3 : Select SYS_CLK divided by 3<br>0x3 = SYSCLK_DIV_4 : Select SYS_CLK divided by 4<br>0x4 = SYSCLK_DIV_5 : Select SYS_CLK divided by 5<br>0x5 = SYSCLK_DIV_6 : Select SYS_CLK divided by 6<br>0x6 = SYSCLK_DIV_7 : Select SYS_CLK divided by 7<br>0x7 = SYSCLK_DIV_8 : Select SYS_CLK divided by 8 |

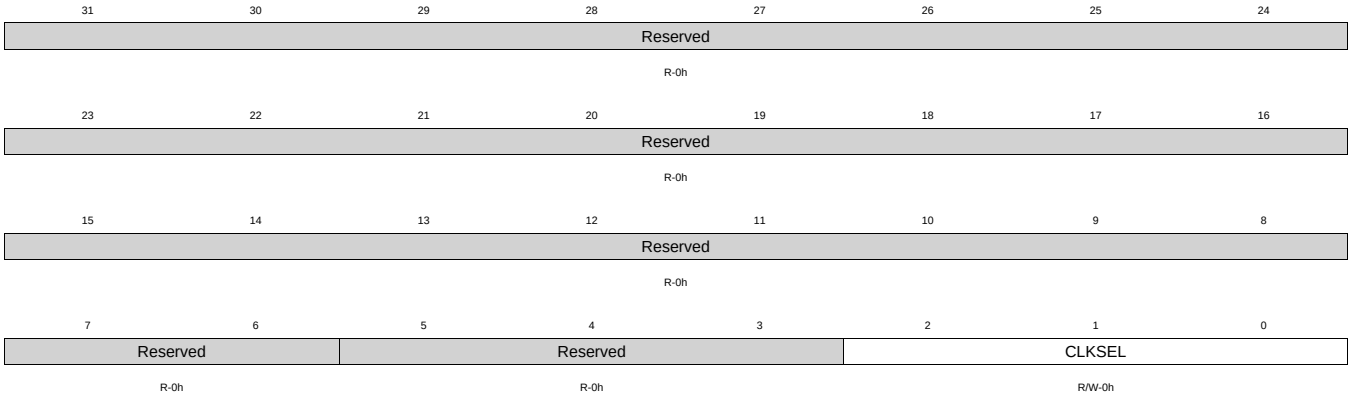

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.8 CM_SYSCLK21_CLKSEL Register (offset = 54h) [reset = 0h]

CM_SYSCLK21_CLKSEL is shown in Figure 2-191 and described in Table 2-211.

Selects the divider value for SYSCLK21 clock

**Figure 2-191. CM_SYSCLK21_CLKSEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-211. CM_SYSCLK21_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-3 | Reserved | R | 0h | |
| 2-0 | CLKSEL | R/W | 0h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_3 : Select SYS_CLK divided by 3<br>0x3 = SYSCLK_DIV_4 : Select SYS_CLK divided by 4<br>0x4 = SYSCLK_DIV_5 : Select SYS_CLK divided by 5<br>0x5 = SYSCLK_DIV_6 : Select SYS_CLK divided by 6<br>0x6 = SYSCLK_DIV_7 : Select SYS_CLK divided by 7<br>0x7 = SYSCLK_DIV_8 : Select SYS_CLK divided by 8 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.9 CM_SYSCLK22_CLKSEL Register (offset = 58h) [reset = 0h]

CM_SYSCLK22_CLKSEL is shown in Figure 2-192 and described in Table 2-212.

Selects the divider value for SYSCLK22 clock

**Figure 2-192. CM_SYSCLK22_CLKSEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | Reserved | | | CLKSEL | | |
| R-0h | | R-0h | | | R/W-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-212. CM_SYSCLK22_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-3 | Reserved | R | 0h | |
| 2-0 | CLKSEL | R/W | 0h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_3 : Select SYS_CLK divided by 3<br>0x3 = SYSCLK_DIV_4 : Select SYS_CLK divided by 4<br>0x4 = SYSCLK_DIV_5 : Select SYS_CLK divided by 5<br>0x5 = SYSCLK_DIV_6 : Select SYS_CLK divided by 6<br>0x6 = SYSCLK_DIV_7 : Select SYS_CLK divided by 7<br>0x7 = SYSCLK_DIV_8 : Select SYS_CLK divided by 8 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.10 CM_APA_CLKSEL Register (offset = 5Ch) [reset = 0h]

CM_APA_CLKSEL is shown in Figure 2-193 and described in Table 2-213.

Selects the divider value for Audio PLL A divider

**Figure 2-193. CM_APA_CLKSEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | Reserved | | | CLKSEL | | |
| R-0h | | R-0h | | | R/W-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-213. CM_APA_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-3 | Reserved | R | 0h | |
| 2-0 | CLKSEL | R/W | 0h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_3 : Select SYS_CLK divided by 3<br>0x3 = SYSCLK_DIV_4 : Select SYS_CLK divided by 4<br>0x4 = SYSCLK_DIV_5 : Select SYS_CLK divided by 5<br>0x5 = SYSCLK_DIV_6 : Select SYS_CLK divided by 6<br>0x6 = SYSCLK_DIV_7 : Select SYS_CLK divided by 7<br>0x7 = SYSCLK_DIV_8 : Select SYS_CLK divided by 8 |

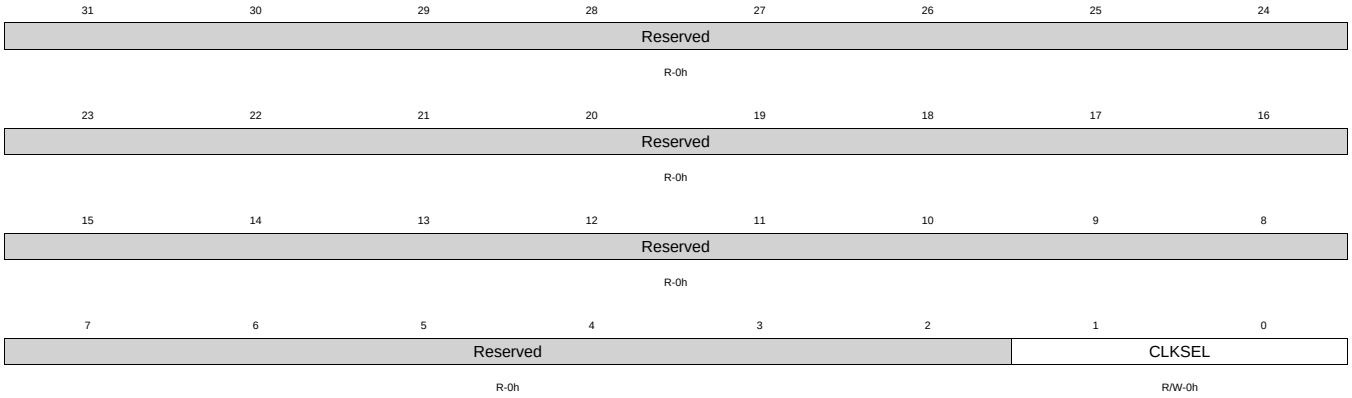

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.11 CM_SYSCLK14_CLKSEL Register (offset = 70h) [reset = 0h]

CM_SYSCLK14_CLKSEL is shown in Figure 2-194 and described in Table 2-214.

Selects the Mux select line for SYSCLK14 clock

**Figure 2-194. CM_SYSCLK14_CLKSEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-214. CM_SYSCLK14_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0h | |
| 1-0 | CLKSEL | R/W | 0h | CLKIN (DPLL_VIDEO1 reference clock)<br>0x0 = SEL1 : Select SYSCLK14 to be B3 divider output<br>0x1 = SEL2 : Select SYSCLK14 to be CLKIN<br>0x2 = SEL3 : Select SYSCLK14 to be C1 divider output<br>0x3 = Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.12 CM_SYSCLK16_CLKSEL Register (offset = 74h) [reset = 0h]

CM_SYSCLK16_CLKSEL is shown in Figure 2-195 and described in Table 2-215.

Selects the Mux select line for SYSCLK16 clock

**Figure 2-195. CM_SYSCLK16_CLKSEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | CLKSEL |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-215. CM_SYSCLK16_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | CLKSEL | R/W | 0h | Selects the Mux select line for SYSCLK16 [warm reset insensitive]<br>0x0 = SEL1 : Select SYSCLK16 to be D1 divider output<br>0x1 = SEL2 : Select SYSCLK16 to be divider B3 output |



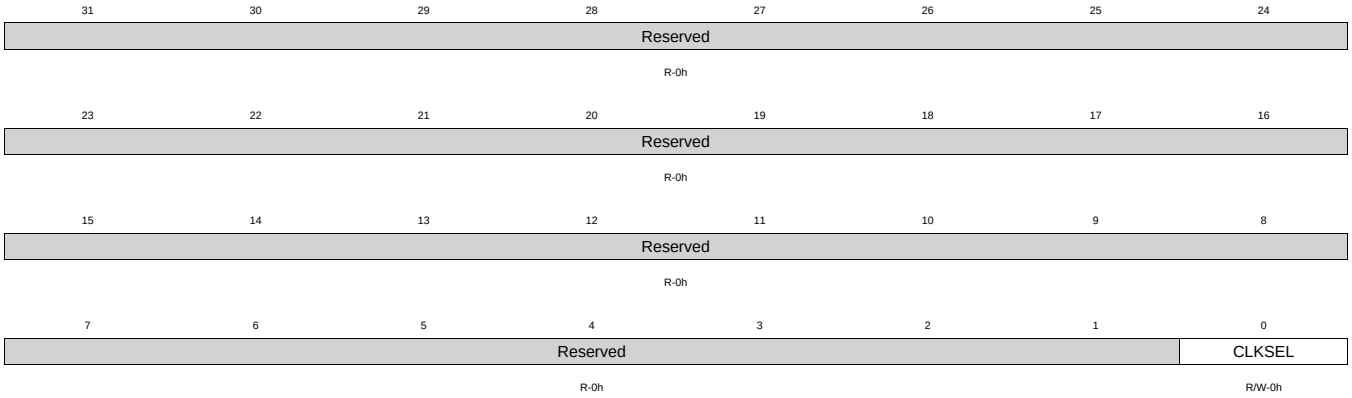
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.13 CM_SYSCLK18_CLKSEL Register (offset = 78h) [reset = 0h]

CM_SYSCLK18_CLKSEL is shown in Figure 2-196 and described in Table 2-216.

Selects the Mux select line for SYSCLK18 clock.

**Figure 2-196. CM_SYSCLK18_CLKSEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-216. CM_SYSCLK18_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | CLKSEL | R/W | 0h | Selects the Mux select line for SYSCLK18 [warm reset insensitive]<br>0x0 = SEL1 : Select 32 Khz Clock<br>0x1 = SEL2 : Select Audio PLL generated 32 Khz Clock |

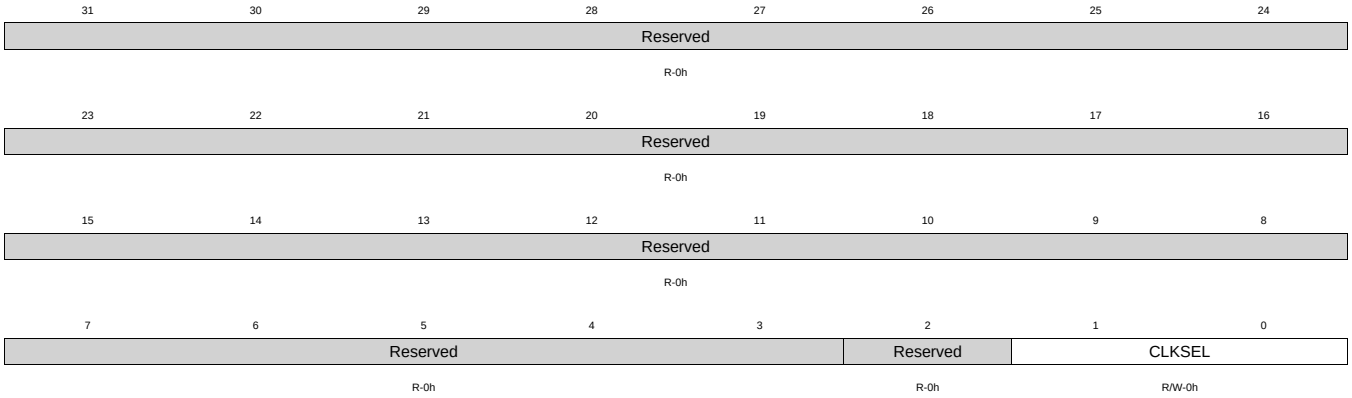

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.14 CM_AUDIOCLK_MCASP0_CLKSEL Register (offset = 7Ch) [reset = 0h]

CM_AUDIOCLK_MCASP0_CLKSEL is shown in Figure 2-197 and described in Table 2-217.

Selects the Mux select line for McASP0 audio clock

**Figure 2-197. CM_AUDIOCLK_MCASP0_CLKSEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-217. CM_AUDIOCLK_MCASP0_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2 | Reserved | R | 0h | |
| 1-0 | CLKSEL | R/W | 0h | Selects the Mux select line for McASP0 audio clock [warm reset insensitive]<br>0x0 = SEL1 : Select McASP0 audio clock to be SYSCLK20<br>0x1 = SEL2 : Select McASP0 audio clock to be SYSCLK21<br>0x2 = SEL3 : Select McASP0 audio clock to be SYSCLK22<br>0x3 = Reserved |

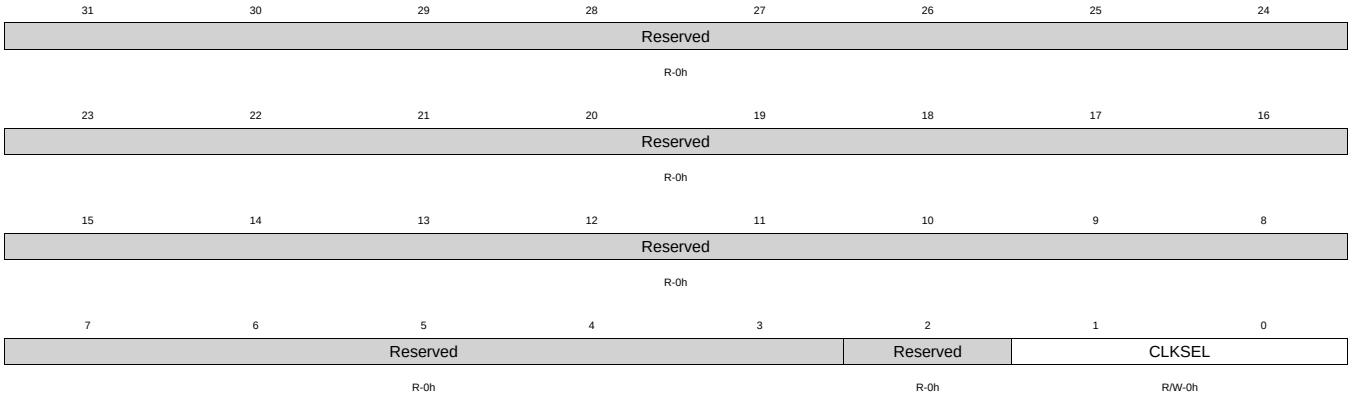

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.15 CM_AUDIOCLK_MCASP1_CLKSEL Register (offset = 80h) [reset = 0h]

CM_AUDIOCLK_MCASP1_CLKSEL is shown in Figure 2-198 and described in Table 2-218.

Selects the Mux select line for McASP1 audio clock

**Figure 2-198. CM_AUDIOCLK_MCASP1_CLKSEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-218. CM_AUDIOCLK_MCASP1_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2 | Reserved | R | 0h | |
| 1-0 | CLKSEL | R/W | 0h | Selects the Mux select line for McASP1 audio clock [warm reset insensitive]<br>0x0 = SEL1 : Select McASP1 audio clock to be SYSCLK20<br>0x1 = SEL2 : Select McASP1 audio clock to be SYSCLK21<br>0x2 = SEL3 : Select McASP1 audio clock to be SYSCLK22<br>0x3 = Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.16 CM_AUDIOCLK_MCASP2_CLKSEL Register (offset = 84h) [reset = 0h]

CM_AUDIOCLK_MCASP2_CLKSEL is shown in Figure 2-199 and described in Table 2-219.

Selects the Mux select line for McASP2 audio clock

**Figure 2-199. CM_AUDIOCLK_MCASP2_CLKSEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | Reserved | CLKSEL | |
| R-0h | | | | | R-0h | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-219. CM_AUDIOCLK_MCASP2_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2 | Reserved | R | 0h | |
| 1-0 | CLKSEL | R/W | 0h | Selects the Mux select line for McASP2 audio clock [warm reset insensitive]<br>0x0 = SEL1 : Select McASP2 audio clock to be SYSCLK20<br>0x1 = SEL2 : Select McASP2 audio clock to be SYSCLK21<br>0x2 = SEL3 : Select McASP2 audio clock to be SYSCLK22<br>0x3 = Reserved |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.17 CM_AUDIOCLK_MCBSP_CLKSEL Register (offset = 88h) [reset = 0h]

CM_AUDIOCLK_MCBSP_CLKSEL is shown in Figure 2-200 and described in Table 2-220.

Selects the Mux select line for McBSP audio clock

**Figure 2-200. CM_AUDIOCLK_MCBSP_CLKSEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | Reserved | CLKSEL | |
| R-0h | | | | | R-0h | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-220. CM_AUDIOCLK_MCBSP_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2 | Reserved | R | 0h | |
| 1-0 | CLKSEL | R/W | 0h | Selects the Mux select line for McBSP audio clock [warm reset insensitive]<br>0x0 = SEL1 : Select McBSP audio clock to be SYSCLK20<br>0x1 = SEL2 : Select McBSP audio clock to be SYSCLK21<br>0x2 = SEL3 : Select McBSP audio clock to be SYSCLK22<br>0x3 = SEL4 : RESERVED. This input of the MCBSP_CLKSEL mux is tied-off to '0' (ground). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.18 CM_HDMI_CLKSEL Register (offset = ACh) [reset = 2h]

CM_HDMI_CLKSEL is shown in Figure 2-201 and described in Table 2-221.

Selects the Mux select line for HDMI audio clock

**Figure 2-201. CM_HDMI_CLKSEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | Reserved | CLKSEL | |
| R-0h | | | | | R-0h | R/W-2h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-221. CM_HDMI_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2 | Reserved | R | 0h | |
| 1-0 | CLKSEL | R/W | 2h | Selects the Mux select line for HDMI audio clock [warm reset insensitive]<br>0x0 = SEL1 : Select HDMI audio clock to be SYSCLK20<br>0x1 = SEL2 : Select HDMI audio clock to be SYSCLK21<br>0x2 = SEL3 : Select HDMI audio clock to be SYSCLK22<br>0x3 = Reserved |



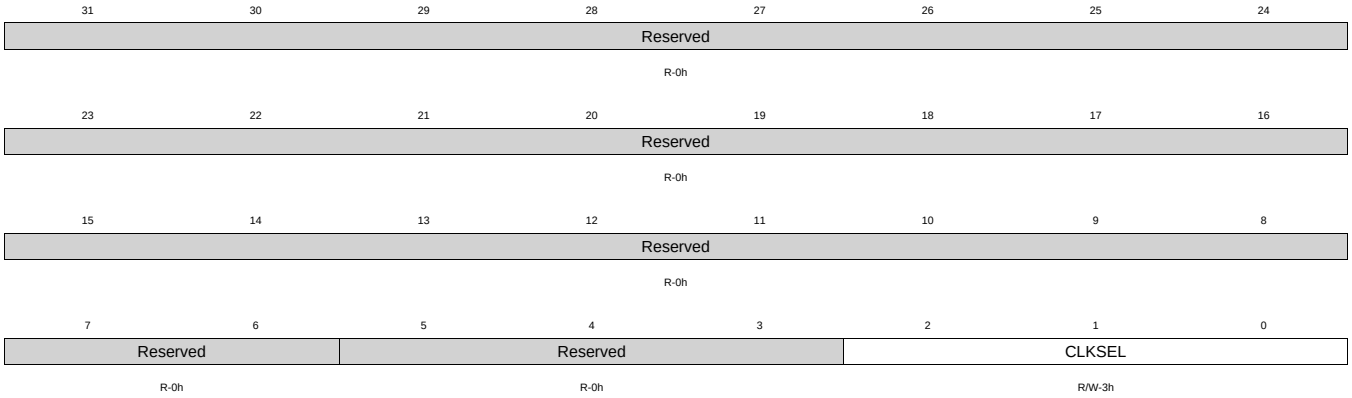
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.5.19 CM_SYSCLK23_CLKSEL Register (offset = B0h) [reset = 3h]

CM_SYSCLK23_CLKSEL is shown in Figure 2-202 and described in Table 2-222.

Selects the divider value for SYSCLK23 which is the SGX functional clock

**Figure 2-202. CM_SYSCLK23_CLKSEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-222. CM_SYSCLK23_CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-3 | Reserved | R | 0h | |
| 2-0 | CLKSEL | R/W | 3h | Selects the divider value [warm reset insensitive]<br>0x0 = SYSCLK_DIV_1 : Select SYS_CLK divided by 1<br>0x1 = SYSCLK_DIV_2 : Select SYS_CLK divided by 2<br>0x2 = SYSCLK_DIV_3 : Select SYS_CLK divided by 3<br>0x3 = SYSCLK_DIV_4 : Select SYS_CLK divided by 4<br>0x4 = SYSCLK_DIV_5 : Select SYS_CLK divided by 5<br>0x5 = SYSCLK_DIV_6 : Select SYS_CLK divided by 6<br>0x6 = SYSCLK_DIV_7 : Select SYS_CLK divided by 7<br>0x7 = SYSCLK_DIV_8 : Select SYS_CLK divided by 8 |

## *2.10.6 CM_ACTIVE Registers*

Table 2-223 lists the memory-mapped registers for the CM_ACTIVE. All register offset addresses not listed in Table 2-223 should be considered as reserved locations and the register contents should not be modified.

**Table 2-223. CM_ACTIVE REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | CM_GEM_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states.<br>It also hold one status bit per clock input of the domain. | Section 2.10.6.1 |
| 20h | CM_ACTIVE_GEM_CLKCTRL | This register manages the DSP Subsystem C674x clocks. | Section 2.10.6.2 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.6.1 CM_GEM_CLKSTCTRL Register (offset = 0h) [reset = 1h]

CM_GEM_CLKSTCTRL is shown in Figure 2-203 and described in Table 2-224.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-203. CM_GEM_CLKSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | Reserved | |
| R-0h | | | | | | R-0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | Reserved | |
| R-0h | | | | | | R-0h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKTRCTRL | |
| R-0h | | | | | | R/W-1h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-224. CM_GEM_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-11 | Reserved | R | 0h | |
| 10-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the GEM clock domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.6.2 CM_ACTIVE_GEM_CLKCTRL Register (offset = 20h) [reset = 70000h]

CM_ACTIVE_GEM_CLKCTRL is shown in Figure 2-204 and described in Table 2-225.

This register manages the DSP subsystem C674x clocks.

**Figure 2-204. CM_ACTIVE_GEM_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | STBYST | IDLEST | |
| R-0h | | | | | R-1h | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-225. CM_ACTIVE_GEM_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including OCP<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>Read 0x1: Reserved<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>Read 0x3: Reserved |

## *2.10.7 CM_DEFAULT Registers*

Table 2-226 lists the memory-mapped registers for the CM_DEFAULT. All register offset addresses not listed in Table 2-226 should be considered as reserved locations and the register contents should not be modified.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-226. CM_DEFAULT REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 10h | CM_DEFAULT_PCI_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.7.1 |
| 18h | CM_DEFAULT_DUCATI_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.7.2 |
| 28h | CM_DEFAULT_DMM_CLKCTRL | This register manages the DMM clocks. | Section 2.10.7.3 |
| 54h | Reserved | | |
| 58h | CM_DEFAULT_USB_CLKCTRL | This register manages the USB clocks. | Section 2.10.7.4 |
| 60h | CM_DEFAULT_SATA_CLKCTRL | This register manages the SATA clocks. | Section 21.6.2 |
| 74h | CM_DEFAULT_DUCATI_CLKCTRL | This register manages the DUCATI clocks. | Section 2.10.7.6 |
| 78h | CM_DEFAULT_PCI_CLKCTRL | This register manages the PCI clocks. | Section 2.10.7.7 |

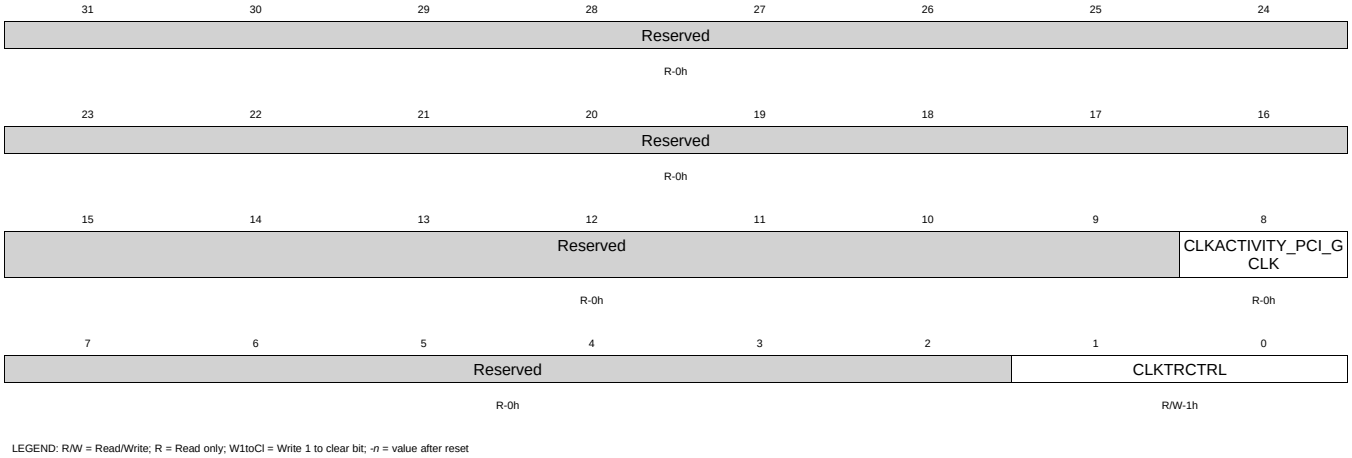

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.7.1 CM_DEFAULT_PCI_CLKSTCTRL Register (offset = 10h) [reset = 1h]

CM_DEFAULT_PCI_CLKSTCTRL is shown in Figure 2-206 and described in Table 2-228.

This register manages the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-205. CM_DEFAULT_PCI_CLKSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-227. CM_DEFAULT_PCI_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | Reserved | R | 0h | |
| 8 | CLKACTIVITY_PCI_GCLK | R | 0h | This field indicates the state of the PCI_GCLK clock in the domain.<br>Read 0x0: Corresponding clock is gated.<br>Read 0x1: Corresponding clock is active. |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the PCI clock domains.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.7.2 CM_DEFAULT_DUCATI_CLKSTCTRL Register (offset = 18h) [reset = 1h]

CM_DEFAULT_DUCATI_CLKSTCTRL is shown in Figure 2-206 and described in Table 2-228.

This register manages the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-206. CM_DEFAULT_DUCATI_CLKSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKACTIVITY_DUCATI_GCLKIN200TR | CLKACTIVITY_DUCATI_GCLKINTR |
| R-0h | | | | | | R-0h | R-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKTRCTRL | |
| R-0h | | | | | | R/W-1h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-228. CM_DEFAULT_DUCATI_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | Reserved | R | 0h | |
| 9 | CLKACTIVITY_DUCATI_GCLKIN200TR | R | 0h | This field indicates the state of the CLKIN200TR clock in the domain<br>Read 0: Corresponding clock is gated<br>Read 1: Corresponding clock is active |
| 8 | CLKACTIVITY_DUCATI_GCLKINTR | R | 0h | This field indicates the state of the CLKINTR clock in the domain.<br>Read 0: Corresponding clock is gated<br>Read 1: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the DUCATI clock domains.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |

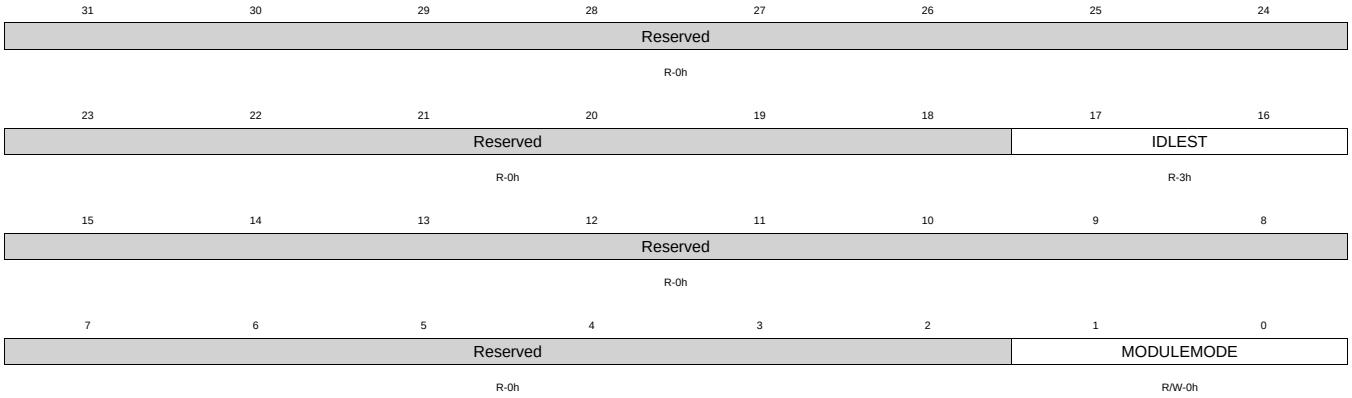

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.7.3 CM_DEFAULT_DMM_CLKCTRL Register (offset = 28h) [reset = 30000h]

CM_DEFAULT_DMM_CLKCTRL is shown in Figure 2-207 and described in Table 2-229.

This register manages the DMM clocks.

**Figure 2-207. CM_DEFAULT_DMM_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-229. CM_DEFAULT_DMM_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

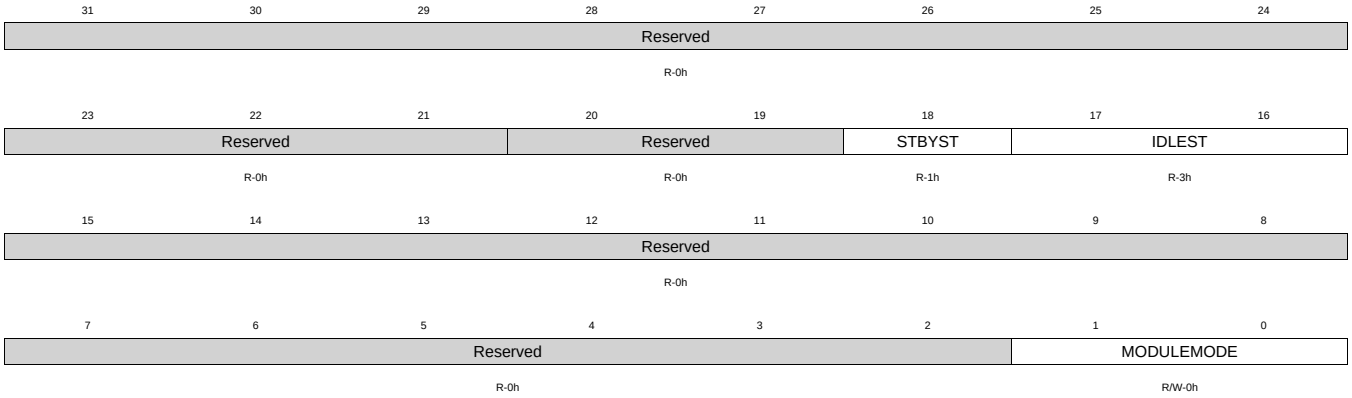

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.7.4 CM_DEFAULT_USB_CLKCTRL Register (offset = 58h) [reset = 70000h]

CM_DEFAULT_USB_CLKCTRL is shown in Figure 2-208 and described in Table 2-230.

This register manages the USB clocks.

**Figure 2-208. CM_DEFAULT_USB_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-230. CM_DEFAULT_USB_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

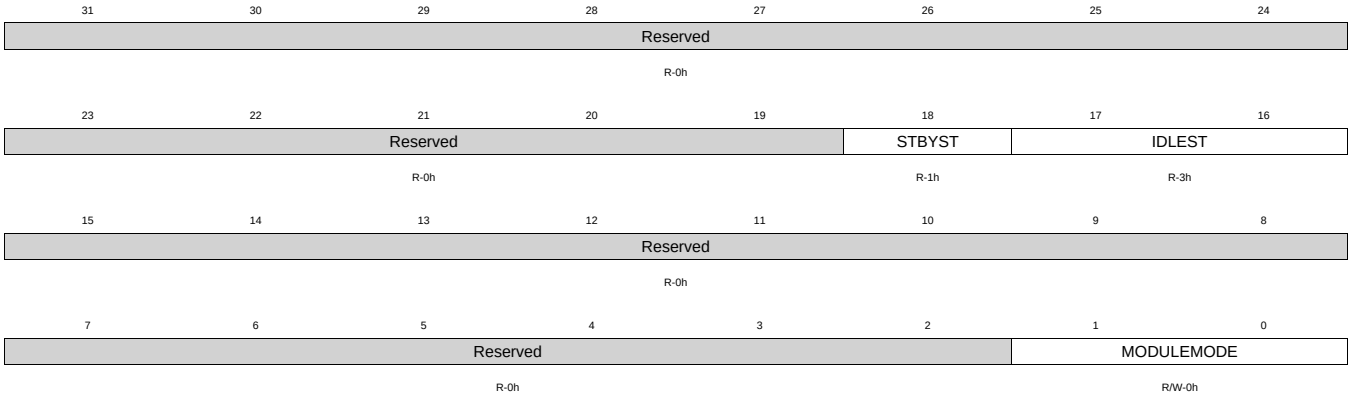

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.7.5 CM_DEFAULT_SATA_CLKCTRL Register (offset = 60h) [reset = 70000h]

CM_DEFAULT_SATA_CLKCTRL is shown in Figure 2-209 and described in Table 2-231.

This register manages the SATA clocks.

**Figure 2-209. CM_DEFAULT_SATA_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-231. CM_DEFAULT_SATA_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.7.6 CM_DEFAULT_DUCATI_CLKCTRL Register (offset = 74h) [reset = 70000h]

CM_DEFAULT_DUCATI_CLKCTRL is shown in Figure 2-209 and described in Table 2-231.

This register manages the DUCATI clocks.

**Figure 2-210. CM_DEFAULT_DUCATI_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | STBYST | IDLEST | |
| R-0h | | | | | R-1h | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-232. CM_DEFAULT_DUCATI_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

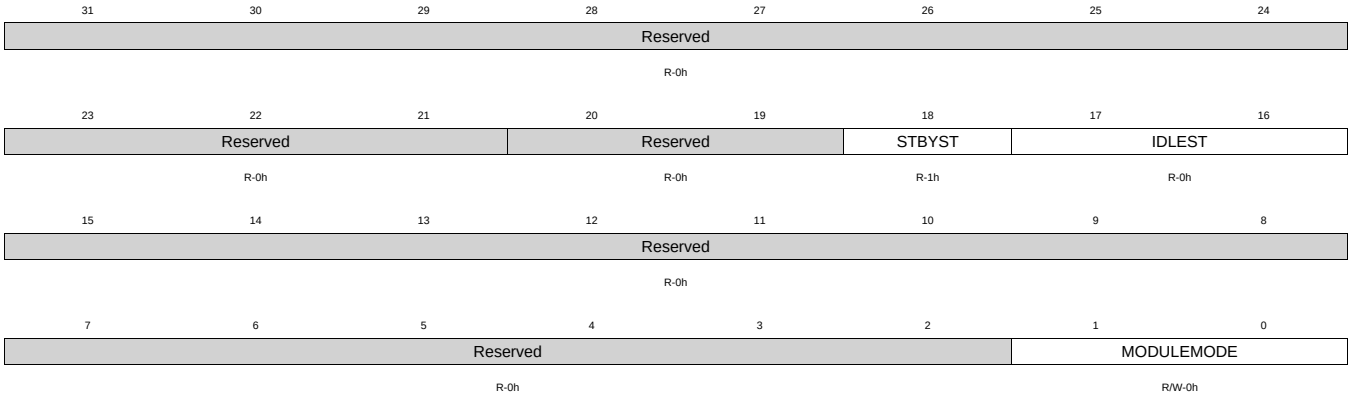

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.7.7 CM_DEFAULT_PCI_CLKCTRL Register (offset = 78h) [reset = 40000h]

CM_DEFAULT_PCI_CLKCTRL is shown in Figure 2-211 and described in Table 2-233.

This register manages the PCI clocks.

**Figure 2-211. CM_DEFAULT_PCI_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-233. CM_DEFAULT_PCI_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 0h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3 RESERVED |

## 2.10.8 CM_HDVICP Registers

Table 2-234 lists the memory-mapped registers for the CM_HDVICP. All register offset addresses not listed in Table 2-234 should be considered as reserved locations and the register contents should not be modified.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-234. CM_HDVICP REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | CM_HDVICP_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.8.1 |
| 20h | CM_HDVICP_CLKCTRL | This register manages the HDVICP clocks. | Section 2.10.8.2 |
| 24h | CM_HDVICP_SL2_CLKCTRL | This register manages the SL2 clocks. | Section 2.10.8.3 |

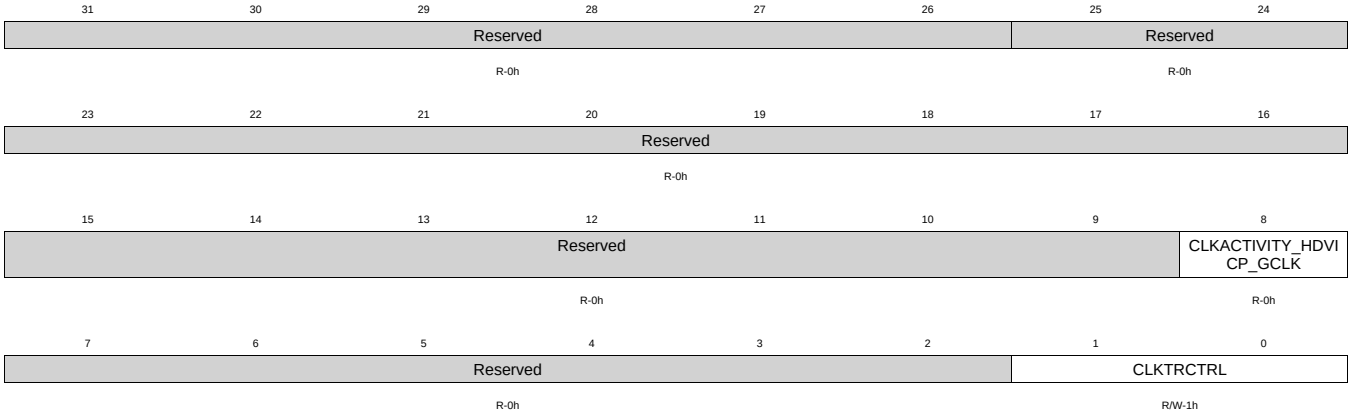

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.8.1 CM_HDVICP_CLKSTCTRL Register (offset = 0h) [reset = 1h]

CM_HDVICP_CLKSTCTRL is shown in Figure 2-212 and described in Table 2-235.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-212. CM_HDVICP_CLKSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-235. CM_HDVICP_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | CLKACTIVITY_HDVICP_GCLK | R | 0h | This field indicates the state of the HDVICP_GCLK clock in the domain.<br>0x0: Inact : Corresponding clock is gated<br>0x1: Act : Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the HDVICP clock domain in HDVICP power domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.8.2 CM_HDVICP_CLKCTRL Register (offset = 20h) [reset = 70000h]

CM_HDVICP_CLKCTRL is shown in Figure 2-213 and described in Table 2-236.

This register manages the HDVICP clocks.

**Figure 2-213. CM_HDVICP_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | STBYST | IDLEST | |
| R-0h | | | | | R-1h | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-236. CM_HDVICP_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |



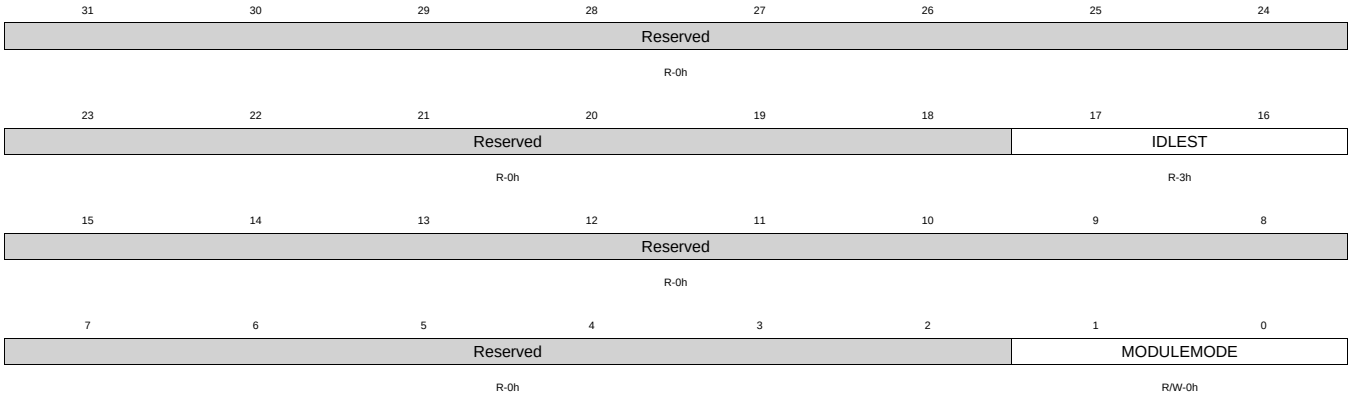
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.8.3 CM_HDVICP_SL2_CLKCTRL Register (offset = 24h) [reset = 30000h]

CM_HDVICP_SL2_CLKCTRL is shown in Figure 2-214 and described in Table 2-237.

This register manages the SL2 clocks.

**Figure 2-214. CM_HDVICP_SL2_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-237. CM_HDVICP_SL2_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func : Module is fully functional, including INTERCONN<br>0x1: Trans : Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle : Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable : Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

## 2.10.9 CM_ISP Registers

Table 2-238 lists the memory-mapped registers for the CM_ISP. All register offset addresses not listed in Table 2-238 should be considered as reserved locations and the register contents should not be modified.

**Table 2-238. CM_ISP REGISTERS**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 0h | CM_ISP_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.9.1 |
| 20h | CM_ISP_ISP_CLKCTRL | This register manages the ISP clocks. | Section 2.10.9.2 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-238. CM_ISP REGISTERS (continued)**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 24h | CM_ISP_FDIF_CLKCTRL | This register manages the FDIF clocks. | Section 2.10.9.3 |

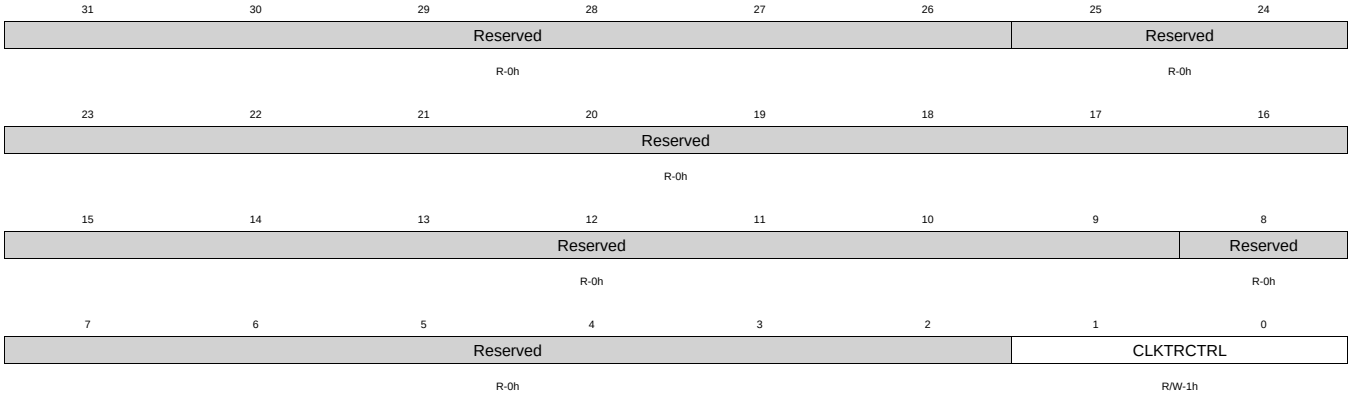

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.9.1 CM_ISP_CLKSTCTRL Register (offset = 0h) [reset = 1h]

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_ISP_CLKSTCTRL is shown in Figure 2-215 and described in Table 2-239.

**Figure 2-215. CM_ISP_CLKSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-239. CM_ISP_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | Reserved | R | 0h | |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the ISP clock domain in ISP power domain.<br>0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain.<br>2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain.<br>3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |

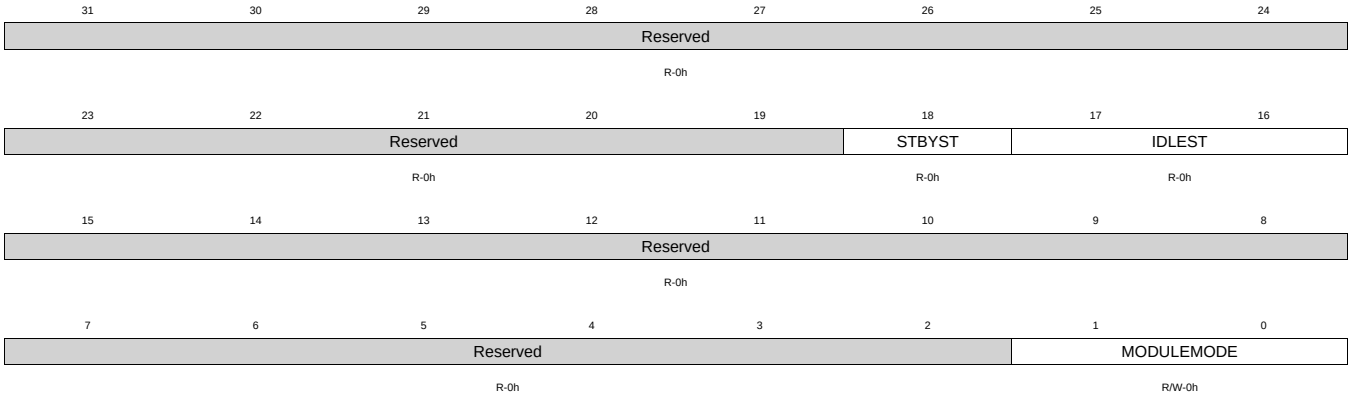

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.9.2 CM_ISP_ISP_CLKCTRL Register (offset = 20h) [reset = 0h]

This register manages the ISP clocks.

CM_ISP_ISP_CLKCTRL is shown in Figure 2-216 and described in Table 2-240.

**Figure 2-216. CM_ISP_ISP_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-240. CM_ISP_ISP_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 0h | Module standby status.<br>0 - Func: Module is functional (not in standby)<br>1 - Standby: Module is in standby |
| 17-16 | IDLEST | R | 0h | Module idle status.<br>0 - Func: Module is fully functional, including OCP<br>1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock<br>3 - Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>1 - RESERVED_1: Reserved<br>2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>3 - RESERVED: Reserved |



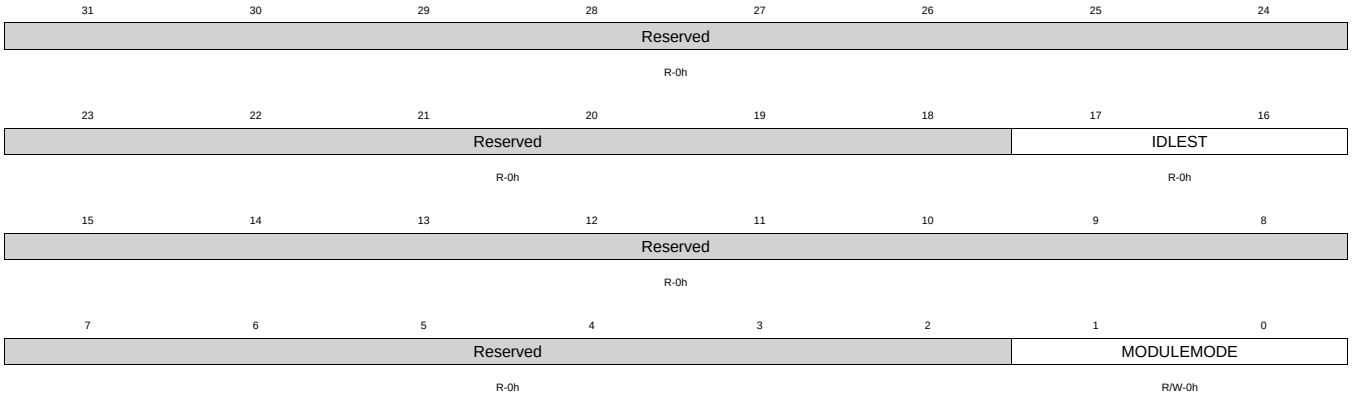
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.9.3 CM_ISP_FDIF_CLKCTRL Register (offset = 24h) [reset = 0h]

This register manages the FDIF clocks.

CM_ISP_FDIF_CLKCTRL is shown in Figure 2-217 and described in Table 2-241.

**Figure 2-217. CM_ISP_FDIF_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-241. CM_ISP_FDIF_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 0h | Module idle status.<br>0 - Func: Module is fully functional, including OCP<br>1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock<br>3 - Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>1 - RESERVED_1: Reserved<br>2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>3 - RESERVED: Reserved |

## *2.10.10 CM_DSS Registers*

Table 2-242 lists the memory-mapped registers for the CM_DSS. All register offset addresses not listed in Table 2-242 should be considered as reserved locations and the register contents should not be modified.

**Table 2-242. CM_DSS REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | CM_DSS_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states.<br>It also hold one status bit per clock input of the domain. | Section 2.10.10.1 |
| 20h | CM_DSS_DSS_CLKCTRL | This register manages the DSS clocks. | Section 2.10.10.2 |
| 24h | CM_DSS_HDMI_CLKCTRL | This register manages the HDMI clocks. | Section 2.10.10.3 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.10.1 CM_DSS_CLKSTCTRL Register (offset = 0h) [reset = 1h]

CM_DSS_CLKSTCTRL is shown in Figure 2-218 and described in Table 2-243.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-218. CM_DSS_CLKSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | Reserved | |
| R-0h | | | | | | R-0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | Reserved |
| R-0h | | | | | | | R-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKTRCTRL | |
| R-0h | | | | | | R/W-1h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-243. CM_DSS_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | Reserved | R | 0h | |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the DSS clock domain in DSS power domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |

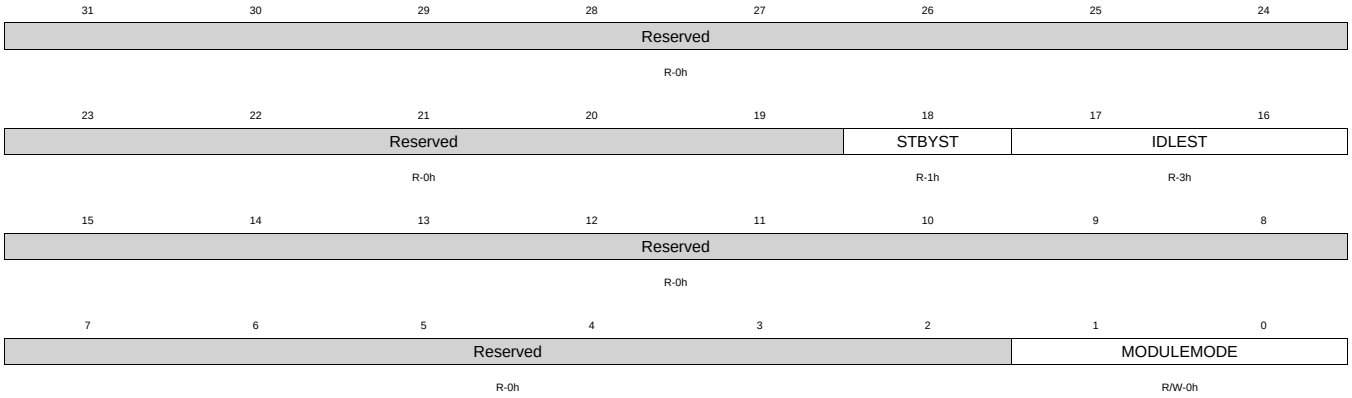

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.10.2 CM_DSS_DSS_CLKCTRL Register (offset = 20h) [reset = 70000h]

CM_DSS_DSS_CLKCTRL is shown in Figure 2-219 and described in Table 2-244.

This register manages the DSS clocks.

**Figure 2-219. CM_DSS_DSS_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-244. CM_DSS_DSS_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0 Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

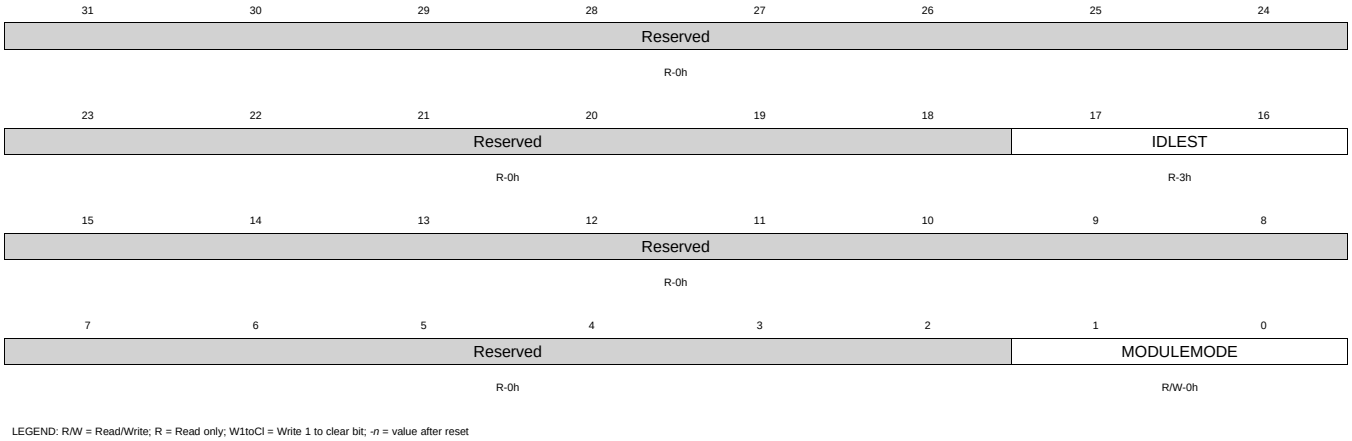

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.10.3 CM_DSS_HDMI_CLKCTRL Register (offset = 24h) [reset = 30000h]

CM_DSS_HDMI_CLKCTRL is shown in Figure 2-220 and described in Table 2-245.

This register manages the HDMI clocks.

**Figure 2-220. CM_DSS_HDMI_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-245. CM_DSS_HDMI_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

## *2.10.11 CM_SGX Registers*

Table 2-246 lists the memory-mapped registers for the CM_SGX. All register offset addresses not listed in Table 2-246 should be considered as reserved locations and the register contents should not be modified.

**Table 2-246. CM_SGX REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | CM_SGX_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states.<br>It also hold one status bit per clock input of the domain. | Section 2.10.11.1 |
| 20h | CM_SGX_SGX_CLKCTRL | This register manages the SGX clocks. | Section 2.10.11.2 |



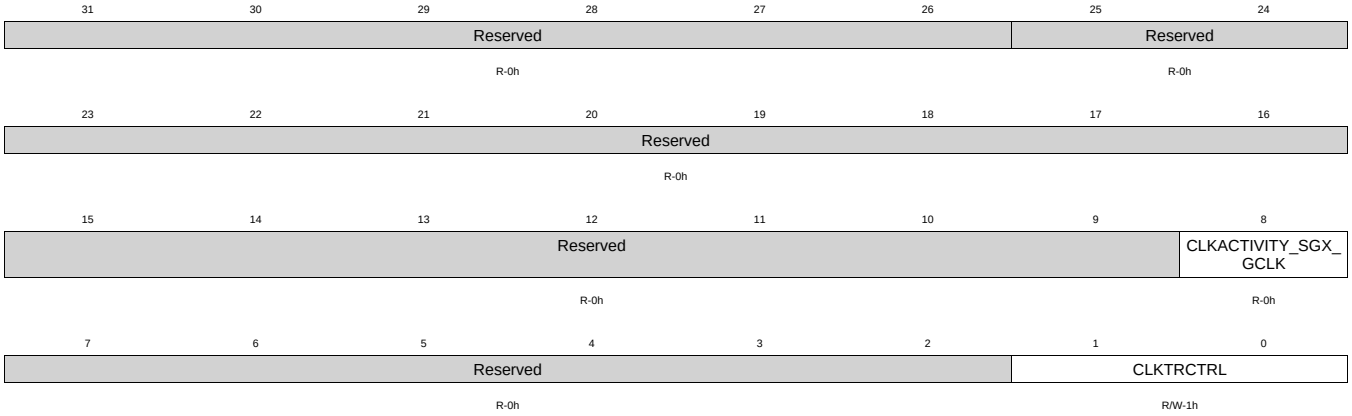
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.11.1 CM_SGX_CLKSTCTRL Register (offset = 0h) [reset = 1h]

CM_SGX_CLKSTCTRL is shown in Figure 2-221 and described in Table 2-247.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-221. CM_SGX_CLKSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-247. CM_SGX_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | CLKACTIVITY_SGX_GCLK | R | 0h | This field indicates the state of the SGX_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the SGX clock domain in SGX power domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.11.2 CM_SGX_SGX_CLKCTRL Register (offset = 20h) [reset = 70000h]

CM_SGX_SGX_CLKCTRL is shown in Figure 2-222 and described in Table 2-248.

This register manages the SGX clocks.

**Figure 2-222. CM_SGX_SGX_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | STBYST | IDLEST | |
| R-0h | | | | | R-1h | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-248. CM_SGX_SGX_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func : Module is functional (not in standby)<br>0x1: Standby : Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

## *2.10.12 PRM_ACTIVE Registers*

Table 2-249 lists the memory-mapped registers for the PRM_ACTIVE. All register offset addresses not listed in Table 2-249 should be considered as reserved locations and the register contents should not be modified.

The ACTIVE domain contains the DSP logic and DSP memories.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-249. PRM_ACTIVE REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | PM_ACTIVE_PWRSTCTRL | This register controls the ACTIVE power state to reach upon a domain sleep transition [warm reset insensitive] | Section 2.10.12.1 |
| 4h | PM_ACTIVE_PWRSTST | This register provides a status on the current ACTIVE power domain state.<br>[warm reset insensitive] | Section 2.10.12.2 |
| 10h | RM_ACTIVE_RSTCTRL | This register controls the release of the ACTIVE Domain resets. | Section 2.10.12.3 |
| 14h | RM_ACTIVE_RSTST | This register logs the different reset sources of the ACTIVE domain.<br>Each bit is set upon release of the domain reset signal. Must be cleared by software.<br>[warm reset insensitive] | Section 2.10.12.4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.12.1 PM_ACTIVE_PWRSTCTRL Register (offset = 0h) [reset = 30000h]

PM_ACTIVE_PWRSTCTRL is shown in Figure 2-223 and described in Table 2-250.

This register controls the ACTIVE power state to reach upon a domain sleep transition [warm reset insensitive]

**Figure 2-223. PM_ACTIVE_PWRSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | Active_MEM_ONState | |
| R-0h | | | | | | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | LowPowerStateChange | Reserved | | PowerState | |
| R-0h | | | R/W-0h | R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-250. PM_ACTIVE_PWRSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | Active_MEM_ONState | R | 3h | Active Domain memory state when domain is ON. |
| 15-5 | Reserved | R | 0h | |
| 4 | LowPowerStateChange | R/W | 0h | Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain.<br>0x0: DIS: Do not request a low power state change.<br>0x1: EN: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON. |
| 3-2 | Reserved | R | 0h | |
| 1-0 | PowerState | R/W | 0h | Power state control<br>0x0: OFF: OFF State [warm reset insensitive]<br>0x1: Reserved<br>0x2: Reserved<br>0x3: ON: ON State [warm reset insensitive] |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.12.2 PM_ACTIVE_PWRSTST Register (offset = 4h) [reset = 17h]

PM_ACTIVE_PWRSTST is shown in Figure 2-224 and described in Table 2-251.

This register provides a status on the current ACTIVE power domain state. [warm reset insensitive]

**Figure 2-224. PM_ACTIVE_PWRSTST Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | InTransition | Reserved | | | |
| R-0h | | | R-0h | R-0h | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | Active_MEM_StateSt | | Reserved | LogicStateSt | PowerStateSt | |
| R-0h | | R-1h | | R-0h | R-1h | R-3h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-251. PM_ACTIVE_PWRSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20 | InTransition | R | 0h | Domain transition status<br>0x0: No: No on-going transition on power domain<br>0x1: Ongoing: Power domain transition is in progress. |
| 19-6 | Reserved | R | 0h | |
| 5-4 | Active_MEM_StateSt | R | 1h | Active domain memory state status<br>0x0: Mem_off : Memory is OFF<br>0x2: Reserved : Reserved<br>0x3: Mem_on : Memory is ON |
| 3 | Reserved | R | 0h | |
| 2 | LogicStateSt | R | 1h | Logic state status<br>0x0: OFF : Logic in domain is OFF<br>0x1: ON : Logic in domain is ON |
| 1-0 | PowerStateSt | R | 3h | Current Power State Status<br>0x0: OFF : OFF State<br>0x1: Reserved<br>0x2: Reserved1<br>0x3: ON : ON State |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.12.3 RM_ACTIVE_RSTCTRL Register (offset = 10h) [reset = 3h]

RM_ACTIVE_RSTCTRL is shown in Figure 2-225 and described in Table 2-252.

This register controls the release of the ACTIVE Domain resets.

**Figure 2-225. RM_ACTIVE_RSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | | | | | GEM_SW_RST | GEM_LRST |
| R-0h | R-0h | | | | | R/W-1h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-252. RM_ACTIVE_RSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | Reserved | R | 0h | |
| 6-2 | Reserved | R | 0h | |
| 1 | GEM_SW_RST | R/W | 1h | ACTIVE Domain DSP warm reset control<br>0x0: CLEAR : Reset is cleared for the Active domain DSP warm Reset<br>0x1: ASSERT : Reset is asserted for the Active domain DSP warm reset |
| 0 | GEM_LRST | R/W | 1h | ACTIVE Domain DSP local reset control<br>0x0: CLEAR : Reset is cleared for the Active domain DSP local Reset<br>0x1: ASSERT : Reset is asserted for the Active domain DSP Local reset |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.12.4 RM_ACTIVE_RSTST Register (offset = 14h) [reset = 0h]

RM_ACTIVE_RSTST is shown in Figure 2-226 and described in Table 2-253.

This register logs the different reset sources of the ACTIVE domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

**Figure 2-226. RM_ACTIVE_RSTST Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | EMULATION_GEM_RST | GEM_LRST_REQ | Reserved | GEM_GRST | GEM_LRST |
| R-0h | | | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-253. RM_ACTIVE_RSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | Reserved | R | 0h | |
| 4 | EMULATION_GEM_RST | R/W | 0h | DSP Local CPU has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module<br>0x0: RESET_NO : No emulation reset<br>0x1: RESET_YES : Local ARM has been reset upon emulation reset |
| 3 | GEM_LRST_REQ | R/W | 0h | Local Reser request generated by DSP( Because of Debug) |
| 2 | Reserved | R | 0h | |
| 1 | GEM_GRST | R/W | 0h | DSP SS warm reset<br>0x0: RESET_NO: No SW reset occured<br>0x1: RESET_YES: DSP warm reset has been asserted. |
| 0 | GEM_LRST | R/W | 0h | DSP local SW reset<br>0x0: RESET_NO: No SW reset occured<br>0x1: RESET_YES: DSP Local reset has been asserted |

## *2.10.13 PRM_DEFAULT Registers*

Table 2-254 lists the memory-mapped registers for the PRM_DEFAULT. All register offset addresses not listed in Table 2-254 should be considered as reserved locations and the register contents should not be modified.

**Table 2-254. PRM_DEFAULT REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | PM_DEFAULT_PWRSTCTRL | This register controls the DEFAULT power state to reach upon a domain sleep transition [warm reset insensitive]. | Section 2.10.13.1 |
| 10h | RM_DEFAULT_RSTCTRL | This register controls the release of the DEFAULT sub-system resets. | Section 2.10.13.2 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-254. PRM_DEFAULT REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 14h | RM_DEFAULT_RSTST | This register logs the different reset sources of the DEFAULT domain.<br>Each bit is set upon release of the domain reset signal.<br>Must be cleared by software.<br>[warm reset insensitive] | Section 2.10.13.3 |

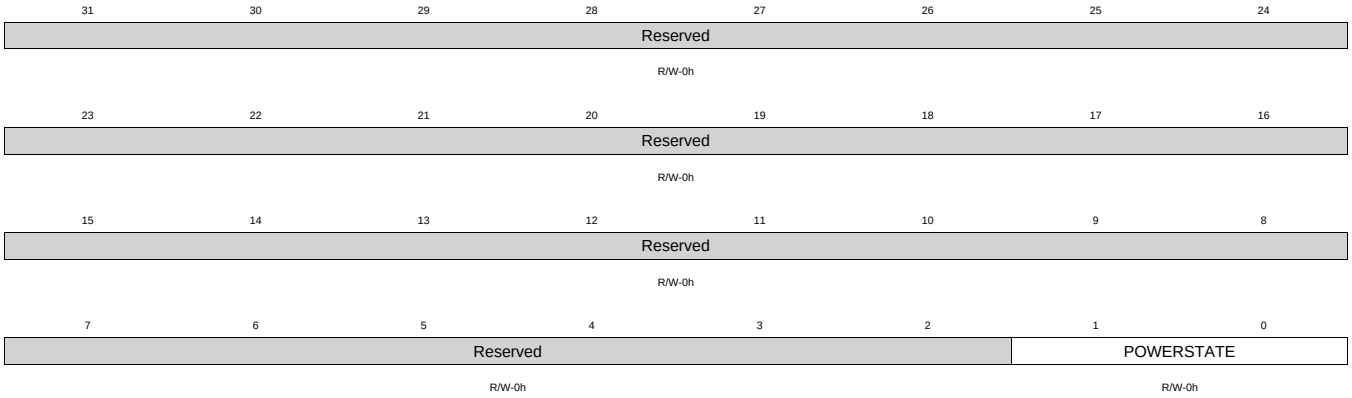

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.13.1 PM_DEFAULT_PWRSTCTRL Register (offset = 0h) [reset = 0h]

PM_DEFAULT_PWRSTCTRL is shown in Figure 2-227 and described in Table 2-255.

This register controls the DEFAULT power state to reach upon a domain sleep transition [warm reset insensitive].

**Figure 2-227. PM_DEFAULT_PWRSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-255. PM_DEFAULT_PWRSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R/W | 0h | |
| 1-0 | POWERSTATE | R/W | 0h | Power state control:<br>0x0: OFF State<br>0x3: ON State: This POWERSTATE field of this register (bit 1:0) needs to be written to ON state (0x3 value) and keep it there. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.13.2 RM_DEFAULT_RSTCTRL Register (offset = 10h) [reset = 9Fh]

RM_DEFAULT_RSTCTRL is shown in Figure 2-228 and described in Table 2-256.

This register controls the release of the DEFAULT sub-system resets.

**Figure 2-228. RM_DEFAULT_RSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCI_LRST | USB2_LRST | USB1_LRST | DUCATI_RST3 | DUCATI_M3_RST2 | DUCATI_M3_RST1 | Reserved | |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-256. RM_DEFAULT_RSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | Reserved | R | 0h | |
| 7 | PCI_LRST | R/W | 1h | ACTIVE domain PCI Local reset control<br>0x0: CLEAR : Reset is cleared.<br>0x1: ASSERT : Reset is asserted. |
| 6 | USB2_LRST | R/W | 1h | USB2 local reset control<br>0x0: Local Reset is cleared for USB2.<br>0x1: Local Reset is asserted for USB2. |
| 5 | USB1_LRST | R/W | 1h | USB1 local reset control<br>0x0: Local Reset is cleared for USB1.<br>0x1: Local Reset is asserted for USB1. |
| 4 | DUCATI_RST3 | R/W | 1h | Media Controller logic and MMU reset control<br>0x0: CLEAR : Reset is cleared for the Media Controller logic and MMU<br>0x1: ASSERT : Reset is asserted for the Media Controller logic and MMU |
| 3 | DUCATI_M3_RST2 | R/W | 1h | Media Controller Second M3 reset control<br>0x0: CLEAR : Reset is cleared for the ALWON sequencer CPU2<br>0x1: ASSERT : Reset is asserted for the ALWON sequencer CPU2 |
| 2 | DUCATI_M3_RST1 | R/W | 1h | Media Controller First M3 reset control<br>0x0: CLEAR : Reset is cleared for the ALWON sequencer CPU1<br>0x1: ASSERT : Reset is asserted for the ALWON sequencer CPU1 |
| 1-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.13.3 RM_DEFAULT_RSTST Register (offset = 14h) [reset = 0h]

RM_DEFAULT_RSTST is shown in Figure 2-229 and described in Table 2-257.

This register logs the different reset sources of the DEFAULT domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

**Figure 2-229. RM_DEFAULT_RSTST Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | Reserved | | | | |
| R-0h | | | R-0h | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | ICECRUSHER_m3_2_RST | ICECRUSHER_M3_1_RST | EMULATION_M3_2_RST | EMULATION_M3_1_RST | Reserved | | |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PCI_LRST | USB2_LRST | USB1_LRST | DUCATI_RST3 | DUCATI_M3_RST2 | DUCATI_M3_RST1 | Reserved | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-257. RM_DEFAULT_RSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20-15 | Reserved | R | 0h | |
| 14 | ICECRUSHER_m3_2_RST | R/W | 0h | DSS controller M3_2 has been reset due to DSS controller ICECRUSHER1 reset event<br>0x0: RESET_NO : No icecrusher reset<br>0x1: RESET_YES : Sequencer2 has been reset upon icecrusher reset |
| 13 | ICECRUSHER_M3_1_RST | R/W | 0h | DSS controller M3_1 has been reset due to DSS controller ICECRUSHER1 reset event<br>0x0: RESET_NO : No icecrusher reset<br>0x1: RESET_YES : Sequencer1 has been reset upon icecrusher reset |
| 12 | EMULATION_M3_2_RST | R/W | 0h | DSS controller M3_2 has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module<br>0x0: RESET_NO : No emulation reset<br>0x1: RESET_YES : M3_2 has been reset upon emulation reset |
| 11 | EMULATION_M3_1_RST | R/W | 0h | DSS controller M3_1 has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module<br>0x0: RESET_NO : No emulation reset<br>0x1: RESET_YES : M3_1 has been reset upon emulation reset |
| 10-8 | Reserved | R | 0h | |
| 7 | PCI_LRST | R/W | 0h | PCI local SW reset<br>0x0: RESET_NO : No SW reset occured<br>0x1: RESET_YES : PCI Local M3 has been reset upon SW reset |
| 6 | USB2_LRST | R/W | 0h | USB2 Local Reset<br>0x0: RESET_NO: No SW reset occured<br>0x1: RESET_YES: USB2 Local Logic has been reset upon SW reset |
| 5 | USB1_LRST | R/W | 0h | USB1 Local Reset<br>0x0: RESET_NO: No SW reset occured<br>0x1: RESET_YES: USB1 Local Logic has been reset upon SW reset |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-257. RM_DEFAULT_RSTST Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 4 | DUCATI_RST3 | R/W | 0h | Media Controller logic and MMU SW reset<br>0x0: RESET_NO : No SW reset occured<br>0x1: RESET_YES : Media Controller logic and MMU has been reset upon SW reset |
| 3 | DUCATI_M3_RST2 | R/W | 0h | Media Controller second M3 SW reset<br>0x0: RESET_NO : No SW reset occured<br>0x1: RESET_YES : Cortex M3_2 has been reset upon SW reset |
| 2 | DUCATI_M3_RST1 | R/W | 0h | Media Controller first cortex M3 SW reset<br>0x0: RESET_NO : No SW reset occured<br>0x1: RESET_YES : Cortex M3_1 has been reset upon SW reset |
| 1-0 | Reserved | R | 0h | |

### 2.10.14 PRM_HDVICP Registers

Table 2-258 lists the memory-mapped registers for the PRM_HDVICP. All register offset addresses not listed in Table 2-258 should be considered as reserved locations and the register contents should not be modified.

**Table 2-258. PRM_HDVICP REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | PM_HDVICP_PWRSTCTRL | This register controls the HDVICP power state to reach upon a domain sleep transition [warm reset insensitive] | Section 2.10.14.1 |
| 4h | PM_HDVICP_PWRSTST | This register provides a status on the current HDVICP power domain state.<br>[warm reset insensitive] | Section 2.10.14.2 |
| 10h | RM_HDVICP_RSTCTRL | This register controls the release of the HDVICP sub-system resets. | Section 2.10.14.3 |
| 14h | RM_HDVICP_RSTST | This register logs the different reset sources of the HDVICP domain.<br>Each bit is set upon release of the domain reset signal.<br>Must be cleared by software.<br>[warm reset insensitive] | Section 2.10.14.4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.14.1 PM_HDVICP_PWRSTCTRL Register (offset = 0h) [reset = 30000h]

PM_HDVICP_PWRSTCTRL is shown in Figure 2-230 and described in Table 2-259.

This register controls the HDVICP power state to reach upon a domain sleep transition [warm reset insensitive]

**Figure 2-230. PM_HDVICP_PWRSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | HDVICP_MEM_ONState | |
| R-0h | | | | | | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | LowPowerStateChange | Reserved | | PowerState | |
| R-0h | | | R/W-0h | R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-259. PM_HDVICP_PWRSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | HDVICP_MEM_ONState | R | 3h | HDVICP memory state when domain is ON. |
| 15-5 | Reserved | R | 0h | |
| 4 | LowPowerStateChange | R/W | 0h | Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain.<br>0x0: DIS : Do not request a low power state change.<br>0x1: EN : Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON. |
| 3-2 | Reserved | R | 0h | |
| 1-0 | PowerState | R/W | 0h | Power state control<br>0x0: OFF : OFF State<br>0x3: ON : ON State |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.14.2 PM_HDVICP_PWRSTST Register (offset = 4h) [reset = 17h]

PM_HDVICP_PWRSTST is shown in Figure 2-231 and described in Table 2-260.

This register provides a status on the current HDVICP power domain state. [warm reset insensitive]

**Figure 2-231. PM_HDVICP_PWRSTST Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | InTransition | Reserved | | | |
| R-0h | | | R-0h | R-0h | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | HDVICP_MEM_StateSt | | Reserved | LogicStateSt | PowerStateSt | |
| R-0h | | R-1h | | R-0h | R-1h | R-3h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-260. PM_HDVICP_PWRSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20 | InTransition | R | 0h | Domain transition status<br>0x0: No : No on-going transition on power domain<br>0x1: Ongoing : Power domain transition is in progress. |
| 19-6 | Reserved | R | 0h | |
| 5-4 | HDVICP_MEM_StateSt | R | 1h | HDVICP memory state status<br>0x0: Mem_off : Memory is OFF<br>0x2: Reserved : Reserved<br>0x3: Mem_on : Memory is ON |
| 3 | Reserved | R | 0h | |
| 2 | LogicStateSt | R | 1h | Logic state status<br>0x0: OFF : Logic in domain is OFF<br>0x1: ON : Logic in domain is ON |
| 1-0 | PowerStateSt | R | 3h | Current Power State Status<br>0x0: OFF : OFF State [warm reset insensitive]<br>0x3: ON : ON State [warm reset insensitive] |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.14.3 RM_HDVICP_RSTCTRL Register (offset = 10h) [reset = 7h]

RM_HDVICP_RSTCTRL is shown in Figure 2-232 and described in Table 2-261.

This register controls the release of the HDVICP sub-system resets.

**Figure 2-232. RM_HDVICP_RSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | HDVICP_RST3 | HDVICP_RST2 | HDVICP_RST1 |
| R-0h | | | | | R/W-1h | R/W-1h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-261. RM_HDVICP_RSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2 | HDVICP_RST3 | R/W | 1h | HDVICP logic and SL2 reset control<br>0x0: CLEAR: Reset is cleared for the HDVICP logic and SL2<br>0x1: ASSERT: Reset is asserted for HDVICP logic and SL2 |
| 1 | HDVICP_RST2 | R/W | 1h | HDVICP sequencer2 reset control<br>0x0: CLEAR : Reset is cleared for the HDVICP sequencer CPU2<br>0x1: ASSERT : Reset is asserted for the HDVICP sequencer CPU2 |
| 0 | HDVICP_RST1 | R/W | 1h | HDVICP sequencer1 reset control<br>0x0: CLEAR : Reset is cleared for the HDVICP sequencer CPU1<br>0x1: ASSERT : Reset is asserted for the HDVICP sequencer CPU1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.14.4 RM_HDVICP_RSTST Register (offset = 14h) [reset = 0h]

RM_HDVICP_RSTST is shown in Figure 2-233 and described in Table 2-262.

This register logs the different reset sources of the HDVICP domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

**Figure 2-233. RM_HDVICP_RSTST Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | ICECRUSHER_SEQ2_RST2 | ICECRUSHER_SEQ1_RST1 | EMULATION_SEQ2_RST2 | EMULATION_SEQ1_RST1 | HDVICP_RST3 | HDVICP_RST2 | HDVICP_RST1 |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-262. RM_HDVICP_RSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-7 | Reserved | R | 0h | |
| 6 | ICECRUSHER_SEQ2_RST2 | R/W | 0h | Sequencer2 CPU has been reset due to HDVICP ICECRUSHER1 reset event<br>0x0 = RESET_NO : No icecrusher reset<br>0x1 = RESET_YES : Sequencer2 has been reset upon icecrusher reset |
| 5 | ICECRUSHER_SEQ1_RST1 | R/W | 0h | Sequencer1 CPU has been reset due to HDVICP ICECRUSHER1 reset event<br>0x0 = RESET_NO : No icecrusher reset<br>0x1 = RESET_YES : Sequencer1 has been reset upon icecrusher reset |
| 4 | EMULATION_SEQ2_RST2 | R/W | 0h | Sequencer2 CPU has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module<br>0x0 = RESET_NO : No emulation reset<br>0x1 = RESET_YES : Sequencer2 has been reset upon emulation reset |
| 3 | EMULATION_SEQ1_RST1 | R/W | 0h | Sequencer1 CPU has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module<br>0x0 = RESET_NO : No emulation reset<br>0x1 = RESET_YES : Sequencer1 has been reset upon emulation reset |
| 2 | HDVICP_RST3 | R/W | 0h | HDVICP logic and SL2 SW reset<br>0x0 = RESET_NO : No SW reset occured<br>0x1 = RESET_YES : HDVICP logic and SL2 has been reset upon SW reset |
| 1 | HDVICP_RST2 | R/W | 0h | HDVICP Sequencer2 CPU SW reset<br>0x0 = RESET_NO : No SW reset occured<br>0x1 = RESET_YES : Sequencer2 has been reset upon SW reset |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-262. RM_HDVICP_RSTST Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | HDVICP_RST1 | R/W | 0h | HDVICP Sequencer1 CPU SW reset<br>0x0 = RESET_NO : No SW reset occured<br>0x1 = RESET_YES : Sequencer1 has been reset upon SW reset |

### 2.10.15 PRM_ISP Registers

Table 2-263 lists the memory-mapped registers for the PRM_ISP. All register offset addresses not listed in Table 2-263 should be considered as reserved locations and the register contents should not be modified.

**Table 2-263. PRM_ISP REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | PM_ISP_PWRSTCTRL | This register controls the ISP power state to reach upon a domain sleep transition [warm reset insensitive] | Section 2.10.15.1 |
| 4h | PM_ISP_PWRSTST | This register provides a status on the current ISP power domain state.<br>[warm reset insensitive] | Section 2.10.15.2 |
| 10h | RM_ISP_RSTCTRL | This register controls the release of the ISP sub-system resets. | Section 2.10.15.3 |
| 14h | RM_ISP_RSTST | This register logs the different reset sources of the ISP domain.<br>Each bit is set upon release of the domain reset signal.<br>Must be cleared by software.<br>[warm reset insensitive] | Section 2.10.15.4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.15.1 PM_ISP_PWRSTCTRL Register (offset = 0h) [reset = 30000h]

PM_ISP_PWRSTCTRL is shown in Figure 2-234 and described in Table 2-264.

This register controls the ISP power state to reach upon a domain sleep transition [warm reset insensitive]

**Figure 2-234. PM_ISP_PWRSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | ISP_MEM_ONState | |
| R-0h | | | | | | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | LowPowerStateChange | Reserved | | PowerState | |
| R-0h | | | R/W-0h | R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-264. PM_ISP_PWRSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | ISP_MEM_ONState | R | 3h | ISP memory state when domain is ON. |
| 15-5 | Reserved | R | 0h | |
| 4 | LowPowerStateChange | R/W | 0h | Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain.<br>0x0: DIS : Do not request a low power state change.<br>0x1: EN : Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON. |
| 3-2 | Reserved | R | 0h | |
| 1-0 | PowerState | R/W | 0h | Power state control<br>0x0: OFF : OFF State<br>0x3: ON : ON State |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.15.2 PM_ISP_PWRSTST Register (offset = 4h) [reset = 17h]

PM_ISP_PWRSTST is shown in Figure 2-235 and described in Table 2-265.

This register provides a status on the current ISP power domain state. [warm reset insensitive]

**Figure 2-235. PM_ISP_PWRSTST Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | InTransition | Reserved | | | |
| R-0h | | | R-0h | R-0h | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ISP_MEM_StateSt | | Reserved | LogicStateSt | PowerStateSt | |
| R-0h | | R-1h | | R-0h | R-1h | R-3h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-265. PM_ISP_PWRSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20 | InTransition | R | 0h | Domain transition status<br>0x0: No : No on-going transition on power domain<br>0x1: Ongoing : Power domain transition is in progress. |
| 19-6 | Reserved | R | 0h | |
| 5-4 | ISP_MEM_StateSt | R | 1h | ISP memory state status<br>0x0: Mem_off : Memory is OFF<br>0x2: Reserved : Reserved<br>0x3: Mem_on : Memory is ON |
| 3 | Reserved | R | 0h | |
| 2 | LogicStateSt | R | 1h | Logic state status<br>0x0: OFF : Logic in domain is OFF<br>0x1: ON : Logic in domain is ON |
| 1-0 | PowerStateSt | R | 3h | Current Power State Status<br>0x0: OFF : OFF State [warm reset insensitive]<br>0x3: ON : ON State [warm reset insensitive] |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.15.3 RM_ISP_RSTCTRL Register (offset = 10h) [reset = 7h]

RM_ISP_RSTCTRL is shown in Figure 2-236 and described in Table 2-266.

This register controls the release of the ISP sub-system resets.

**Figure 2-236. RM_ISP_RSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | ISP_RST | Reserved | Reserved |
| R-0h | | | | | R/W-1h | R/W-1h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-266. RM_ISP_RSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2 | ISP_RST | R/W | 1h | ISP logic and FDIF reset control<br>0x0: CLEAR : Reset is cleared for the ISP logic and FDIF<br>0x1: ASSERT : Reset is asserted for ISP logic and FDIF |
| 1 | Reserved | R/W | 1h | |
| 0 | Reserved | R/W | 1h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.15.4 RM_ISP_RSTST Register (offset = 14h) [reset = 0h]

RM_ISP_RSTST is shown in Figure 2-237 and described in Table 2-267.

This register logs the different reset sources of the ISP domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

**Figure 2-237. RM_ISP_RSTST Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | Reserved | Reserved | Reserved | Reserved | ISP_RST | Reserved | Reserved |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-267. RM_ISP_RSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-7 | Reserved | R | 0h | |
| 6 | Reserved | R/W | 0h | |
| 5 | Reserved | R/W | 0h | |
| 4 | Reserved | R/W | 0h | |
| 3 | Reserved | R/W | 0h | |
| 2 | ISP_RST | R/W | 0h | ISP logic and FDIF SW reset<br>0x0: RESET_NO : No SW reset occured<br>0x1: RESET_YES : ISP logic and FDIF has been reset upon SW reset |
| 1 | Reserved | R/W | 0h | |
| 0 | Reserved | R/W | 0h | |

## *2.10.16 PRM_DSS Registers*

Table 2-268 lists the memory-mapped registers for the PRM_DSS. All register offset addresses not listed in Table 2-268 should be considered as reserved locations and the register contents should not be modified.

**Table 2-268. PRM_DSS REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | PM_DSS_PWRSTCTRL | This register controls the DSS power state to reach upon a domain sleep transition [warm reset insensitive] | Section 2.10.16.1 |
| 4h | PM_DSS_PWRSTST | This register provides a status on the current DSS power domain state.<br>[warm reset insensitive] | Section 2.10.16.2 |
| 10h | RM_DSS_RSTCTRL | This register controls the release of the DSS sub-system resets. | Section 2.10.16.3 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-268. PRM_DSS REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 14h | RM_DSS_RSTST | This register logs the different reset sources of the DSS domain.<br>Each bit is set upon release of the domain reset signal.<br>Must be cleared by software.<br>[warm reset insensitive] | Section 2.10.16.4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.16.1 PM_DSS_PWRSTCTRL Register (offset = 0h) [reset = 30000h]

PM_DSS_PWRSTCTRL is shown in Figure 2-238 and described in Table 2-269.

This register controls the DSS power state to reach upon a domain sleep transition [warm reset insensitive]

**Figure 2-238. PM_DSS_PWRSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | DSS_MEM_ONState | |
| R-0h | | | | | | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | LowPowerStateChange | Reserved | | PowerState | |
| R-0h | | | R/W-0h | R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-269. PM_DSS_PWRSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | DSS_MEM_ONState | R | 3h | DSS memory state when domain is ON. |
| 15-5 | Reserved | R | 0h | |
| 4 | LowPowerStateChange | R/W | 0h | Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain.<br>0x0 = DIS : Do not request a low power state change.<br>0x1 = EN : Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON. |
| 3-2 | Reserved | R | 0h | |
| 1-0 | PowerState | R/W | 0h | Power state control<br>0x0 = OFF : OFF State<br>0x3 = ON : ON State |

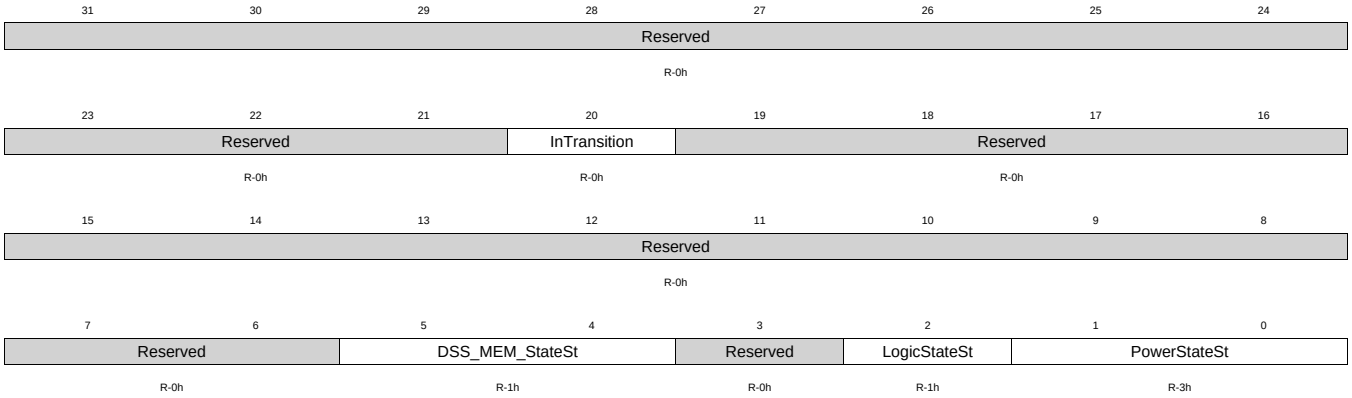

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.16.2 PM_DSS_PWRSTST Register (offset = 4h) [reset = 17h]

PM_DSS_PWRSTST is shown in Figure 2-239 and described in Table 2-270.

This register provides a status on the current DSS power domain state. [warm reset insensitive]

**Figure 2-239. PM_DSS_PWRSTST Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-270. PM_DSS_PWRSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20 | InTransition | R | 0h | Domain transition status<br>0x0 = No : No on-going transition on power domain<br>0x1 = Ongoing : Power domain transition is in progress. |
| 19-6 | Reserved | R | 0h | |
| 5-4 | DSS_MEM_StateSt | R | 1h | DSS memory state status<br>0x0 = Mem_off : Memory is OFF<br>0x2 = Reserved : Reserved<br>0x3 = Mem_on : Memory is ON |
| 3 | Reserved | R | 0h | |
| 2 | LogicStateSt | R | 1h | Logic state status<br>0x0 = OFF : Logic in domain is OFF<br>0x1 = ON : Logic in domain is ON |
| 1-0 | PowerStateSt | R | 3h | Current Power State Status<br>0x0 = OFF : OFF State [warm reset insensitive]<br>0x3 = ON : ON State [warm reset insensitive] |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.16.3 RM_DSS_RSTCTRL Register (offset = 10h) [reset = 7h]

RM_DSS_RSTCTRL is shown in Figure 2-240 and described in Table 2-271.

This register controls the release of the DSS sub-system resets.

**Figure 2-240. RM_DSS_RSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | DSS_RST | Reserved | Reserved |
| R-0h | | | | | R/W-1h | R/W-1h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-271. RM_DSS_RSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | |
| 2 | DSS_RST | R/W | 1h | DSS logic and HDMI reset control<br>0x0 = CLEAR : Reset is cleared for the DSS logic and HDMI<br>0x1 = ASSERT : Reset is asserted for DSS logic and HDMI |
| 1 | Reserved | R/W | 1h | |
| 0 | Reserved | R/W | 1h | |

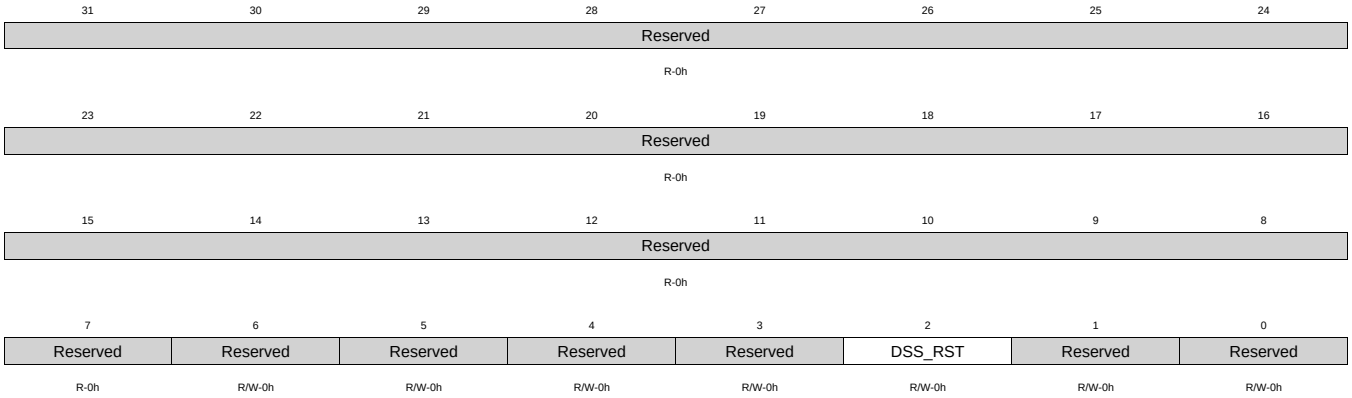

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.16.4 RM_DSS_RSTST Register (offset = 14h) [reset = 0h]

RM_DSS_RSTST is shown in Figure 2-241 and described in Table 2-272.

This register logs the different reset sources of the DSS domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

**Figure 2-241. RM_DSS_RSTST Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-272. RM_DSS_RSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-7 | Reserved | R | 0h | |
| 6 | Reserved | R/W | 0h | |
| 5 | Reserved | R/W | 0h | |
| 4 | Reserved | R/W | 0h | |
| 3 | Reserved | R/W | 0h | |
| 2 | DSS_RST | R/W | 0h | DSS logic and HDMI SW reset<br>0x0 = RESET_NO : No SW reset occured<br>0x1 = RESET_YES : DSS logic and HDMI has been reset upon SW reset |
| 1 | Reserved | R/W | 0h | |
| 0 | Reserved | R/W | 0h | |

## *2.10.17 PRM_SGX Registers*

Table 2-273 lists the memory-mapped registers for the PRM_SGX. All register offset addresses not listed in Table 2-273 should be considered as reserved locations and the register contents should not be modified.

**Table 2-273. PRM_SGX REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | PM_SGX_PWRSTCTRL | This register controls the SGX power state to reach upon a domain sleep transition.<br>[warm reset insensitive] | Section 2.10.17.1 |
| 4h | RM_SGX_RSTCTRL | This register controls the release of the SGX Domain resets. | Section 2.10.17.2 |
| 10h | PM_SGX_PWRSTST | This register provides a status on the current HDVICP power domain state.<br>[warm reset insensitive] | Section 2.10.17.3 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-273. PRM_SGX REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 14h | RM_SGX_RSTST | This register logs the different reset sources of the SGX domain.<br>Each bit is set upon release of the domain reset signal.<br>Must be cleared by software.<br>[warm reset insensitive] | Section 2.10.17.4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.17.1 PM_SGX_PWRSTCTRL Register (offset = 0h) [reset = 30000h]

PM_SGX_PWRSTCTRL is shown in Figure 2-242 and described in Table 2-274.

This register controls the SGX power state to reach upon a domain sleep transition. [warm reset insensitive]

**Figure 2-242. PM_SGX_PWRSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | SGX_MEM_ONState | |
| R-0h | | | | | | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | LowPowerStateChange | Reserved | | PowerState | |
| R-0h | | | R/W-0h | R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-274. PM_SGX_PWRSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | SGX_MEM_ONState | R | 3h | SGX memory state when domain is ON. |
| 15-5 | Reserved | R | 0h | |
| 4 | LowPowerStateChange | R/W | 0h | Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain.<br>0x0: DIS : Do not request a low power state change.<br>0x1: EN : Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON. |
| 3-2 | Reserved | R | 0h | |
| 1-0 | PowerState | R/W | 0h | Power state control<br>0x0: OFF : OFF State<br>0x3: ON : ON State |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.17.2 RM_SGX_RSTCTRL Register (offset = 4h) [reset = 1h]

RM_SGX_RSTCTRL is shown in Figure 2-243 and described in Table 2-275.

This register controls the release of the SGX Domain resets.

**Figure 2-243. RM_SGX_RSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | Reserved | | Reserved | SGX_RST |
| R-0h | | | | R-0h | | R-0h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-275. RM_SGX_RSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | Reserved | R | 0h | |
| 3-2 | Reserved | R | 0h | |
| 1 | Reserved | R | 0h | |
| 0 | SGX_RST | R/W | 1h | SGX domain local reset control<br>0x0: CLEAR : Reset is cleared for the SGX.<br>0x1: ASSERT : Reset is asserted for the SGX. |

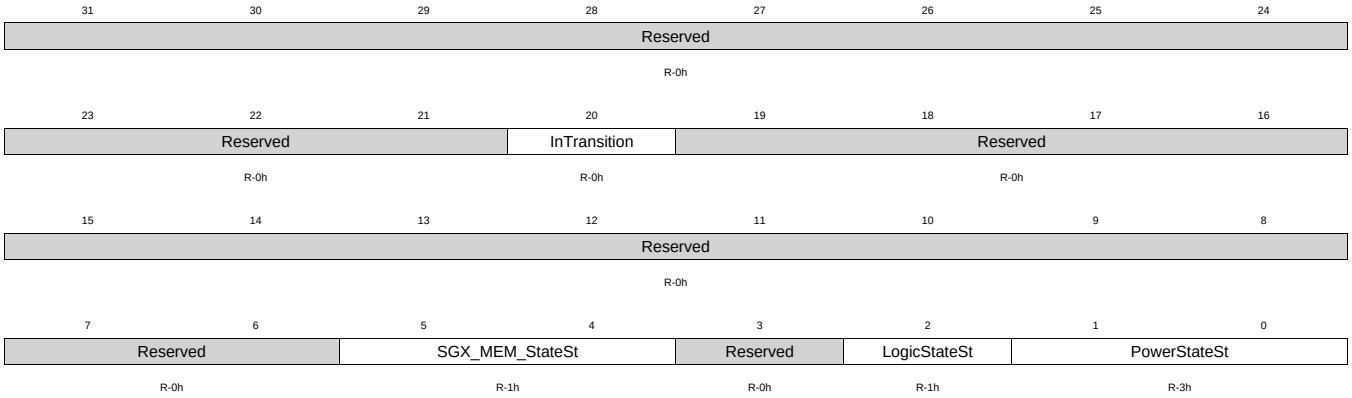

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.17.3 PM_SGX_PWRSTST Register (offset = 10h) [reset = 17h]

PM_SGX_PWRSTST is shown in Figure 2-244 and described in Table 2-276.

This register provides a status on the current HDVICP power domain state. [warm reset insensitive]

**Figure 2-244. PM_SGX_PWRSTST Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-276. PM_SGX_PWRSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | Reserved | R | 0h | |
| 20 | InTransition | R | 0h | Domain transition status<br>0x0: No : No on-going transition on power domain<br>0x1: Ongoing : Power domain transition is in progress. |
| 19-6 | Reserved | R | 0h | |
| 5-4 | SGX_MEM_StateSt | R | 1h | SGX memory state status<br>0x0: Mem_off : Memory is OFF<br>0x2: Reserved : Reserved<br>0x3: Mem_on : Memory is ON |
| 3 | Reserved | R | 0h | |
| 2 | LogicStateSt | R | 1h | Logic state status<br>0x0: OFF : Logic in domain is OFF<br>0x1: ON : Logic in domain is ON |
| 1-0 | PowerStateSt | R | 3h | Current Power State Status<br>0x0: OFF : OFF State [warm reset insensitive]<br>0x3: ON : ON State [warm reset insensitive] |



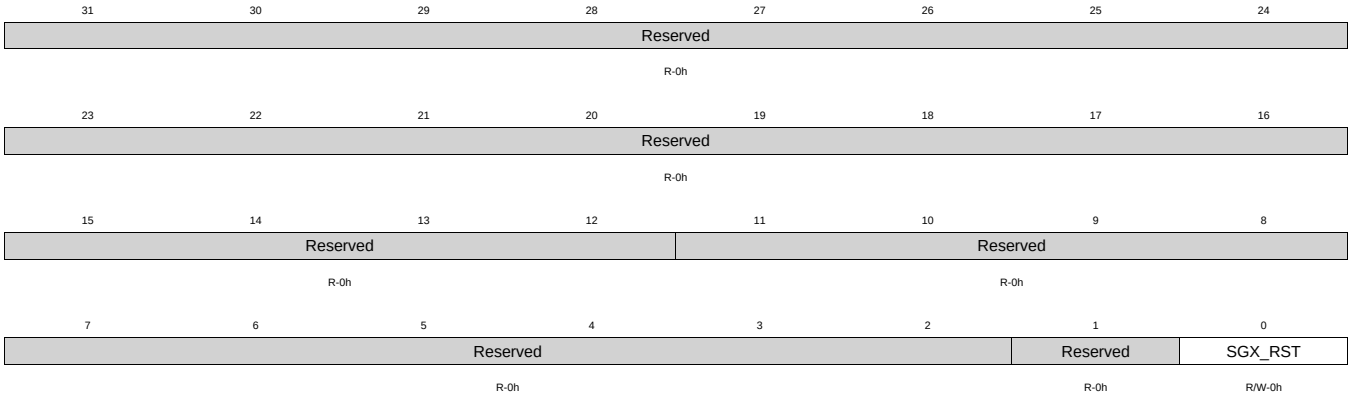
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.17.4 RM_SGX_RSTST Register (offset = 14h) [reset = 0h]

RM_SGX_RSTST is shown in Figure 2-245 and described in Table 2-277.

This register logs the different reset sources of the SGX domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

**Figure 2-245. RM_SGX_RSTST Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-277. RM_SGX_RSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | Reserved | R | 0h | |
| 11-2 | Reserved | R | 0h | |
| 1 | Reserved | R | 0h | |
| 0 | SGX_RST | R/W | 0h | SGX Domain Logic Reset<br>0x0: RESET_NO : No SW reset occured<br>0x1: RESET_YES : SGX Domain Logic has been reset upon SW reset |

## *2.10.18 CM_ALWON Registers*

Table 2-278 lists the memory-mapped registers for the CM_ALWON. All register offset addresses not listed in Table 2-278 should be considered as reserved locations and the register contents should not be modified.

**Table 2-278. CM_ALWON REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | CM_ALWON_L3_SLOW_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states.<br>It also hold one status bit per clock input of the domain. | Section 2.10.18.1 |
| 4h | CM_ETHERNET_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states.<br>It also hold one status bit per clock input of the domain. | Section 2.10.18.2 |
| 8h | CM_ALWON_L3_MED_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states.<br>It also hold one status bit per clock input of the domain. | Section 2.10.18.3 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-278. CM_ALWON REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| Ch | CM_MMU_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.18.4 |
| 10h | CM_MMUCFG_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.18.5 |
| 14h | CM_ALWON_OCMC_0_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.18.6 |
| 18h | CM_ALWON_VCP_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.18.7 |
| 1Ch | CM_ALWON_MPU_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.18.8 |
| 20h | CM_ALWON_SYSCLK4_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.18.9 |
| 24h | CM_ALWON_SYSCLK5_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.18.10 |
| 28h | CM_ALWON_SYSCLK6_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.18.11 |
| 2Ch | CM_ALWON_RTC_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.18.12 |
| 30h | CM_ALWON_L3_FAST_CLKSTCTRL | This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. | Section 2.10.18.13 |
| 140h | CM_ALWON_MCASP0_CLKCTRL | This register manages the MCASP_0 clocks. | Section 2.10.18.14 |
| 144h | CM_ALWON_MCASP1_CLKCTRL | This register manages the MCASP1 clocks. | Section 2.10.18.15 |
| 148h | CM_ALWON_MCASP2_CLKCTRL | This register manages the MCASP2 clocks. | Section 2.10.18.16 |
| 14Ch | CM_ALWON_MCBSP_CLKCTRL | This register manages the MCBSP clocks. | Section 2.10.18.17 |
| 150h | CM_ALWON_UART_0_CLKCTRL | This register manages the UART_0 clocks. | Section 2.10.18.18 |
| 154h | CM_ALWON_UART_1_CLKCTRL | This register manages the UART_1 clocks. | Section 2.10.18.19 |
| 158h | CM_ALWON_UART_2_CLKCTRL | This register manages the UART_2 clocks. | Section 2.10.18.20 |
| 15Ch | CM_ALWON_GPIO_0_CLKCTRL | This register manages the GPIO_0 clocks. | Section 2.10.18.21 |
| 160h | CM_ALWON_GPIO_1_CLKCTRL | This register manages the GPIO_1, GPIO_2 and GPIO_3 clocks. | Section 2.10.18.22 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-278. CM_ALWON REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 164h | CM_ALWON_I2C_0_CLKCTRL | This register manages the I2C_0 and I2C_2 clocks. | Section 2.10.18.23 |
| 168h | CM_ALWON_I2C_1_CLKCTRL | This register manages the I2C_1 and I2C_3 clocks. | Section 2.10.18.24 |
| 16Ch | CM_ALWON_MCASP_3_4_5_CLKCTRL | This register manages the MCASP_3_4_5 clocks. | Section 2.10.18.25 |
| 170h | CM_ALWON_ATL_CLKCTRL | This register manages the ATL clocks. | Section 2.10.18.26 |
| 174h | CM_ALWON_MLB_CLKCTRL | This register manages the MLB clocks. | Section 2.10.18.27 |
| 178h | CM_ALWON_PATA_CLKCTRL | This register manages the PATA clocks. | Section 2.10.18.28 |
| 180h | CM_ALWON_UART_3_CLKCTRL | This register manages the UART 3 clocks. | Section 2.10.18.29 |
| 184h | CM_ALWON_UART_4_CLKCTRL | This register manages the UART 4 clocks. | Section 2.10.18.30 |
| 188h | CM_ALWON_UART_5_CLKCTRL | This register manages the UART 5 clocks. | Section 2.10.18.31 |
| 18Ch | CM_ALWON_WDTIMER_CLKCTRL | This register manages the WDTIMER clocks. | Section 2.10.18.32 |
| 190h | CM_ALWON_SPI_CLKCTRL | This register manages the SPI0, SPI1, SPI2 and SPI3 clocks. | Section 2.10.18.33 |
| 194h | CM_ALWON_MAILBOX_CLKCTRL | This register manages the MAILBOX clocks. | Section 2.10.18.34 |
| 198h | CM_ALWON_SPINBOX_CLKCTRL | This register manages the SPINBOX clocks. | Section 2.10.18.35 |
| 19Ch | CM_ALWON_MMUDATA_CLKCTRL | This register manages the MMU data clocks. | Section 2.10.18.36 |
| 1A8h | CM_ALWON_MMUCFG_CLKCTRL | This register manages the MMU config clocks. | Section 2.10.18.37 |
| 1B4h | CM_ALWON_OCMC_0_CLKCTRL | This register manages the OCMC_0 clocks. | Section 2.10.18.38 |
| 1B8h | CM_ALWON_VCP_CLKCTRL | This register manages the VCP clocks. | Section 2.10.18.39 |
| 1C4h | CM_ALWON_CONTROL_CLKCTRL | This register manages the CONTROL clocks. | Section 2.10.18.40 |
| 1D0h | CM_ALWON_GPMC_CLKCTRL | This register manages the GPMC clocks. | Section 2.10.18.41 |
| 1D4h | CM_ALWON_ETHERNET_0_CLKCTRL | This register manages the ETHERNET_0 switch clocks including both Ethernet ports. | Section 2.10.18.42 |
| 1DCh | CM_ALWON_MPU_CLKCTRL | This register manages the MPU clocks. | Section 2.10.18.43 |
| 1E0h | CM_ALWON_DEBUGSS_CLKCTRL | This register manages the DEBUGSS clocks. | Section 2.10.18.44 |
| 1E4h | CM_ALWON_L3_CLKCTRL | This register manages the L3 clocks. | Section 2.10.18.45 |
| 1E8h | CM_ALWON_L4HS_CLKCTRL | This register manages the L4HS clocks. | Section 2.10.18.46 |
| 1ECh | CM_ALWON_L4LS_CLKCTRL | This register manages the L4LS clocks. | Section 2.10.18.47 |
| 1F0h | CM_ALWON_RTC_CLKCTRL | This register manages the RTC clocks. | Section 2.10.18.48 |
| 1F4h | CM_ALWON_TPCC_CLKCTRL | This register manages the TPCC clocks. | Section 2.10.18.49 |
| 1F8h | CM_ALWON_TPTC0_CLKCTRL | This register manages the TPTC_0 clocks. | Section 2.10.18.50 |
| 1FCh | CM_ALWON_TPTC1_CLKCTRL | This register manages the TPTC_1 clocks. | Section 2.10.18.51 |
| 200h | CM_ALWON_TPTC2_CLKCTRL | This register manages the TPTC_2 clocks. | Section 2.10.18.52 |
| 204h | CM_ALWON_TPTC3_CLKCTRL | This register manages the TPTC_3 clocks. | Section 2.10.18.53 |
| 218h | CM_ALWON_DCAN_0_1_CLKCTRL | This register manages the DCAN_0_1 clocks. | Section 2.10.18.54 |
| 21Ch | CM_ALWON_MMCHS_0_CLKCTRL | This register manages the MMCHS_0 clocks. | Section 2.10.18.55 |
| 220h | CM_ALWON_MMCHS_1_CLKCTRL | This register manages the MMCHS_1 clocks. | Section 2.10.18.56 |
| 224h | CM_ALWON_MMCHS_2_CLKCTRL | This register manages the MMCHS_2 clocks. | Section 2.10.18.57 |
| 228h | CM_ALWON_CUST_EFUSE_CLKCTRL | This register manages the Customer Efuse clocks. | Section 2.10.18.58 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.1 CM_ALWON_L3_SLOW_CLKSTCTRL Register (offset = 0h) [reset = C002h]

CM_ALWON_L3_SLOW_CLKSTCTRL is shown in Figure 2-246 and described in Table 2-279.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-246. CM_ALWON_L3_SLOW_CLKSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CLKACTIVITY_SDIO_CLKADPI_GCLK | CLKACTIVITY_TIMER7_GCLK | CLKACTIVITY_TIMER6_GCLK | CLKACTIVITY_TIMER5_GCLK |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CLKACTIVITY_TIMER4_GCLK | CLKACTIVITY_TIMER3_GCLK | CLKACTIVITY_TIMER2_GCLK | CLKACTIVITY_TIMER1_GCLK | Reserved | Reserved | CLKACTIVITY_SPI_GSYSCLK | CLKACTIVITY_I2C_GSYSCLK |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CLKACTIVITY_GPIO_1_GDBCLK | CLKACTIVITY_GPIO_0_GDBCLK | CLKACTIVITY_UART_GFCLK | CLKACTIVITY_MCBSP_AUX_GCLK | CLKACTIVITY_MCASP2_AUX_GCLK | CLKACTIVITY_MCASP1_AUX_GCLK | CLKACTIVITY_MCASP0_AUX_GCLK | CLKACTIVITY_L3_SLOW_GCLK |
| R-1h | R-1h | R-0h | R-0h | R-0h | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | CLKTRCTRL | |
| R-0h | | | | | | R/W-2h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-279. CM_ALWON_L3_SLOW_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | |
| 27 | CLKACTIVITY_SDIO_CLKADPI_GCLK | R | 0h | This field indicates the state of the SDIO_ADPI clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 26 | CLKACTIVITY_TIMER7_GCLK | R | 0h | This field indicates the state of the TIMER7 CLKTIMER clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 25 | CLKACTIVITY_TIMER6_GCLK | R | 0h | This field indicates the state of the TIMER6 CLKTIMER clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 24 | CLKACTIVITY_TIMER5_GCLK | R | 0h | This field indicates the state of the TIMER5 CLKTIMER clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 23 | CLKACTIVITY_TIMER4_GCLK | R | 0h | This field indicates the state of the TIMER4 CLKTIMER clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 22 | CLKACTIVITY_TIMER3_GCLK | R | 0h | This field indicates the state of the TIMER3 CLKTIMER clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 21 | CLKACTIVITY_TIMER2_GCLK | R | 0h | This field indicates the state of the TIMER2 CLKTIMER clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-279. CM_ALWON_L3_SLOW_CLKSTCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | CLKACTIVITY_TIMER1_GCLK | R | 0h | This field indicates the state of the TIMER1 CLKTIMER clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 19 | Reserved | R | 0h | Reserved |
| 18 | Reserved | R | 0h | Reserved |
| 17 | CLKACTIVITY_SPI_GSYSCLK | R | 0h | This field indicates the state of the SPI_GSYSCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 16 | CLKACTIVITY_I2C_GSYSCLK | R | 0h | This field indicates the state of the I2C_GSYSCLK clock in the domain.<br>0x0: Inact : Corresponding clock is gated<br>0x1: Act : Corresponding clock is active |
| 15 | CLKACTIVITY_GPIO_1_GDBCLK | R | 1h | This field indicates the state of the GPIO_GDBCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 14 | CLKACTIVITY_GPIO_0_GDBCLK | R | 1h | This field indicates the state of the GPIO_GDBCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 13 | CLKACTIVITY_UART_GFCLK | R | 0h | This field indicates the state of the UART_GFCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 12 | CLKACTIVITY_MCBSP_AUX_GCLK | R | 0h | This field indicates the state of the MCBSP_AUX_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 11 | CLKACTIVITY_MCASP2_AUX_GCLK | R | 0h | This field indicates the state of the MCASP2_AUX_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 10 | CLKACTIVITY_MCASP1_AUX_GCLK | R | 0h | This field indicates the state of the MCASP1_AUX_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 9 | CLKACTIVITY_MCASP0_AUX_GCLK | R | 0h | This field indicates the state of the MCASP0_AUX_GCLK clock in the domain.<br>0x0: Inact : Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 8 | CLKACTIVITY_L3_SLOW_GCLK | R | 1h | This field indicates the state of the CM_ALWON_L3_SLOW_CLKSTCTRL clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-279. CM_ALWON_L3_SLOW_CLKSTCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1-0 | CLKTRCTRL | R/W | 2h | Controls the clock state transition of the L3_SLOW clock domain in Always ON power domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |

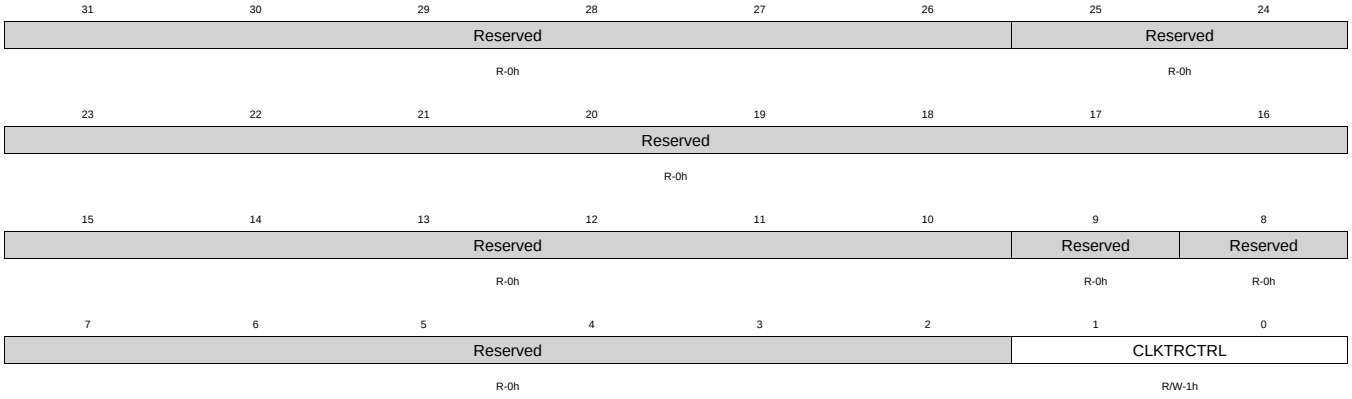

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.2 CM_ETHERNET_CLKSTCTRL Register (offset = 4h) [reset = 1h]

CM_ETHERNET_CLKSTCTRL is shown in Figure 2-247 and described in Table 2-280.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-247. CM_ETHERNET_CLKSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-280. CM_ETHERNET_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-10 | Reserved | R | 0h | |
| 9 | Reserved | R | 0h | |
| 8 | Reserved | R | 0h | |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the ETHERNET clock domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.3 CM_ALWON_L3_MED_CLKSTCTRL Register (offset = 8h) [reset = 2h]

CM_ALWON_L3_MED_CLKSTCTRL is shown in Figure 2-248 and described in Table 2-281.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-248. CM_ALWON_L3_MED_CLKSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | Reserved | |
| R-0h | | | | | | R-0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | Reserved | Reserved | Reserved |
| R-0h | | | | | R-0h | R-0h | R-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKTRCTRL | |
| R-0h | | | | | | R-2h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-281. CM_ALWON_L3_MED_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-11 | Reserved | R | 0h | |
| 10 | Reserved | R | 0h | |
| 9 | Reserved | R | 0h | |
| 8 | Reserved | R | 0h | |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R | 2h | Controls the clock state transition of the L3 Medium clock domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.4 CM_MMU_CLKSTCTRL Register (offset = Ch) [reset = 1h]

CM_MMU_CLKSTCTRL is shown in Figure 2-249 and described in Table 2-282.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-249. CM_MMU_CLKSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | Reserved | |
| R-0h | | | | | | R-0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | CLKACTIVITY_MMU_GCLK |
| R-0h | | | | | | | R-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKTRCTRL | |
| R-0h | | | | | | R/W-1h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-282. CM_MMU_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | CLKACTIVITY_MMU_GCLK | R | 0h | This field indicates the state of the MMU_GICLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the MMU clock domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |

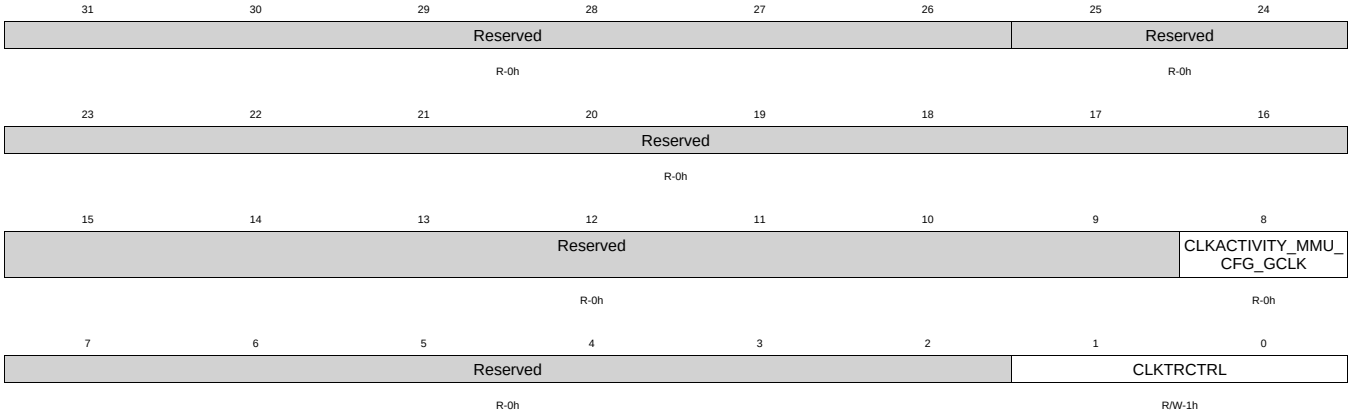

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.5 CM_MMUCFG_CLKSTCTRL Register (offset = 10h) [reset = 1h]

CM_MMUCFG_CLKSTCTRL is shown in Figure 2-250 and described in Table 2-283.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-250. CM_MMUCFG_CLKSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-283. CM_MMUCFG_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | CLKACTIVITY_MMU_CFG_GCLK | R | 0h | This field indicates the state of the MMU_CFG_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the MMU CFG clock domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |

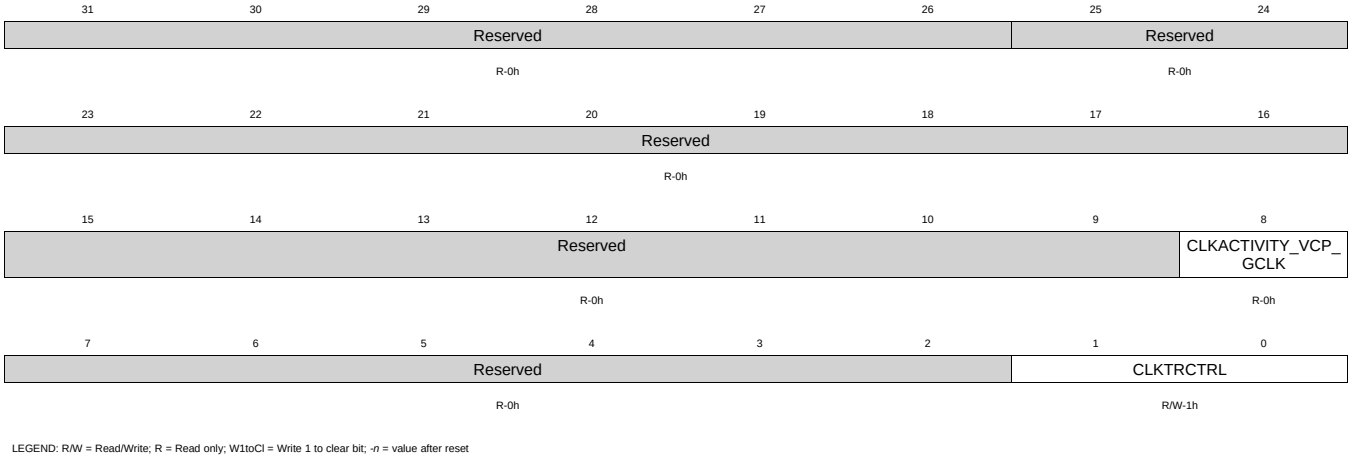

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.6 CM_ALWON_OCMC_0_CLKSTCTRL Register (offset = 14h) [reset = 1h]

CM_ALWON_OCMC_0_CLKSTCTRL is shown in Figure 2-251 and described in Table 2-284.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-251. CM_ALWON_OCMC_0_CLKSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-284. CM_ALWON_OCMC_0_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | CLKACTIVITY_VCP_GCLK | R | 0h | This field indicates the state of the VCP_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the OCMC clock domain in DEFAULT power domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |

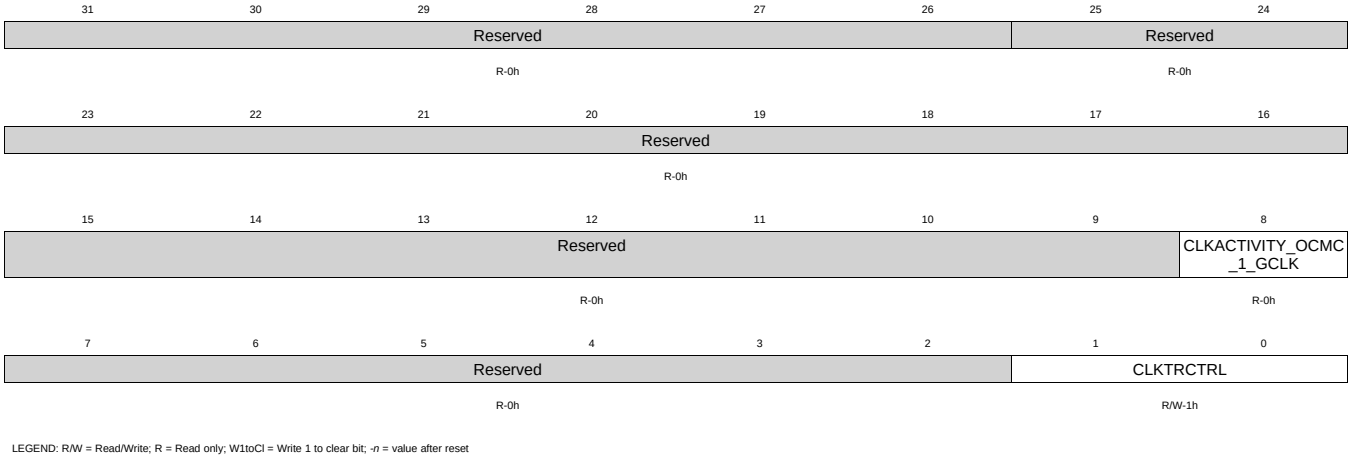

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.7 CM_ALWON_VCP_CLKSTCTRL Register (offset = 18h) [reset = 1h]

CM_ALWON_VCP_CLKSTCTRL is shown in Figure 2-252 and described in Table 2-285.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-252. CM_ALWON_VCP_CLKSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-285. CM_ALWON_VCP_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | CLKACTIVITY_OCMC_1_GCLK | R | 0h | This field indicates the state of the OCMC_1_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the OCMC_2 clock domain in DEFAULT power domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |

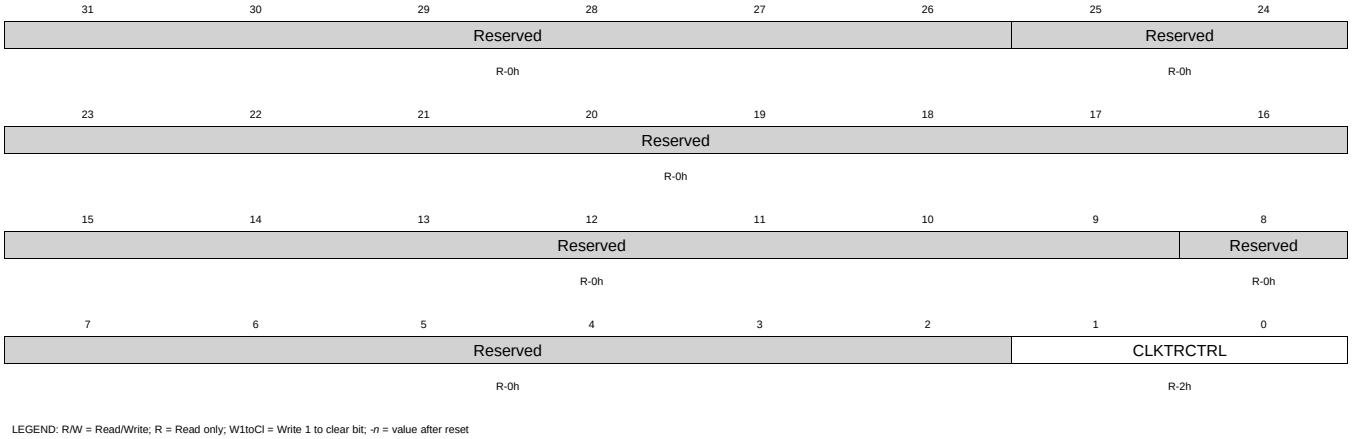

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.8 CM_ALWON_MPU_CLKSTCTRL Register (offset = 1Ch) [reset = 2h]

CM_ALWON_MPU_CLKSTCTRL is shown in Figure 2-253 and described in Table 2-286.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-253. CM_ALWON_MPU_CLKSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-286. CM_ALWON_MPU_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | Reserved | R | 0h | |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R | 2h | Controls the clock state transition of the MPU clock domain in Always ON power domain.<br>0x0: Reserved<br>0x1: Reserved<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.9 CM_ALWON_SYSCLK4_CLKSTCTRL Register (offset = 20h) [reset = 2h]

CM_ALWON_SYSCLK4_CLKSTCTRL is shown in Figure 2-254 and described in Table 2-287.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-254. CM_ALWON_SYSCLK4_CLKSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | Reserved | |
| R-0h | | | | | | R-0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CLKACTIVITY_L3_F_EN_GCLK | CLKACTIVITY_L3_S_GCLK | CLKACTIVITY_L3_M_GCLK | CLKACTIVITY_SYSCLK4_GCLK |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKTRCTRL | |
| R-0h | | | | | | R-2h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-287. CM_ALWON_SYSCLK4_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-12 | Reserved | R | 0h | |
| 11 | CLKACTIVITY_L3_F_EN_GCLK | R | 0h | This field indicates the state of the L3_F_EN_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 10 | CLKACTIVITY_L3_S_GCLK | R | 0h | This field indicates the state of the L3_S_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 9 | CLKACTIVITY_L3_M_GCLK | R | 0h | This field indicates the state of the L3_M_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 8 | CLKACTIVITY_SYSCLK4_GCLK | R | 0h | This field indicates the state of the SYSCLK4_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R | 2h | Controls the clock state transition of the SYSCLK4 clock domain in Always ON power domain.<br>0x0: Reserved<br>0x1: Reserved<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.10 CM_ALWON_SYSCLK5_CLKSTCTRL Register (offset = 24h) [reset = 2h]

CM_ALWON_SYSCLK5_CLKSTCTRL is shown in Figure 2-255 and described in Table 2-288.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-255. CM_ALWON_SYSCLK5_CLKSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | Reserved | |
| R-0h | | | | | | R-0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKACTIVITY_DEBUG_CLKA_GCLK | CLKACTIVITY_SYSCLK5_GCLK |
| R-0h | | | | | | R-0h | R-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKTRCTRL | |
| R-0h | | | | | | R-2h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-288. CM_ALWON_SYSCLK5_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-10 | Reserved | R | 0h | |
| 9 | CLKACTIVITY_DEBUG_CLKA_GCLK | R | 0h | This field indicates the state of the Debug clockA clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 8 | CLKACTIVITY_SYSCLK5_GCLK | R | 0h | This field indicates the state of the SYSCLK5_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R | 2h | Controls the clock state transition of the SYSCLK5 clock domain in Always ON power domain.<br>0x0: Reserved<br>0x1: Reserved<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.11 CM_ALWON_SYSCLK6_CLKSTCTRL Register (offset = 28h) [reset = 2h]

CM_ALWON_SYSCLK6_CLKSTCTRL is shown in Figure 2-256 and described in Table 2-289.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-256. CM_ALWON_SYSCLK6_CLKSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | Reserved | |
| R-0h | | | | | | R-0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | CLKACTIVITY_SYSCLK6_GCLK |
| R-0h | | | | | | | R-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKTRCTRL | |
| R-0h | | | | | | R-2h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-289. CM_ALWON_SYSCLK6_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | CLKACTIVITY_SYSCLK6_GCLK | R | 0h | This field indicates the state of the SYSCLK6_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R | 2h | Controls the clock state transition of the SYSCLK6 clock domain in Always ON power domain.<br>0x0: Reserved<br>0x1: Reserved<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.12 CM_ALWON_RTC_CLKSTCTRL Register (offset = 2Ch) [reset = 2h]

CM_ALWON_RTC_CLKSTCTRL is shown in Figure 2-257 and described in Table 2-290.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-257. CM_ALWON_RTC_CLKSTCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | Reserved | |
| R-0h | | | | | | R-0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | CLKACTIVITY_RTC_GCLK |
| R-0h | | | | | | | R-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLKTRCTRL | |
| R-0h | | | | | | R-2h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-290. CM_ALWON_RTC_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-9 | Reserved | R | 0h | |
| 8 | CLKACTIVITY_RTC_GCLK | R | 0h | This field indicates the state of the RTC_GCLK clock in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R | 2h | Controls the clock state transition of the RTC clock domain in Always ON power domain.<br>0x0 = Reserved<br>0x1 = Reserved<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3 = Reserved |

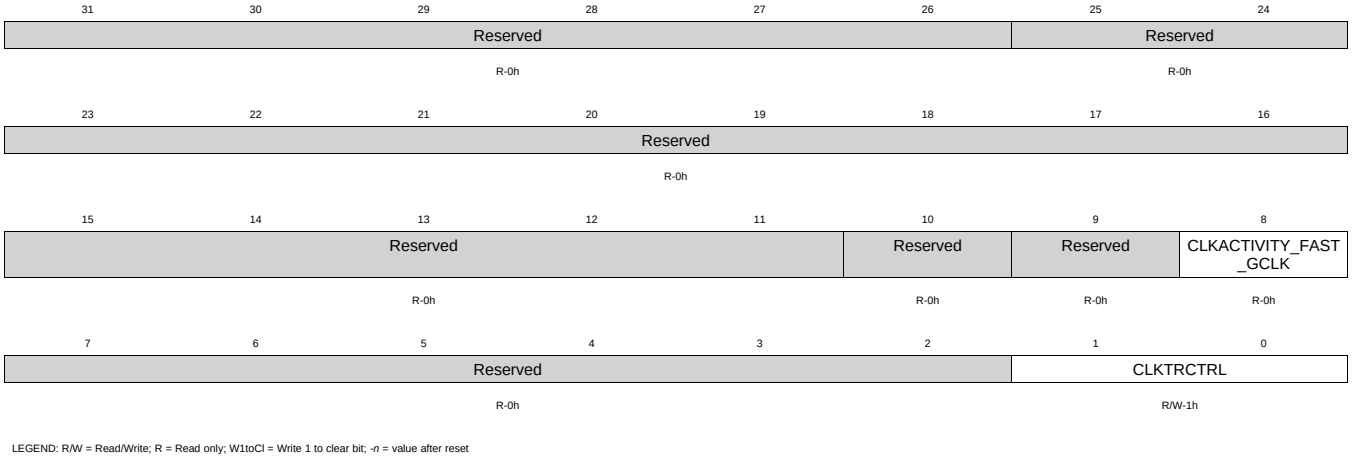

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.13 CM_ALWON_L3_FAST_CLKSTCTRL Register (offset = 30h) [reset = 1h]

CM_ALWON_L3_FAST_CLKSTCTRL is shown in Figure 2-258 and described in Table 2-291.

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

**Figure 2-258. CM_ALWON_L3_FAST_CLKSTCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-291. CM_ALWON_L3_FAST_CLKSTCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25-11 | Reserved | R | 0h | |
| 10 | Reserved | R | 0h | |
| 9 | Reserved | R | 0h | |
| 8 | CLKACTIVITY_FAST_GCLK | R | 0h | This field indicates the state of the L3 Fast clock for TPTC and TPCC in the domain.<br>0x0: Inact: Corresponding clock is gated<br>0x1: Act: Corresponding clock is active |
| 7-2 | Reserved | R | 0h | |
| 1-0 | CLKTRCTRL | R/W | 1h | Controls the clock state transition of the L3 Fast clock domain.<br>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.<br>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.<br>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.<br>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions. |

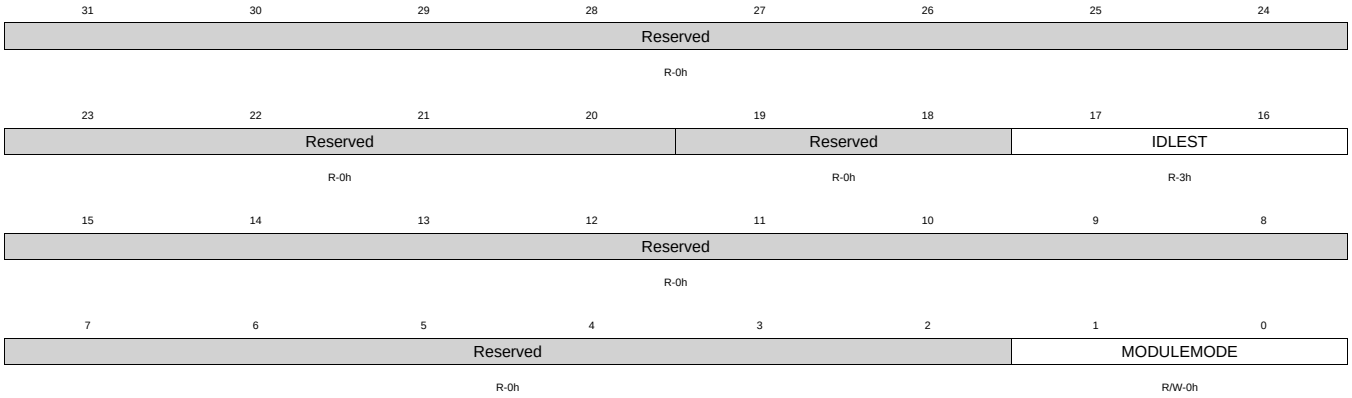

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.14 CM_ALWON_MCASP0_CLKCTRL Register (offset = 140h) [reset = 30000h]

CM_ALWON_MCASP0_CLKCTRL is shown in Figure 2-259 and described in Table 2-292.

This register manages the MCASP_0 clocks.

**Figure 2-259. CM_ALWON_MCASP0_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-292. CM_ALWON_MCASP0_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

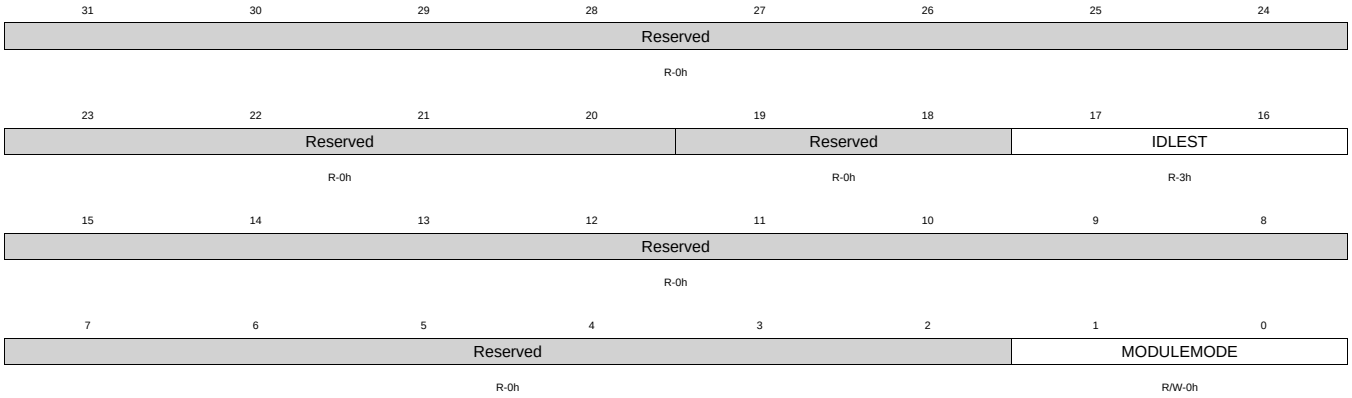

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.15 CM_ALWON_MCASP1_CLKCTRL Register (offset = 144h) [reset = 30000h]

CM_ALWON_MCASP1_CLKCTRL is shown in Figure 2-260 and described in Table 2-293.

This register manages the MCASP1 clocks.

**Figure 2-260. CM_ALWON_MCASP1_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-293. CM_ALWON_MCASP1_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

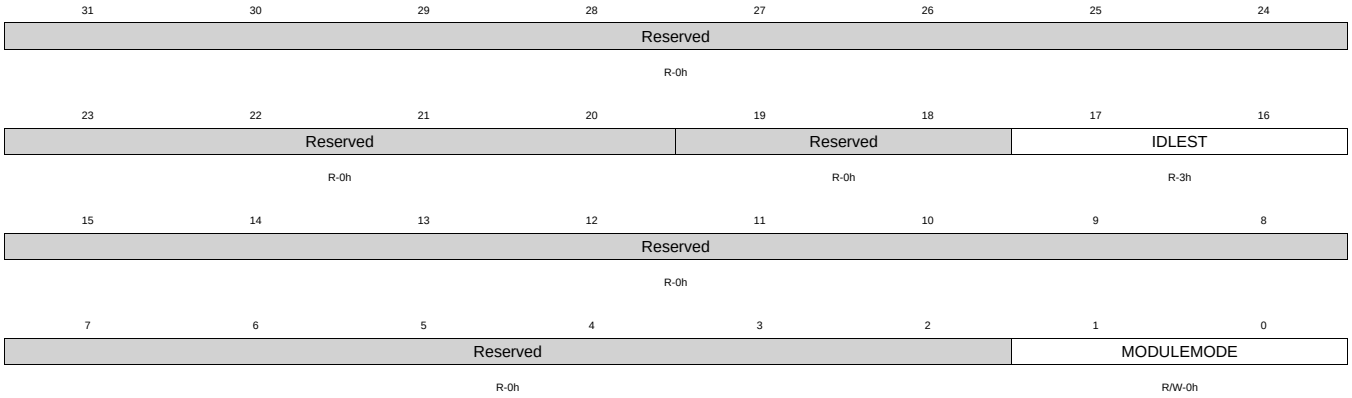

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.16 CM_ALWON_MCASP2_CLKCTRL Register (offset = 148h) [reset = 30000h]

CM_ALWON_MCASP2_CLKCTRL is shown in Figure 2-261 and described in Table 2-294.

This register manages the MCASP2 clocks.

**Figure 2-261. CM_ALWON_MCASP2_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-294. CM_ALWON_MCASP2_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

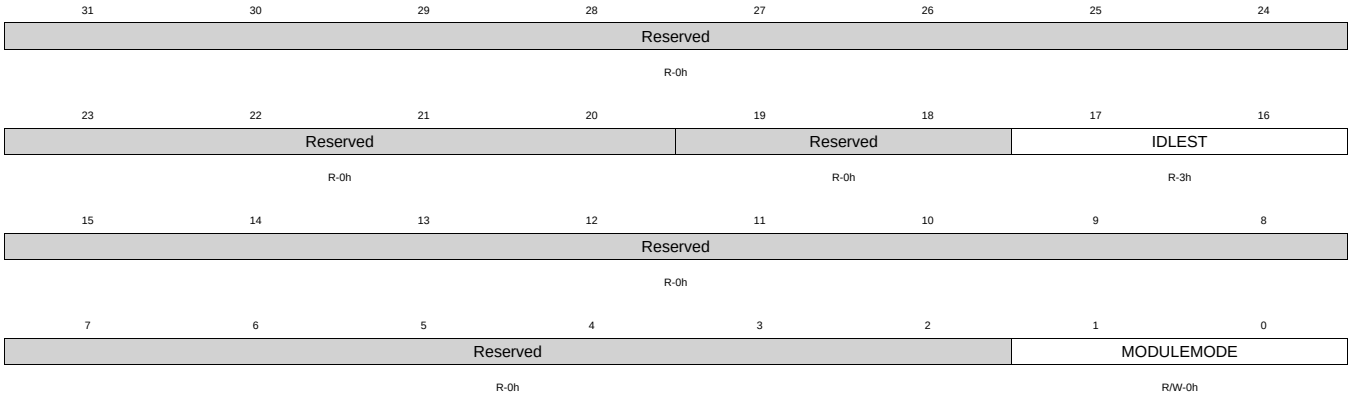

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.17 CM_ALWON_MCBSP_CLKCTRL Register (offset = 14Ch) [reset = 30000h]

CM_ALWON_MCBSP_CLKCTRL is shown in Figure 2-262 and described in Table 2-295.

This register manages the MCBSP clocks.

**Figure 2-262. CM_ALWON_MCBSP_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-295. CM_ALWON_MCBSP_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

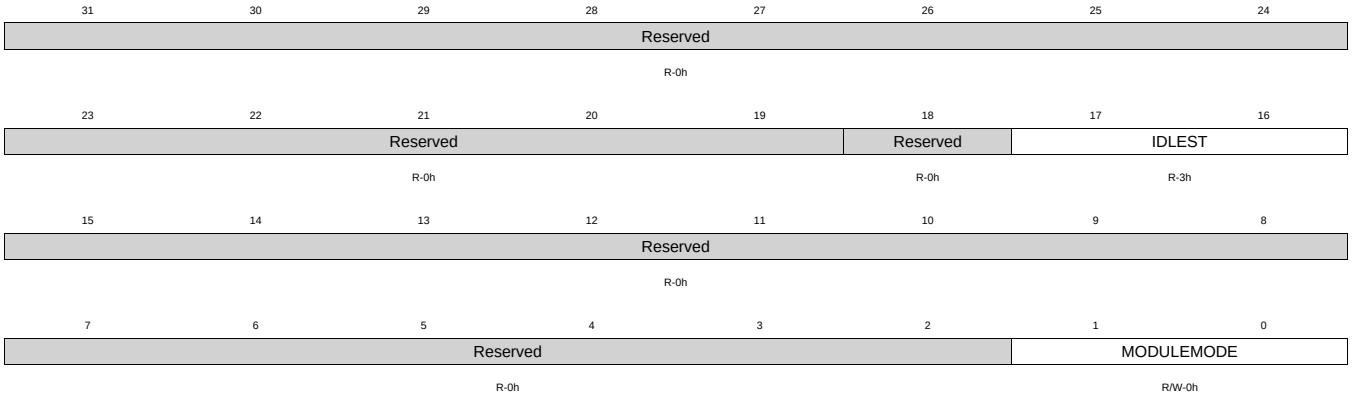

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.18 CM_ALWON_UART_0_CLKCTRL Register (offset = 150h) [reset = 30000h]

CM_ALWON_UART_0_CLKCTRL is shown in Figure 2-263 and described in Table 2-296.

This register manages the UART_0 clocks.

**Figure 2-263. CM_ALWON_UART_0_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-296. CM_ALWON_UART_0_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

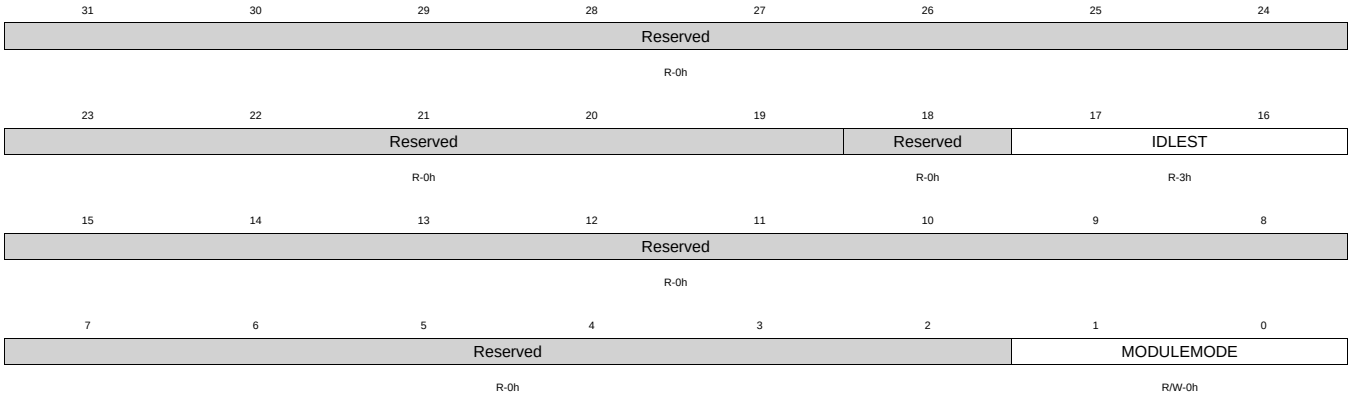

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.19 CM_ALWON_UART_1_CLKCTRL Register (offset = 154h) [reset = 30000h]

CM_ALWON_UART_1_CLKCTRL is shown in Figure 2-264 and described in Table 2-297.

This register manages the UART_1 clocks.

**Figure 2-264. CM_ALWON_UART_1_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-297. CM_ALWON_UART_1_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

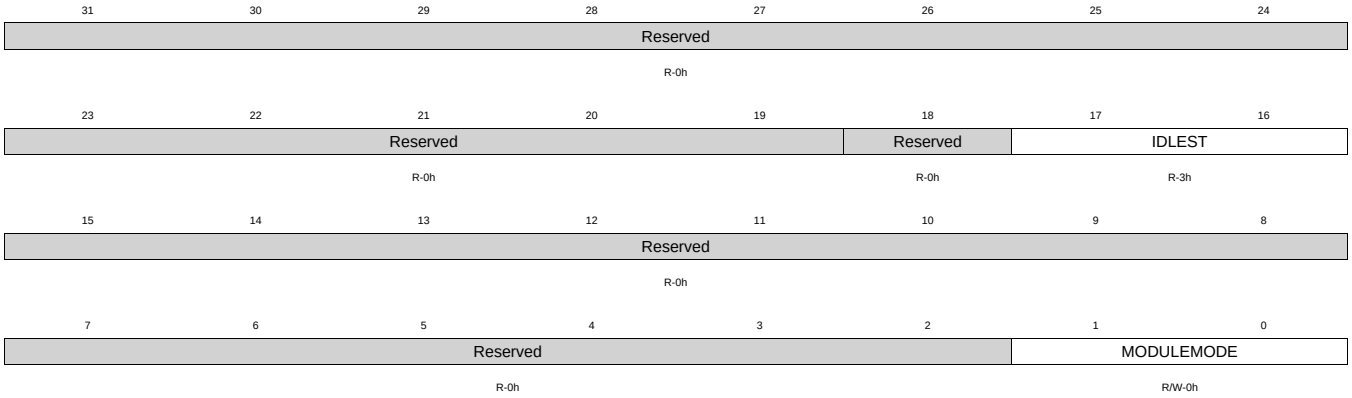

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.20 CM_ALWON_UART_2_CLKCTRL Register (offset = 158h) [reset = 30000h]

CM_ALWON_UART_2_CLKCTRL is shown in Figure 2-265 and described in Table 2-298.

This register manages the UART_2 clocks.

**Figure 2-265. CM_ALWON_UART_2_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-298. CM_ALWON_UART_2_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.21 CM_ALWON_GPIO_0_CLKCTRL Register (offset = 15Ch) [reset = 30100h]

CM_ALWON_GPIO_0_CLKCTRL is shown in Figure 2-266 and described in Table 2-299.

This register manages the GPIO_0 clocks.

**Figure 2-266. CM_ALWON_GPIO_0_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | Reserved | IDLEST | |
| R-0h | | | | | R-0h | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | OPTFCLKEN_DBCLK |
| R-0h | | | | | | | R/W-1h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-299. CM_ALWON_GPIO_0_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-9 | Reserved | R | 0h | |
| 8 | OPTFCLKEN_DBCLK | R/W | 1h | Optional functional clock control.<br>0x0: FCLK_DIS: Optional functional clock is disabled<br>0x1: FCLK_EN: Optional functional clock is enabled |
| 7-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |



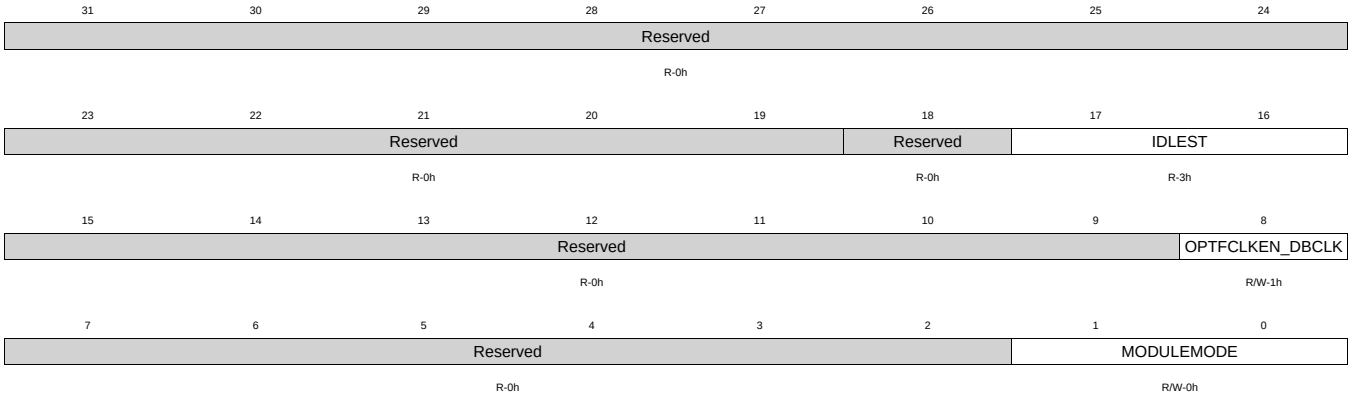
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.22 CM_ALWON_GPIO_1_CLKCTRL Register (offset = 160h) [reset = 30100h]

CM_ALWON_GPIO_1_CLKCTRL is shown in Figure 2-267 and described in Table 2-300.

This register manages the GPIO_1, GPIO_2 and GPIO_3 clocks.

**Figure 2-267. CM_ALWON_GPIO_1_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-300. CM_ALWON_GPIO_1_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-9 | Reserved | R | 0h | |
| 8 | OPTFCLKEN_DBCLK | R/W | 1h | Optional functional clock control.<br>0x0: FCLK_DIS: Optional functional clock is disabled<br>0x1: FCLK_EN: Optional functional clock is enabled |
| 7-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Also, GPIO2 and GPIO3 are clubbed with GPIO1 (For Clocking and Power management).<br>0x3: RESERVED |

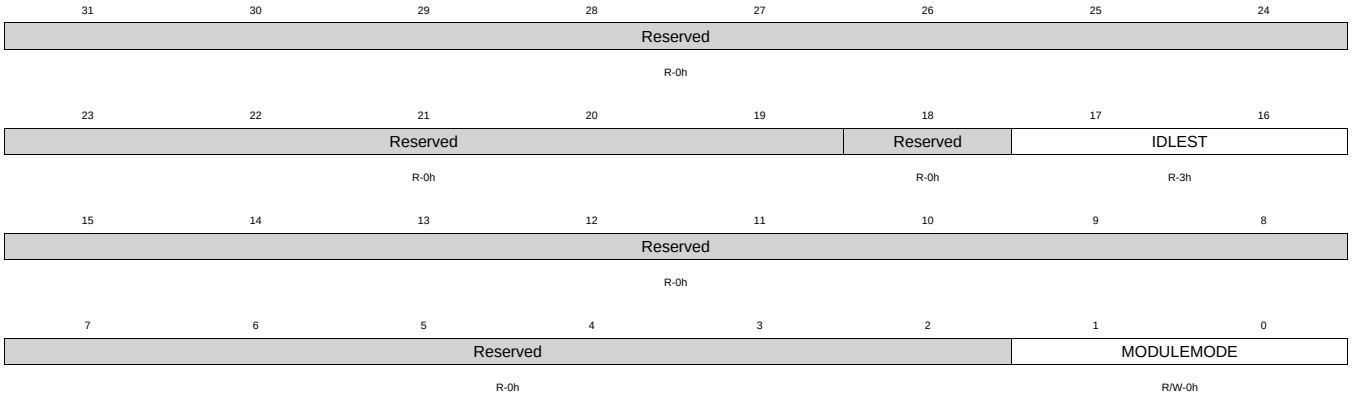

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.23 CM_ALWON_I2C_0_CLKCTRL Register (offset = 164h) [reset = 30000h]

CM_ALWON_I2C_0_CLKCTRL is shown in Figure 2-268 and described in Table 2-301.

This register manages the I2C_0 and I2C_2 clocks.

**Figure 2-268. CM_ALWON_I2C_0_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-301. CM_ALWON_I2C_0_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE : Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

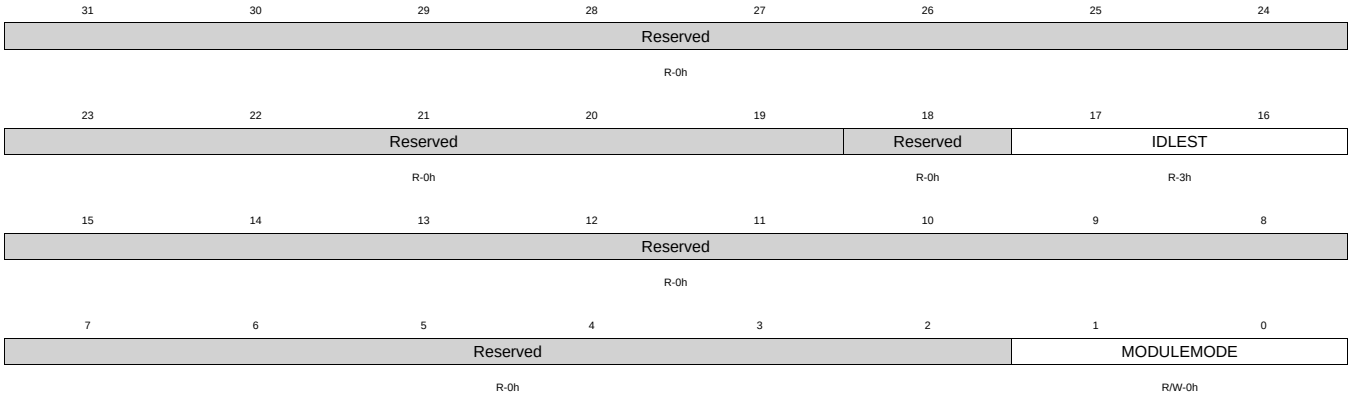

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.24 CM_ALWON_I2C_1_CLKCTRL Register (offset = 168h) [reset = 30000h]

CM_ALWON_I2C_1_CLKCTRL is shown in Figure 2-269 and described in Table 2-302.

This register manages the I2C_1 and I2C_3 clocks.

**Figure 2-269. CM_ALWON_I2C_1_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-302. CM_ALWON_I2C_1_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

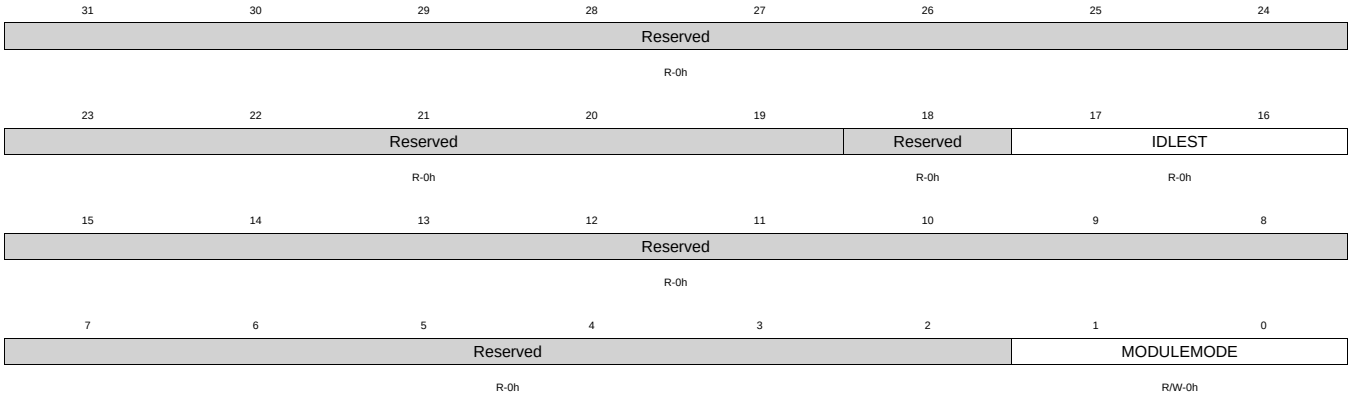

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.25 CM_ALWON_MCASP_3_4_5_CLKCTRL Register (offset = 16Ch) [reset = 30002h]

CM_ALWON_MCASP_3_4_5_CLKCTRL is shown in Figure 2-270 and described in Table 2-303.

This register manages the MCASP_3_4_5 clocks.

**Figure 2-270. CM_ALWON_MCASP_3_4_5_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-303. CM_ALWON_MCASP_3_4_5_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 0h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | RW | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

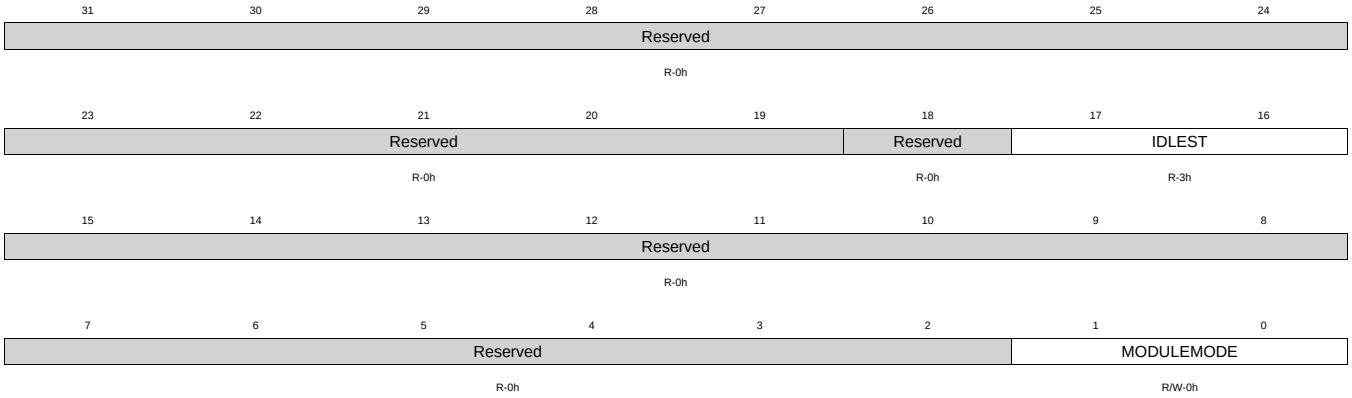

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.26 CM_ALWON_ATL_CLKCTRL Register (offset = 170h) [reset = 30000h]

CM_ALWON_ATL_CLKCTRL is shown in Figure 2-271 and described in Table 2-304.

This register manages the ATL clocks.

**Figure 2-271. CM_ALWON_ATL_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-304. CM_ALWON_ATL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

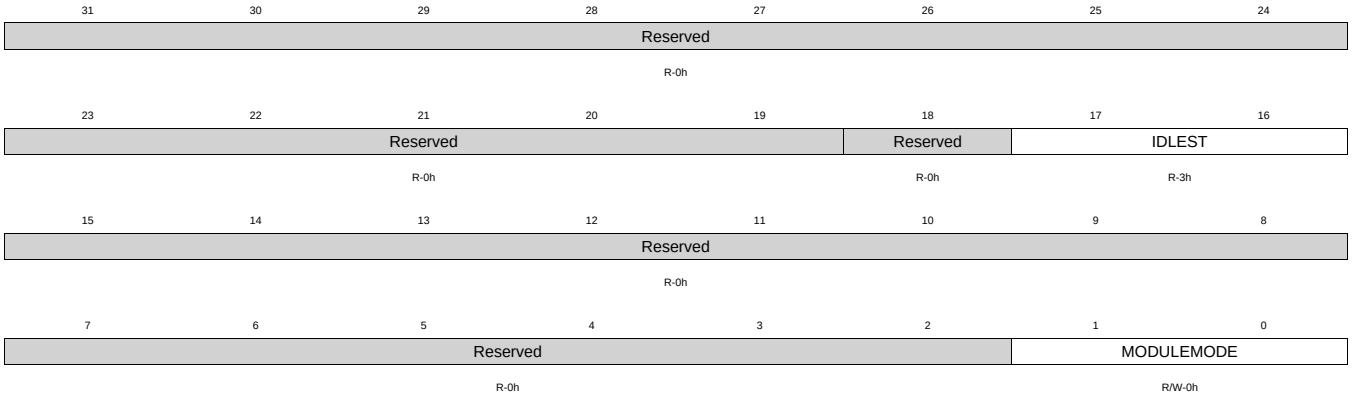

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.27 CM_ALWON_MLB_CLKCTRL Register (offset = 174h) [reset = 30000h]

CM_ALWON_MLB_CLKCTRL is shown in Figure 2-272 and described in Table 2-305.

This register manages the MLB clocks.

**Figure 2-272. CM_ALWON_MLB_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-305. CM_ALWON_MLB_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

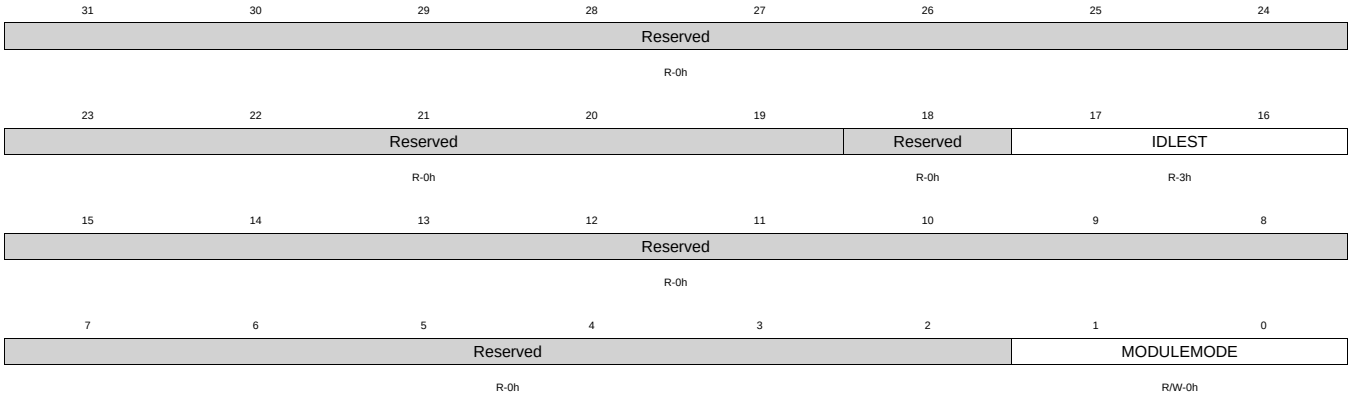

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.28 CM_ALWON_PATA_CLKCTRL Register (offset = 178h) [reset = 30000h]

CM_ALWON_PATA_CLKCTRL is shown in Figure 2-273 and described in Table 2-306.

This register manages the PATA clocks.

**Figure 2-273. CM_ALWON_PATA_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-306. CM_ALWON_PATA_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

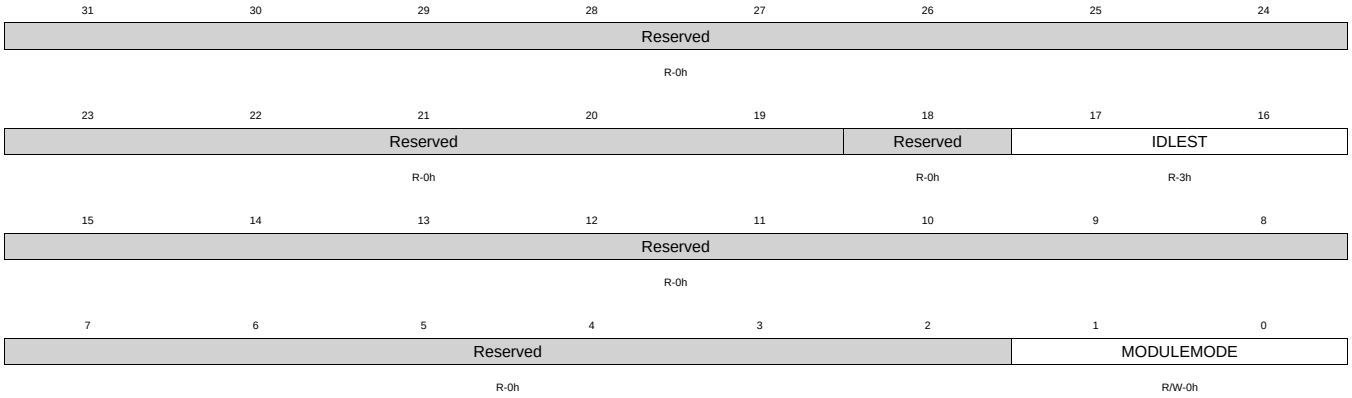

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.29 CM_ALWON_UART_3_CLKCTRL Register (offset = 180h) [reset = 30000h]

CM_ALWON_UART_3_CLKCTRL is shown in Figure 2-274 and described in Table 2-307.

This register manages the UART 3 clocks.

**Figure 2-274. CM_ALWON_UART_3_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-307. CM_ALWON_UART_3_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |



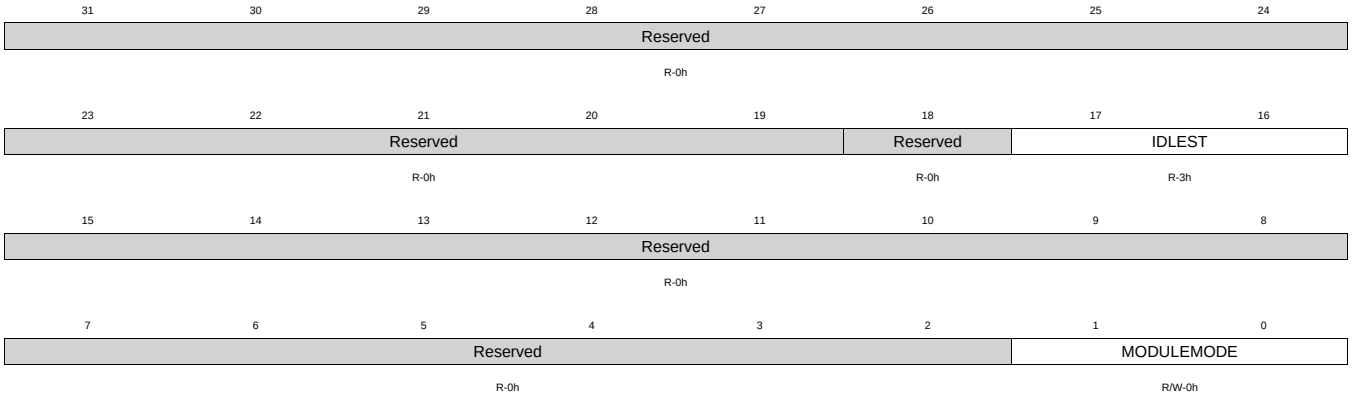
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.30 CM_ALWON_UART_4_CLKCTRL Register (offset = 184h) [reset = 30000h]

CM_ALWON_UART_4_CLKCTRL is shown in Figure 2-275 and described in Table 2-308.

This register manages the UART 4 clocks.

**Figure 2-275. CM_ALWON_UART_4_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-308. CM_ALWON_UART_4_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.31 CM_ALWON_UART_5_CLKCTRL Register (offset = 188h) [reset = 30000h]

CM_ALWON_UART_5_CLKCTRL is shown in Figure 2-276 and described in Table 2-309.

This register manages the UART 5 clocks.

**Figure 2-276. CM_ALWON_UART_5_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | Reserved | IDLEST | |
| R-0h | | | | | R-0h | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-309. CM_ALWON_UART_5_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |



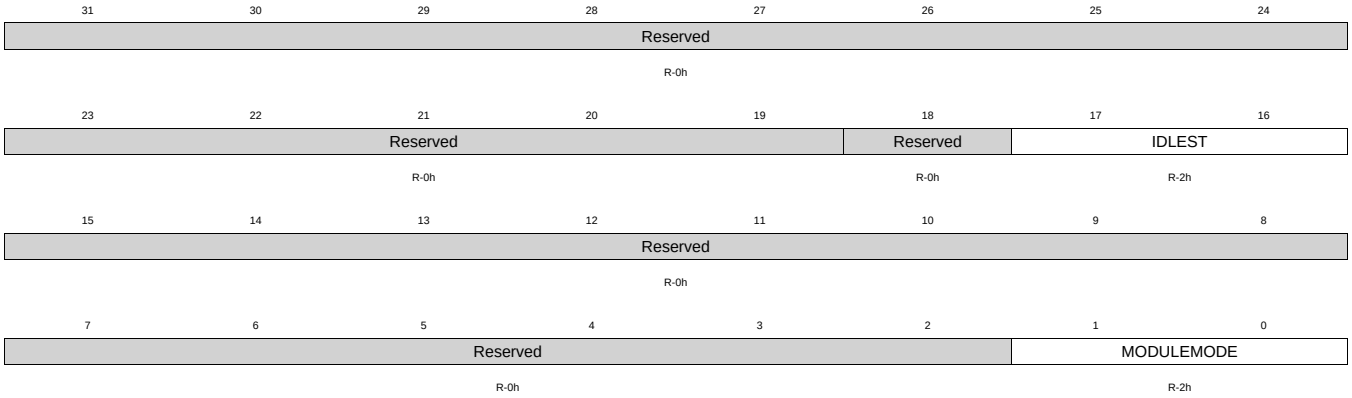
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.32 CM_ALWON_WDTIMER_CLKCTRL Register (offset = 18Ch) [reset = 20002h]

CM_ALWON_WDTIMER_CLKCTRL is shown in Figure 2-277 and described in Table 2-310.

This register manages the WDTIMER clocks.

**Figure 2-277. CM_ALWON_WDTIMER_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-310. CM_ALWON_WDTIMER_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 2h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R | 2h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

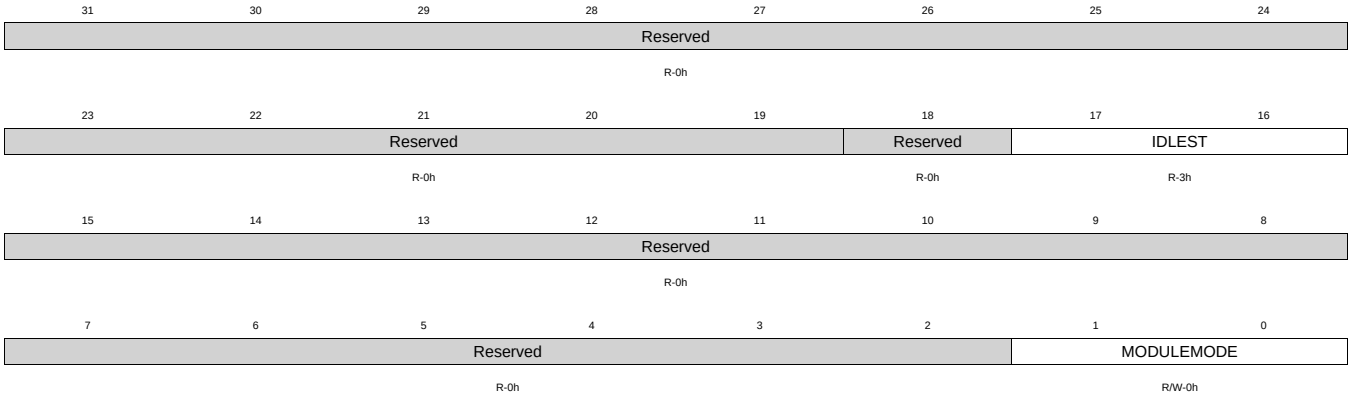

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.33 CM_ALWON_SPI_CLKCTRL Register (offset = 190h) [reset = 30000h]

CM_ALWON_SPI_CLKCTRL is shown in Figure 2-278 and described in Table 2-311.

This register manages the SPI0, SPI1, SPI2 and SPI3 clocks.

**Figure 2-278. CM_ALWON_SPI_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-311. CM_ALWON_SPI_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3 Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

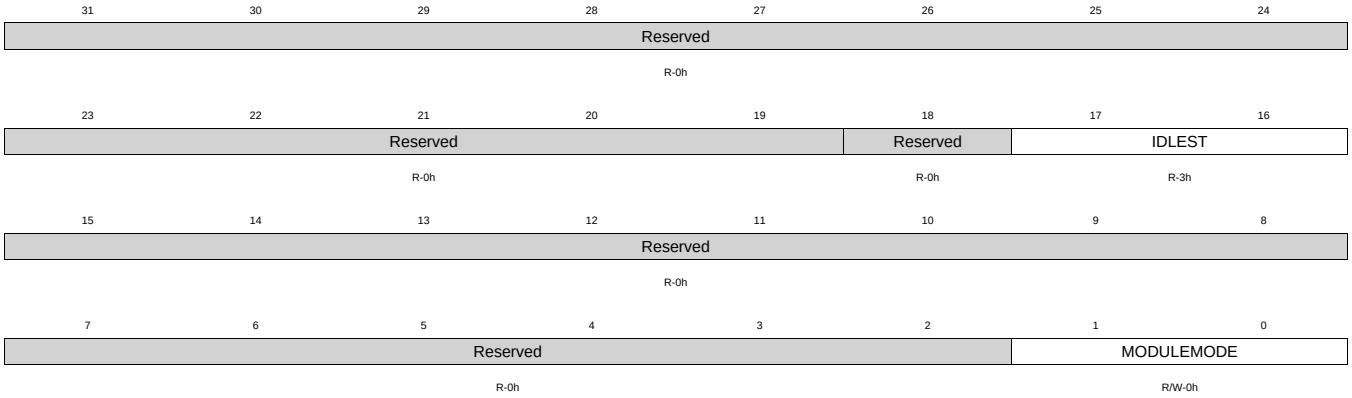

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.34 CM_ALWON_MAILBOX_CLKCTRL Register (offset = 194h) [reset = 30000h]

CM_ALWON_MAILBOX_CLKCTRL is shown in Figure 2-279 and described in Table 2-312.

This register manages the MAILBOX clocks.

**Figure 2-279. CM_ALWON_MAILBOX_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-312. CM_ALWON_MAILBOX_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

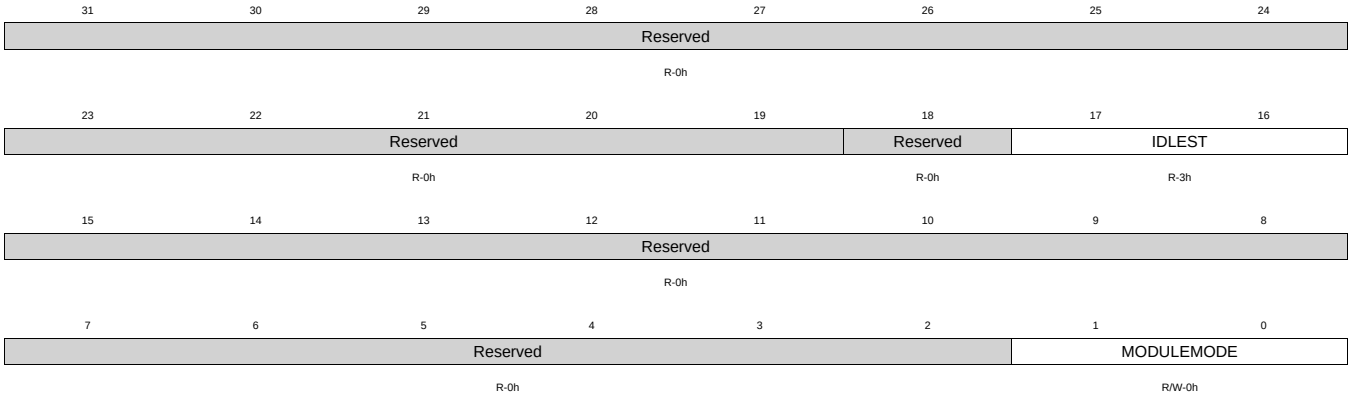

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.35 CM_ALWON_SPINBOX_CLKCTRL Register (offset = 198h) [reset = 30000h]

CM_ALWON_SPINBOX_CLKCTRL is shown in Figure 2-280 and described in Table 2-313.

This register manages the SPINBOX clocks.

**Figure 2-280. CM_ALWON_SPINBOX_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-313. CM_ALWON_SPINBOX_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED : Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

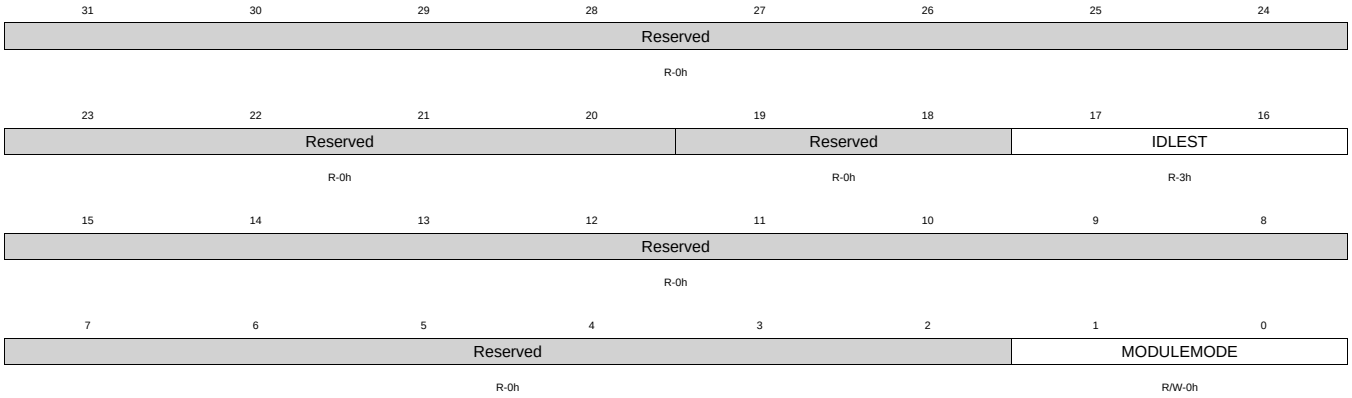

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.36 CM_ALWON_MMUDATA_CLKCTRL Register (offset = 19Ch) [reset = 30000h]

CM_ALWON_MMUDATA_CLKCTRL is shown in Figure 2-281 and described in Table 2-314.

This register manages the MMU data clocks.

**Figure 2-281. CM_ALWON_MMUDATA_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-314. CM_ALWON_MMUDATA_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.37 CM_ALWON_MMUCFG_CLKCTRL Register (offset = 1A8h) [reset = 30000h]

CM_ALWON_MMUCFG_CLKCTRL is shown in Figure 2-282 and described in Table 2-315.

This register manages the MMU config clocks.

**Figure 2-282. CM_ALWON_MMUCFG_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | Reserved | | IDLEST | |
| R-0h | | | | R-0h | | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-315. CM_ALWON_MMUCFG_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.38 CM_ALWON_OCMC_0_CLKCTRL Register (offset = 1B4h) [reset = 30000h]

CM_ALWON_OCMC_0_CLKCTRL is shown in Figure 2-283 and described in Table 2-316.

This register manages the OCMC_0 clocks.

**Figure 2-283. CM_ALWON_OCMC_0_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | IDLEST | |
| R-0h | | | | | | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-316. CM_ALWON_OCMC_0_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle : Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

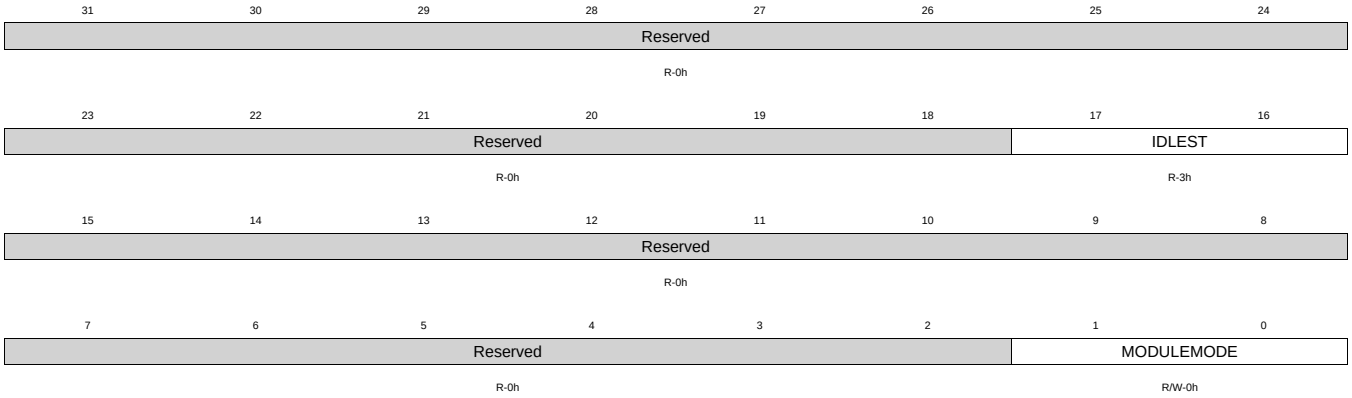

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.39 CM_ALWON_VCP_CLKCTRL Register (offset = 1B8h) [reset = 30000h]

CM_ALWON_VCP_CLKCTRL is shown in Figure 2-284 and described in Table 2-317.

This register manages the VCP clocks.

**Figure 2-284. CM_ALWON_VCP_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-317. CM_ALWON_VCP_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

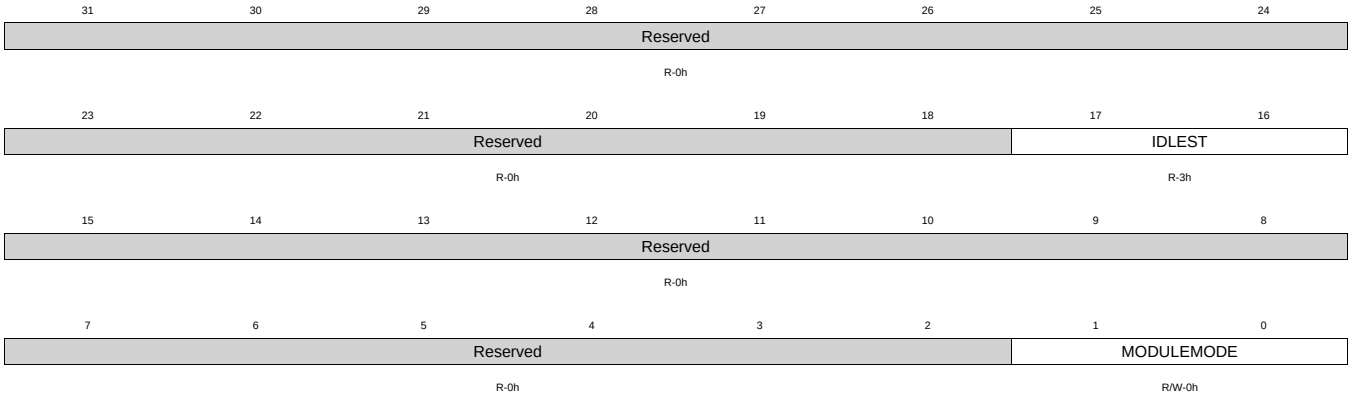

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.40 CM_ALWON_CONTROL_CLKCTRL Register (offset = 1C4h) [reset = 30000h]

CM_ALWON_CONTROL_CLKCTRL is shown in Figure 2-285 and described in Table 2-318.

This register manages the CONTROL clocks.

**Figure 2-285. CM_ALWON_CONTROL_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-318. CM_ALWON_CONTROL_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

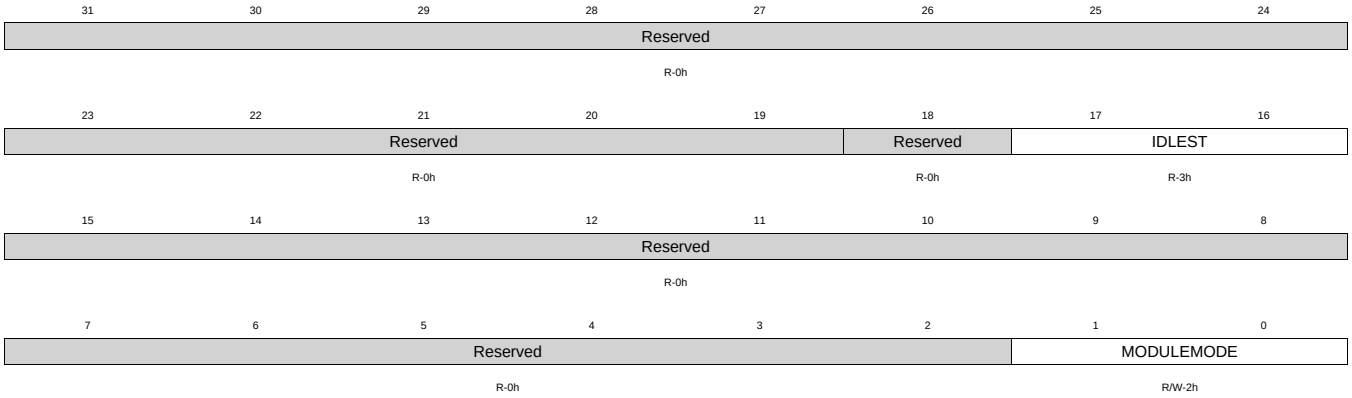

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.41 CM_ALWON_GPMC_CLKCTRL Register (offset = 1D0h) [reset = 30002h]

CM_ALWON_GPMC_CLKCTRL is shown in Figure 2-286 and described in Table 2-319.

This register manages the GPMC clocks.

**Figure 2-286. CM_ALWON_GPMC_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-319. CM_ALWON_GPMC_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 2h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.42 CM_ALWON_ETHERNET_0_CLKCTRL Register (offset = 1D4h) [reset = 70000h]

CM_ALWON_ETHERNET_0_CLKCTRL is shown in Figure 2-287 and described in Table 2-320.

This register manages the ETHERNET_0 switch clocks including both Ethernet ports.

**Figure 2-287. CM_ALWON_ETHERNET_0_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | STBYST | IDLEST | |
| R-0h | | | | | R-1h | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-320. CM_ALWON_ETHERNET_0_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

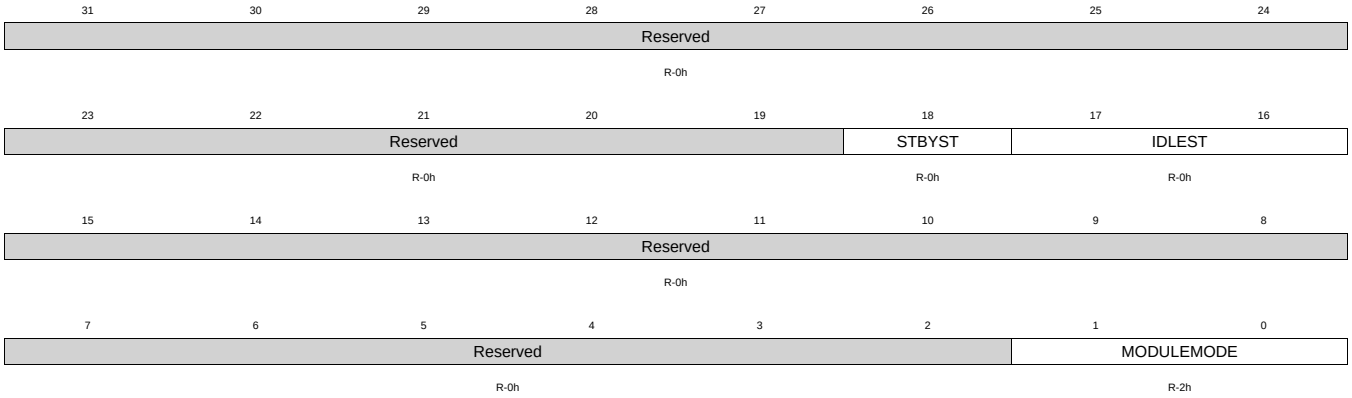

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.43 CM_ALWON_MPU_CLKCTRL Register (offset = 1DCh) [reset = 2h]

CM_ALWON_MPU_CLKCTRL is shown in Figure 2-288 and described in Table 2-321.

This register manages the MPU clocks.

**Figure 2-288. CM_ALWON_MPU_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-321. CM_ALWON_MPU_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | STBYST | R | 0h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 0h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R | 2h | Control the way mandatory clocks are managed.<br>0x0: Reserved<br>0x1: Reserved<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.44 CM_ALWON_DEBUGSS_CLKCTRL Register (offset = 1E0h) [reset = 12500302h]

CM_ALWON_DEBUGSS_CLKCTRL is shown in Figure 2-289 and described in Table 2-322.

This register manages the DEBUGSS clocks.

**Figure 2-289. CM_ALWON_DEBUGSS_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | STM_PMD_CLKDIVSEL | | | TRC_PMD_CLKDIVSEL | | |
| R-0h | | R/W-2h | | | R/W-2h | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| STM_PMD_CLKSEL | | TRC_PMD_CLKSEL | | Reserved | STBYST | IDLEST | |
| R/W-1h | | R/W-1h | | R-0h | R-0h | R-0h | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| Reserved | | | | | | OPTCLK_DEBUG_CLKA | OPTCLK_DEBUG_SYSCLK |
| R-0h | | | | | | R/W-1h | R/W-1h |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R-2h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-322. CM_ALWON_DEBUGSS_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29-27 | STM_PMD_CLKDIVSEL | R/W | 2h | STM clock divider control<br>0x0: Reserved<br>0x1: CLKDIV_1: Divide By 1<br>0x2: CLKDIV_2: Divide By 2<br>0x3: Reserved<br>0x4: CLKDIV_4: Divide By 4<br>0x5: Reserved<br>0x6: Reserved<br>0x7: Reserved |
| 26-24 | TRC_PMD_CLKDIVSEL | R/W | 2h | TPIU clock divider control<br>0x0: Reserved<br>0x1: CLKDIV_1: Divide By 1<br>0x2: CLKDIV_2: Divide By 2<br>0x3: Reserved<br>0x4: CLKDIV_4: Divide By 4<br>0x5: Reserved<br>0x6: Reserved<br>0x7: Reserved |
| 23-22 | STM_PMD_CLKSEL | R/W | 1h | Trace clock source selection for STM<br>0 = SYS_CLK: 00 - Sys Clk<br>1 = REF_CLKA: 01 - Ref. Clk A<br>2 = Reserved<br>3 = Reserved |
| 21-20 | TRC_PMD_CLKSEL | R/W | 1h | Trace clock source selection for TPIU<br>0x0: SYS_CLK: 00 - Sys Clk<br>0x1: REF_CLKA: 01 - Ref. Clk A<br>0x2: Reserved<br>0x3: Reserved |
| 19 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-322. CM_ALWON_DEBUGSS_CLKCTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | STBYST | R | 0h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 0h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-10 | Reserved | R | 0h | |
| 9 | OPTCLK_DEBUG_CLKA | R/W | 1h | Optional functional clock control.<br>0x0 FCLK_DIS: Optional functional clock is disabled<br>0x1: FCLK_EN: Optional functional clock is enabled |
| 8 | OPTCLK_DEBUG_SYSCLK | R/W | 1h | Optional functional clock control.<br>0x0: FCLK_DIS: Optional functional clock is disabled<br>0x1: FCLK_EN: Optional functional clock is enabled |
| 7-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R | 2h | Control the way mandatory clocks are managed.<br>0x0: Reserved<br>0x1: Reserved<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: Reserved |

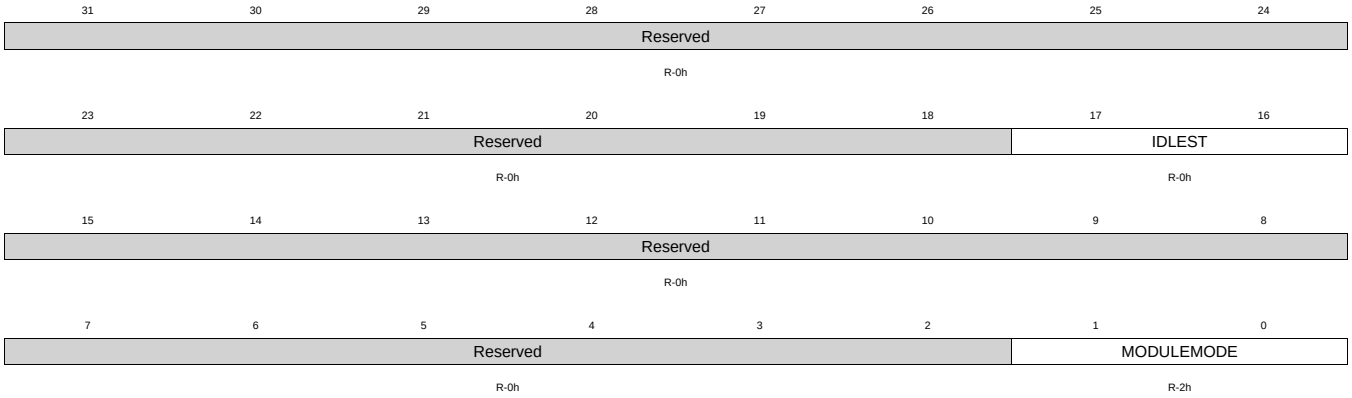

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.45 CM_ALWON_L3_CLKCTRL Register (offset = 1E4h) [reset = 2h]

CM_ALWON_L3_CLKCTRL is shown in Figure 2-290 and described in Table 2-323.

This register manages the L3 clocks.

**Figure 2-290. CM_ALWON_L3_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-323. CM_ALWON_L3_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 0h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R | 2h | Control the way mandatory clocks are managed.<br>0x0: Reserved<br>0x1: Reserved<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.46 CM_ALWON_L4HS_CLKCTRL Register (offset = 1E8h) [reset = 2h]

CM_ALWON_L4HS_CLKCTRL is shown in Figure 2-291 and described in Table 2-324.

This register manages the L4HS clocks.

**Figure 2-291. CM_ALWON_L4HS_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | IDLEST | |
| R-0h | | | | | | R-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R-2h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-324. CM_ALWON_L4HS_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 0h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R | 2h | Control the way mandatory clocks are managed.<br>0x0: Reserved<br>0x1: Reserved<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: Reserved |



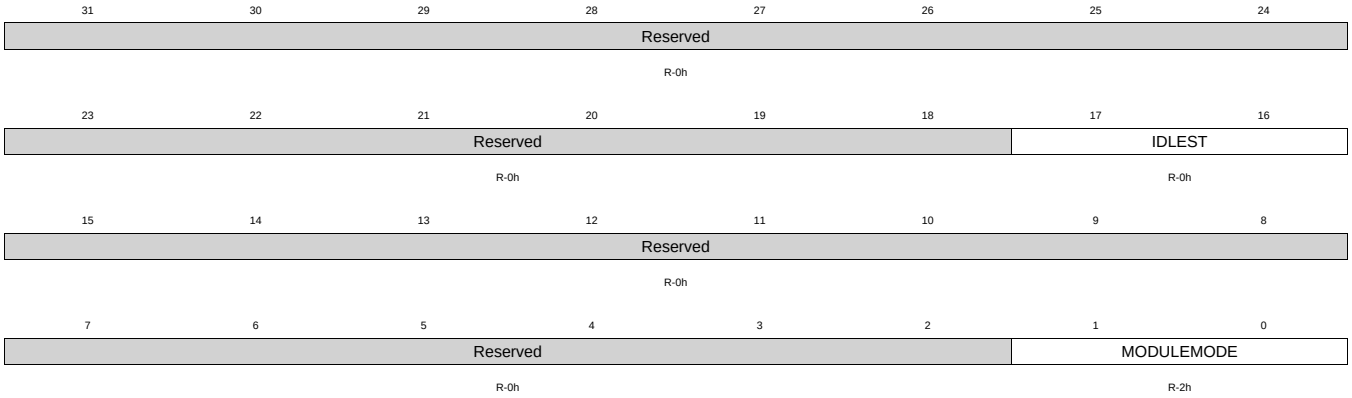
Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.47 CM_ALWON_L4LS_CLKCTRL Register (offset = 1ECh) [reset = 2h]

CM_ALWON_L4LS_CLKCTRL is shown in Figure 2-292 and described in Table 2-325.

This register manages the L4LS clocks.

**Figure 2-292. CM_ALWON_L4LS_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-325. CM_ALWON_L4LS_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 0h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R | 2h | Control the way mandatory clocks are managed.<br>0x0: Reserved<br>0x1: Reserved<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: Reserved |

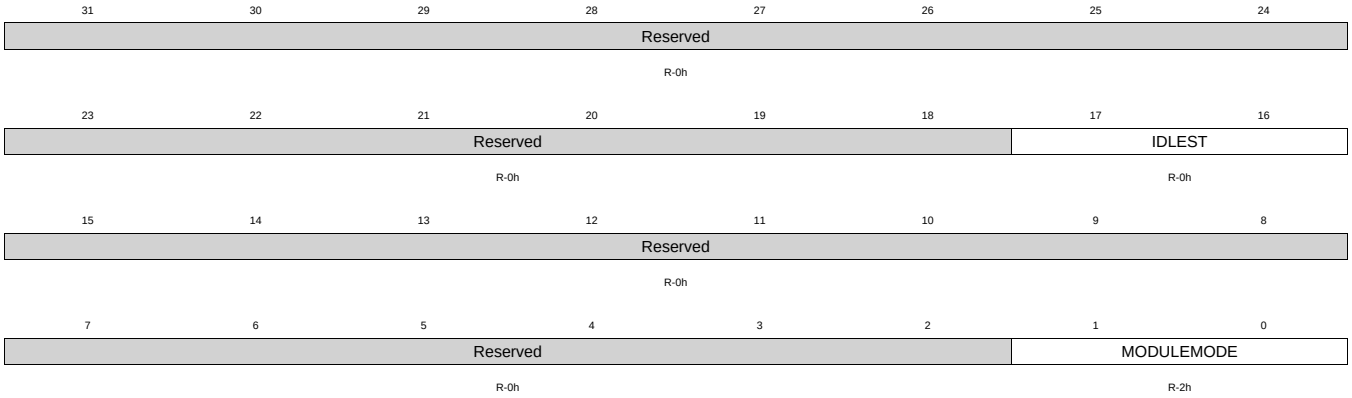

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.48 CM_ALWON_RTC_CLKCTRL Register (offset = 1F0h) [reset = 2h]

CM_ALWON_RTC_CLKCTRL is shown in Figure 2-293 and described in Table 2-326.

This register manages the RTC clocks.

**Figure 2-293. CM_ALWON_RTC_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-326. CM_ALWON_RTC_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 0h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R | 2h | Control the way mandatory clocks are managed.<br>0x0: Reserved<br>0x1: Reserved<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: Reserved |

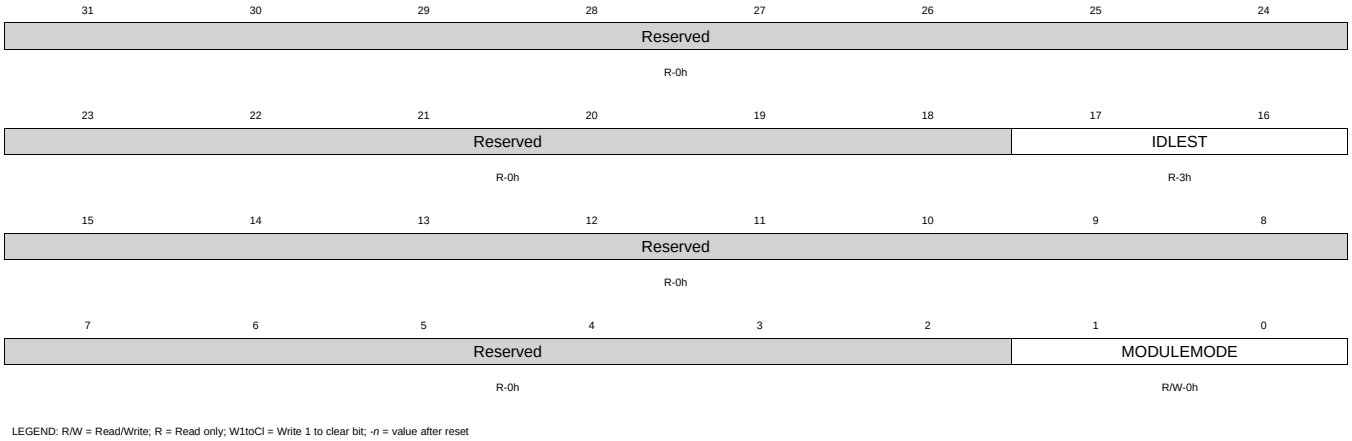

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.49 CM_ALWON_TPCC_CLKCTRL Register (offset = 1F4h) [reset = 30000h]

CM_ALWON_TPCC_CLKCTRL is shown in Figure 2-294 and described in Table 2-327.

This register manages the TPCC clocks.

**Figure 2-294. CM_ALWON_TPCC_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-327. CM_ALWON_TPCC_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.50 CM_ALWON_TPTC0_CLKCTRL Register (offset = 1F8h) [reset = 70000h]

CM_ALWON_TPTC0_CLKCTRL is shown in Figure 2-295 and described in Table 2-328.

This register manages the TPTC_0 clocks.

**Figure 2-295. CM_ALWON_TPTC0_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | Reserved | STBYST | IDLEST | |
| R-0h | | | | R-0h | R-1h | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-328. CM_ALWON_TPTC0_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE : Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |



Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.51 CM_ALWON_TPTC1_CLKCTRL Register (offset = 1FCh) [reset = 70000h]

CM_ALWON_TPTC1_CLKCTRL is shown in Figure 2-296 and described in Table 2-329.

This register manages the TPTC_1 clocks.

**Figure 2-296. CM_ALWON_TPTC1_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | Reserved | STBYST | IDLEST | |
| R-0h | | | | R-0h | R-1h | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-329. CM_ALWON_TPTC1_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED : Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

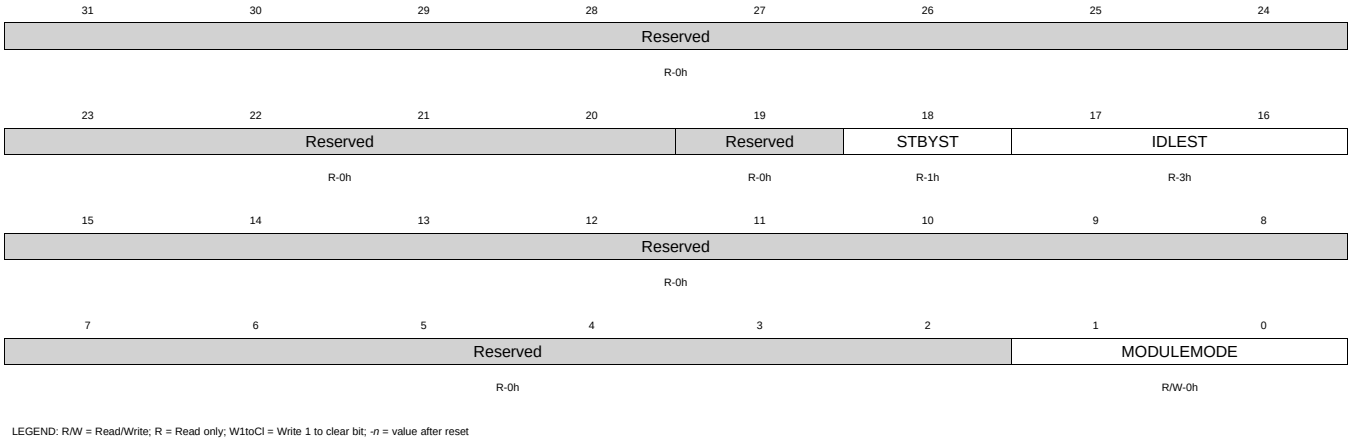

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.52 CM_ALWON_TPTC2_CLKCTRL Register (offset = 200h) [reset = 70000h]

CM_ALWON_TPTC2_CLKCTRL is shown in Figure 2-297 and described in Table 2-330.

This register manages the TPTC_2 clocks.

**Figure 2-297. CM_ALWON_TPTC2_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-330. CM_ALWON_TPTC2_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

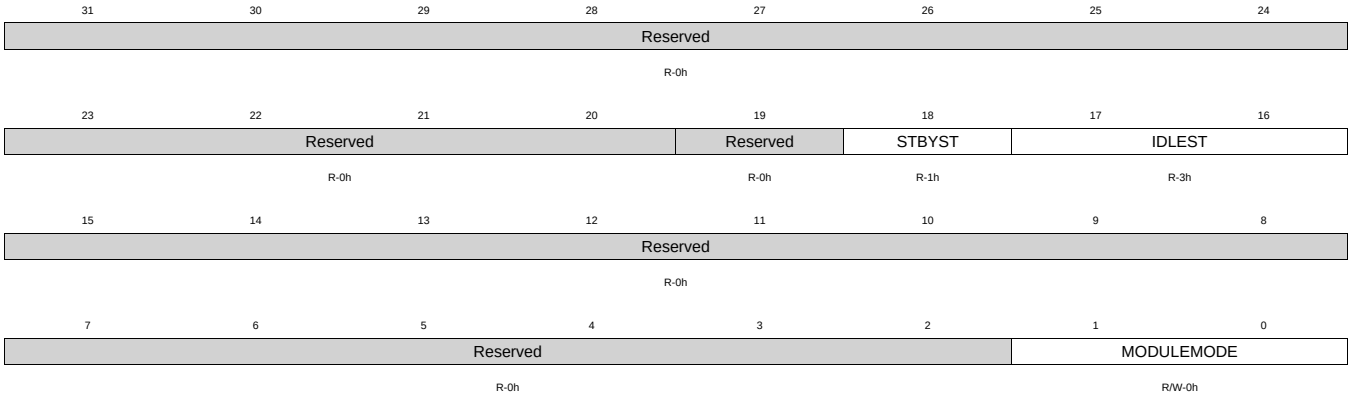

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.53 CM_ALWON_TPTC3_CLKCTRL Register (offset = 204h) [reset = 70000h]

CM_ALWON_TPTC3_CLKCTRL is shown in Figure 2-298 and described in Table 2-331.

This register manages the TPTC_3 clocks.

**Figure 2-298. CM_ALWON_TPTC3_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-331. CM_ALWON_TPTC3_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19 | Reserved | R | 0h | |
| 18 | STBYST | R | 1h | Module standby status.<br>0x0: Func: Module is functional (not in standby)<br>0x1: Standby: Module is in standby |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

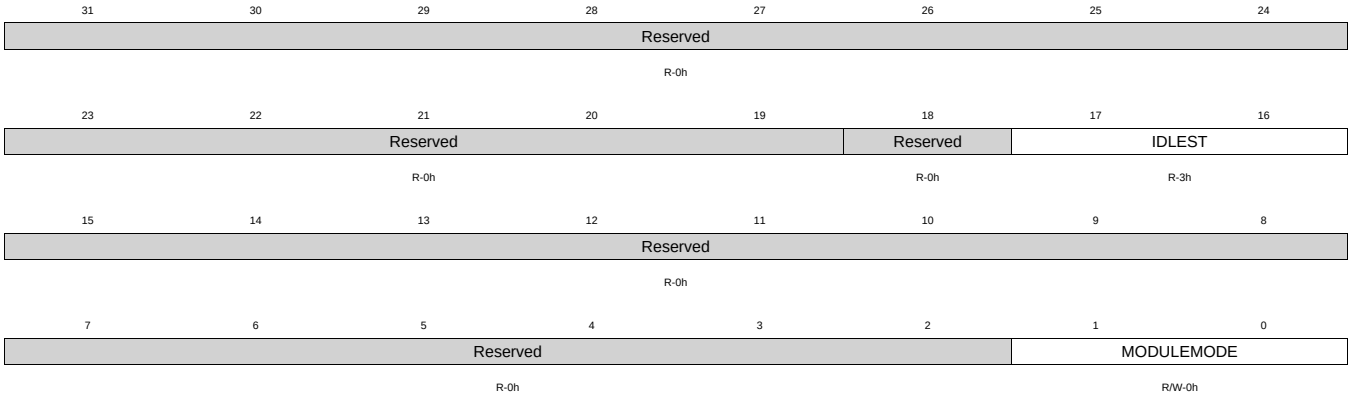

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.54 CM_ALWON_DCAN_0_1_CLKCTRL Register (offset = 218h) [reset = 30000h]

CM_ALWON_DCAN_0_1_CLKCTRL is shown in Figure 2-299 and described in Table 2-332.

This register manages the DCAN_0_1 clocks.

**Figure 2-299. CM_ALWON_DCAN_0_1_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-332. CM_ALWON_DCAN_0_1_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

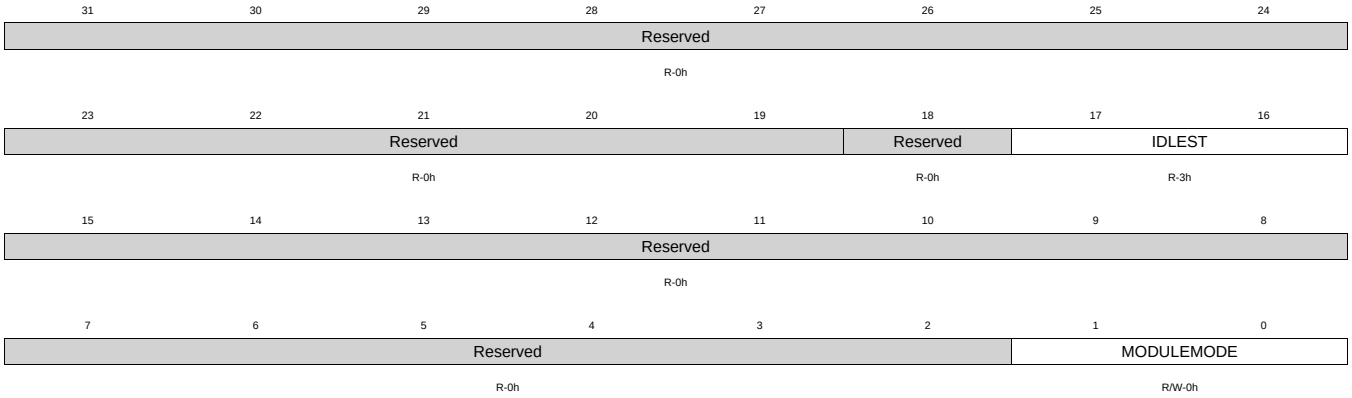

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.55 CM_ALWON_MMCHS_0_CLKCTRL Register (offset = 21Ch) [reset = 30000h]

CM_ALWON_MMCHS_0_CLKCTRL is shown in Figure 2-300 and described in Table 2-333.

This register manages the MMCHS_0 clocks.

**Figure 2-300. CM_ALWON_MMCHS_0_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-333. CM_ALWON_MMCHS_0_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0 Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.56 CM_ALWON_MMCHS_1_CLKCTRL Register (offset = 220h) [reset = 30000h]

CM_ALWON_MMCHS_1_CLKCTRL is shown in Figure 2-301 and described in Table 2-334.

This register manages the MMCHS_1 clocks.

**Figure 2-301. CM_ALWON_MMCHS_1_CLKCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | Reserved | IDLEST | |
| R-0h | | | | | R-0h | R-3h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MODULEMODE | |
| R-0h | | | | | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 2-334. CM_ALWON_MMCHS_1_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

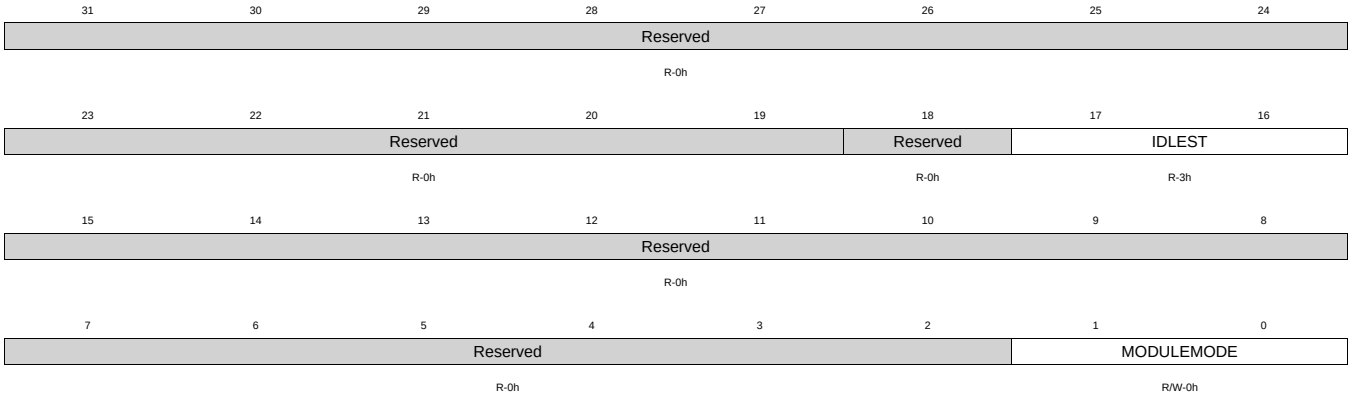

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.57 CM_ALWON_MMCHS_2_CLKCTRL Register (offset = 224h) [reset = 30000h]

CM_ALWON_MMCHS_2_CLKCTRL is shown in Figure 2-302 and described in Table 2-335.

This register manages the MMCHS_2 clocks.

**Figure 2-302. CM_ALWON_MMCHS_2_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-335. CM_ALWON_MMCHS_2_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

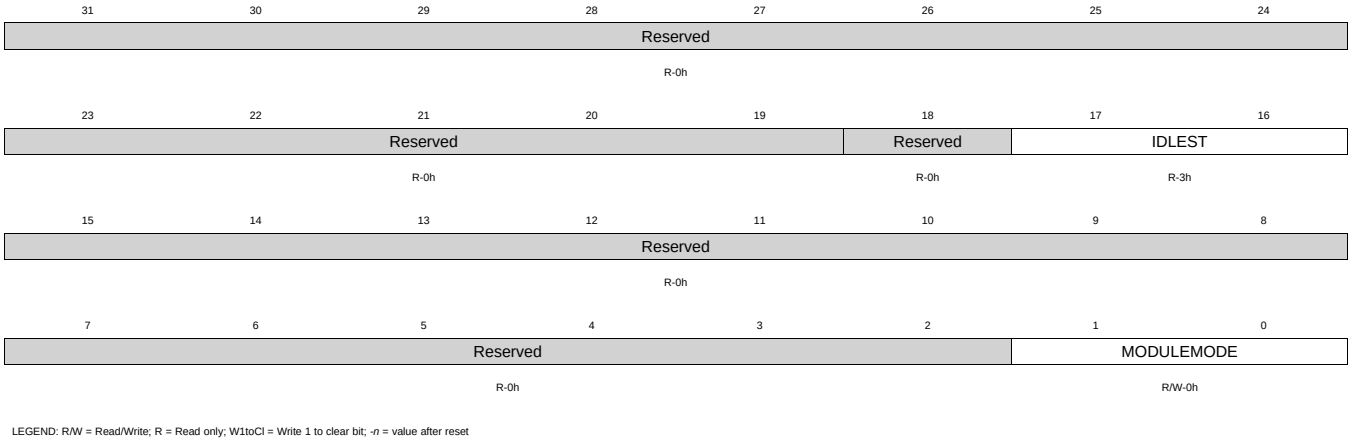

Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.18.58 CM_ALWON_CUST_EFUSE_CLKCTRL Register (offset = 228h) [reset = 30000h]

CM_ALWON_CUST_EFUSE_CLKCTRL is shown in Figure 2-303 and described in Table 2-336.

This register manages the Customer Efuse clocks.

**Figure 2-303. CM_ALWON_CUST_EFUSE_CLKCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-336. CM_ALWON_CUST_EFUSE_CLKCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18 | Reserved | R | 0h | |
| 17-16 | IDLEST | R | 3h | Module idle status.<br>0x0: Func: Module is fully functional, including INTERCONN<br>0x1: Trans: Module is performing transition: wakeup, or sleep, or sleep abortion<br>0x2: Idle: Module is in Idle mode (only INTERCONN part). It is functional if using separate functional clock<br>0x3: Disable: Module is disabled and cannot be accessed |
| 15-2 | Reserved | R | 0h | |
| 1-0 | MODULEMODE | R/W | 0h | Control the way mandatory clocks are managed.<br>0x0: DISABLED: Module is disable by SW. Any INTERCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).<br>0x1: RESERVED<br>0x2: ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.<br>0x3: RESERVED |

## *2.10.19 PRM_ALWON Registers*

Table 2-337 lists the memory-mapped registers for the PRM_ALWON. All register offset addresses not listed in Table 2-337 should be considered as reserved locations and the register contents should not be modified.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 2-337. PRM_ALWON REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 14h | RM_ALWON_RSTST | This register logs the different reset sources of the ALWON domain.<br>Each bit is set upon release of the domain reset signal.<br>Must be cleared by software.<br>[warm reset insensitive] | Section 2.10.19.1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 2.10.19.1 RM_ALWON_RSTST Register (offset = 14h) [reset = 0h]

RM_ALWON_RSTST is shown in Figure 2-304 and described in Table 2-338.

This register logs the different reset sources of the ALWON domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

**Figure 2-304. RM_ALWON_RSTST Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | | | | | | |
| R-0h | R-0h | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | ICECRUSHER_MPU_RST | EMULATION_MPU_RST | Reserved | Reserved | Reserved | Reserved | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R-0h | R-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 2-338. RM_ALWON_RSTST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | Reserved | R | 0h | |
| 14-8 | Reserved | R | 0h | |
| 7 | Reserved | R/W | 0h | |
| 6 | ICECRUSHER_MPU_RST | R/W | 0h | MPU Processor has been reset due to MPU ICECRUSHER1 reset event<br>0x0: RESET_NO : No icecrusher reset<br>0x1: RESET_YES : MPU Processor has been reset upon icecursher reset |
| 5 | EMULATION_MPU_RST | R/W | 0h | MPU Processor has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module<br>0x0: RESET_NO : No emulation reset<br>0x1: RESET_YES : MPU Processor has been reset upon emulation reset |
| 4 | Reserved | R/W | 0h | |
| 3 | Reserved | R | 0h | |
| 2 | Reserved | R | 0h | |
| 1 | Reserved | R | 0h | |
| 0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

# Control Module

This chapter provides information about the control module.

**Topic** **Page**

**3.1 Control Module** ..... **852**
**3.2 CONTROL_MODULE Registers** ..... **856**


Copyright © 2011–2016, Texas Instruments Incorporated

## 3.1 Control Module

This device contains a system-level control logic block. The control module includes status and control logic not addressed within the peripherals or the rest of the device infrastructure.

This module is the primary point of control for the following areas of the device:

- Functional I/O multiplexing
- Device status
- Static device configuration
- Reset isolation logic inside the chip control module
- Lock/unlock mechanism for section of MMR inside the chip control module

The device control module primarily implements a bank of registers accessible (read/write) by the software along with some read-only registers carrying status information.

**Note:** The following operating systems and devices are used interchangeably throughout this chapter:

- DSP and C674x
- MPU and CORTEX-A8

The Control Module registers are documented in the section *CONTROL_MODULE Registers*. The following sections give an overview of the most important functionality provided by the Control Module.

**NOTE:** A processor (or other bus master) must be operating in the Supervisor privilege state in order to write to the control module registers. Registers are readable in either User or Supervisor privilege state.

### *3.1.1 MMR_LOCK Register (Locked/Unlocked States)*

The control module is segmented into 5 regions that are protected by a corresponding MMR_LOCKx register. The intent of the LOCK/UNLOCK mechanism is to prevent critical registers from being corrupted by undesired or malicious writes to the memory mapped addresses.

These regions default to LOCKED state, at which time the corresponding MMRs cannot be written. In order to write to the MMRs of the corresponding region, software must first UNLOCK the region by writing a specified 32-b value to the MMR_LOCKx register. At that point, the MMRs can be successfully written. Software should, in general, relock the registers after the necessary accesses have been done.

Table 3-1 summarizes the MMR_LOCKx registers including offset address range that is protected by each register.

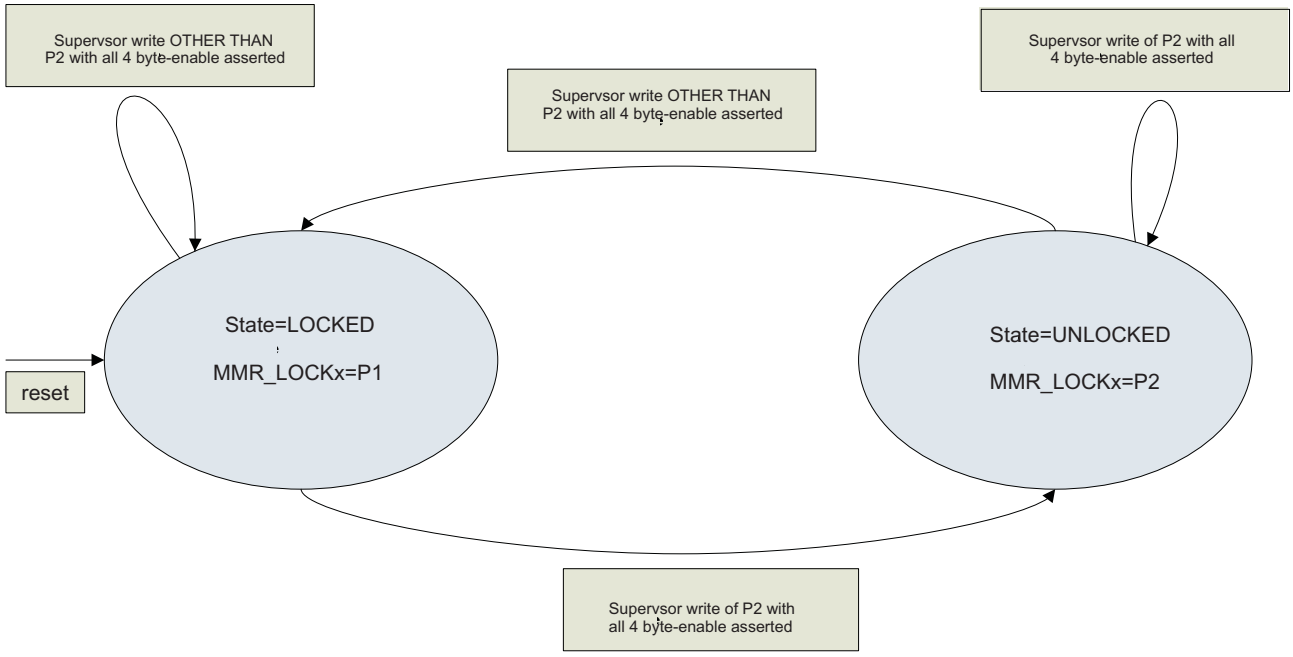
**Table 3-1. MMR_LOCKx Registers**

| Register Name | LOCKED State (register value=P1) | UNLOCKED State (register value=P2) | Start_Addess_x | End_Address_x |
|---|---|---|---|---|
| MMR_LOCK0 | 0x1A1C 8144 | 0x2FF1 AC2B | 0x0400 | 0x05FF |
| MMR_LOCK1 | 0xFDF4 5530 | 0xF757 FDC0 | 0x0600 | 0x07FF |
| MMR_LOCK2 | 0x1AE6 E320 | 0xE2BC 3A6D | 0x0800 | 0x0FFF |
| MMR_LOCK3 | 0x2FFA 927C | 0x1EBF 131D | 0x1000 | 0x12FF |
| MMR_LOCK4 | 0x143F 832C | 0x6F36 1E05 | 0x1300 | 0x17FF |


Copyright © 2011–2016, Texas Instruments Incorporated

Figure 3-1 shows the transition from locked to unlocked state for the MMR_LOCK register.

**Figure 3-1. MMR_LOCKx Transition from LOCKED/UNLOCKED State**




Note: Non-supervisor writes should not reach this register.
Reads should always return the value P1 or P2, depending on the state.

### 3.1.2 Pin Multiplexing (Pin Mux)

Many I/O pin can be internally routed to/from one of several peripheral modules within the device. This routing is referred to as Pin Multiplexing. Pin Multiplexing allows software to choose the subset of internal signals which will be mapped to balls of the device for a given application.

Device-level pin multiplexing is controlled on a pin-by-pin basis by the MUXMODE bits of the PINCNTL1 – PINCNTL270 registers in this control module. The PINCNTLx registers' descriptions and functions can be found in the device data manual, *Pin Multiplexing Control* section.

A second level of pin selection is provided via specific control module registers. The secondary control is described in the following sections.

#### 3.1.2.1 Other Pin Muxing Control

The GPIO_MUX register is used to select whether specific GPIO pins are sourced from device inputs or from internal interrupt sources. Refer to the register description for details.

The GMII_SEL register is used to select between GMII/MII, RMII, and RGMII modes of operation.

#### 3.1.2.2 Special Muxing of GPIO1[7:10] Signals

The GPIO1[7:10] signals can be mapped on the following device balls:

- GPIO1[7] - on device ball W1
- GPIO1[8] - on device ball W2
- GPIO1[9] - on device ball V1
- GPIO1[10] - on device ball V2

To ensure this mapping, the user must not only configure the PINCTNL[62:65] registers properly, but also make the following settings in the MLBP_SIG_IO_CTRL and MLBP_DAT_IO_CTRL registers:

- MLBP_SIG_IO_CTRL register:


Copyright © 2011–2016, Texas Instruments Incorporated

  - Set bit [2] ENLVCMOS to '1' to map GPIO1[7] on device ball W1 and GPIO1[8] on device ball W2.
  - Set bit [1] ENN to '0' to enable GPIO1[8] as output; or set this bit to '1' to enable GPIO1[8] as input.
  - Set bit [0] ENP to '0' to enable GPIO1[7] as output; or set this bit to '1' to enable GPIO1[7] as input.
- MLBP_DAT_IO_CTRL register:
  - Set bit [2] ENLVCMOS to '1' to map GPIO1[9] on device ball V1 and GPIO1[10] on device ball V2.
  - Set bit [1] ENN to '0' to enable GPIO1[10] as output; or set this bit to '1' to enable GPIO1[10] as input.
  - Set bit [0] ENP to '0' to enable GPIO1[9] as output; or set this bit to '1' to enable GPIO1[9] as input.

**NOTE:** The GPIO[7:10] signals can also be mapped on other device balls. The MLBP_SIG_IO_CTRL and MLBP_DAT_IO_CTRL settings are only necessary for mapping GPIO1[7:10] to the [W1, W2, V1, V2] device balls. For more information about device balls mapping, see device data manual.

### 3.1.3 BOOT Status

The Control Module CONTROL_STATUS and BOOTSTAT register allows the software to poll the status of the device boot to determine which boot sequence was detected and if any boot errors persist.

This register can be used by software for verification of a (un)successful boot and if necessary display error messages to the user.

### 3.1.4 DSP / EDMA Memory Management Unit Configuration

The MMU_CFG register is used to optionally route certain initiators through the System MMU. The C674x_DSP MDMA port accesses will always be routed through the System MMU so no configuration bit is instantiated for C674x_DSP. Note that when TPTC0 is routed through the MMU, accesses to the C674x_DSP SDMA port will still by-pass the MMU for increased performance. MMU_CFG Additional details are provided in the *System MMU* chapter.

### 3.1.5 Interrupt and DMA Controller Crossbars

The device contains a large number of system interrupts that are triggered by on chip events. Each CPU has an interrupt controller which allows a limited number of system events to trigger a CPU interrupt.

For the device DSP and Media Controller, to accommodate CPU interrupt generation by all system events, an Interrupt Event Crossbar precedes their corresponding interrupt controllers. To allow a multiple DMA request sources mapping versus each line of the device EDMA subsystem, a DMA Event Crossbar precedes the EDMA3CC controller. This allows a flexible mapping where any input to the Interrupt / DMA event Crossbar can be mapped to any of the interrupt/event outputs of the crossbar (which maps to inputs to the processor interrupt controller or device DMA controller).

This functionality is controlled by the DSP_INTMUX, Media_Controller_INTMUX and EDMA3CC_EVTMUX registers, which format is briefly described in below sections.

#### 3.1.5.1 DSP Interrupt Mux

Each DSP Interrupt mux is controlled by a MMR field which is 6 bit in length (of which only the 5 LSBs are used) .This field value descriptions are available in table *DSP INTC Interrupt Crossbar Mapping* of the section, *Interrupt Requests to DSP INTC*.

This MMR field DSP_INT_MUX_[#int_number], where #int_number is the interrupt number at GEM boundary.There is not interrupt mux for GEM interrupt number a) 0 to 14 and b) 96 to 127 so the range of #int_number is 15 to 95.


Copyright © 2011–2016, Texas Instruments Incorporated

After reset DSP_INT_MUX_[#int_number] defaults to 0b000000, which maps the interrupt from default mapping to interrupt_[#int_number], e.g. interrupt_15's default mapping is MMCHS0's SDINT, interrupt_16's default mapping is McSPI0's SPIINT etc. The mapping for other values of DSP_INT_MUX_[#int_number] and additional details, including interrupt/event mapping, are provided in the *Device Interrupts and EDMA Events* section.

#### 3.1.5.2 Media Controller Interrupt Mux

Each Media Controller interrupt mux is controlled by a MMR field which is 6 bit in length. This field value descriptions are available in table *Media Controller INTC Interrupt Crossbar Mapping* of the section, *Interrupt Requests to Media Controller*.

This MMR field Media_Controller_INT_MUX_[#int_number], where #int_number is the interrupt number at Media Controller boundary. The range of #int_number is 0 to 56.

After reset Media_Controller_INT_MUX_[#int_number] defaults to 0b000000, which maps the interrupt from default mapping to Media_Controller_interrupt_[#int_number], e.g. interrupt_0's default mapping is HDMI's HDMIINT, interrupt_2's default mapping is Timer1's TINT1, etc. The mapping for other values of Media_Controller_INT_MUX_[#int_number] and additional details, including interrupt/event mapping, are provided in the *Device Interrupts and EDMA Events* section.

#### 3.1.5.3 EDMA3CC Event Mux

Each EDMA3CC (TPCC) Event mux is controlled by a MMR field which is 6 bit in length (of which only the 5 LSBs are used). This field value descriptions are available in table *EDMA Multiplexed Synchronization Events* of the section, *EDMA Channel Synchronization Events*.

This MMR field EDMA3CC_MUX_[evt_number], where #evt_number is the event number at EDMA3CC boundary. The range of #evt_number is 0 to 63.

After reset EDMA3CC_MUX_[#evt_number] defaults to 0b000000, which maps the event from default mapping to event_[#evt_number], e.g. EDMA3CC event_0's default mapping is VCP's VCPREVT0, EDMA3CC event_1's default mapping is VCP's VCPXEVT0etc. The mapping for other values of EDMA3CC_MUX_[#evt_number] and additional details, including interrupt/event mapping, are provided in the *Device Interrupts and EDMA Events* section.

### 3.1.6 Device Level Peripheral Configuration

Various peripherals within the device contain various device level configuration that are not contained within the peripheral configuration registers. Device level configuration includes Power Management, MAC Address, USB Configuration / Status Registers, USB Product ID, Vendor ID Registers and PCIe Vendor ID, Product ID Registers.

### 3.1.7 Reset Isolation

The device supports reset isolation for the Ethernet Subsystem. For more information refer to the section *Reset Isolation* of the chapter, *Power, Reset, and Clock Management (PRCM) Module*.

Table 3-2 describes the subset of device pins and the corresponding PINCNTLx registers that can be reset-isolated. The specific combination of pins that are enabled for isolation is controlled by the RESET_ISO register and is detailed in the Register section of this chapter. In summary, the ISO_CONTROL field can enable/disable reset isolation. The GMII_ISO, RMII_ISO0, and RMII_ISO1 fields control which subset of the Pin Groups (and PINCNTL registers) are selected for reset isolation.

**Table 3-2. List of PINCNTLx Registers, reset affected by RESET_ISO.**

| RESET_ISO Pin Group | Associated PINCNTL Registers |
|---|---|
| MDIO/MDCLK | PINCNTL233 to PINCNTL234 |
| EMAC[1]_Mx/Other(1) | PINCNTL204 to PINCNTL227 |
| EMAC[0]_Mx/EMAC[1:0]_RGx | PINCNTL235 to PINCNTL258 |
| EMAC_RMREFCLK | PINCNTL232 |

(1) Several pins not related to the Ethernet Subsystem are included in this Pin Group for device compatibility.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-2. List of PINCNTLx Registers, reset affected by RESET_ISO. (continued)**

| RESET_ISO Pin Group | Associated PINCNTL Registers |
|---|---|
| EMAC[1]_RMx_MUX0 | PINCNTL252 to PINCNTL258 |
| EMAC[1]_RMx_MUX1 | PINCNTL157 to PINCNTL163 |
| EMAC[0]_RMx | PINCNTL236 to PINCNTL242 |

## 3.2 CONTROL_MODULE Registers

Table 3-3 lists the memory-mapped registers for the CONTROL_MODULE. All register offset addresses not listed in Table 3-3 should be considered as reserved locations and the register contents should not be modified.

All these registers can only be modified by SUPERVISR mode access.

**Table 3-3. CONTROL_MODULE REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | CONTROL_REVISION | The CONTROL_REVISION register is a read only register containing the revision number of the Control module. A write to this register has no effect. | Section 3.2.1 |
| 4h | CONTROL_HWINFO | The CONTROL_HWINFO register contains information about the module hardware configuration (typically HDL generics, if any). | Section 3.2.2 |
| 10h | CONTROL_SYSCONFIG | The CONTROL_SYSCONFIG register implements only a reduced set of the register defined in the register implements various standby and emulation control bits. | Section 3.2.3 |
| 40h | CONTROL_STATUS | The CONTROL_STATUS register reflects the system boot and the device type configuration values as sampled when the power-on reset (PORz) or warm reset (RESETz) signal goes high. | Section 3.2.4 |
| 44h | BOOTSTAT | The Boot Status (BOOTSTAT) register indicates the status of the device boot process | Section 3.2.5 |
| 48h | DSPBOOTADDR | The DSP boot address defaults to 0x00800000 (C674x_DSP L2 space). The Host ARM must place DSP boot code into C674x_DSP L2 prior to releasing C674x_DSP reset or change the DSPBOOTADDR to point to the code location. The DSPBOOTADDR register is reset to its default value by a hard reset only and retains its current value over a soft reset. | Section 3.2.6 |
| 60h | MMR_LOCK0 | The MMR_LOCKx Register is used allow/block writes to the registers in address region Start_Addess_x - End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x - End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of MMR_LOCKx registers. | Section 3.2.7 |
| 64h | MMR_LOCK1 | The MMR_LOCKx Register is used allow/block writes to the registers in address region Start_Addess_x - End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x - End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of MMR_LOCKx registers. | Section 3.2.8 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-3. CONTROL_MODULE REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 68h | MMR_LOCK2 | The MMR_LOCKx Register is used allow/block writes to the registers in address region Start_Addess_x - End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x - End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of MMR_LOCKx registers. | Section 3.2.9 |
| 6Ch | MMR_LOCK3 | The MMR_LOCKx Register is used allow/block writes to the registers in address region Start_Addess_x - End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x - End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of MMR_LOCKx registers. | Section 3.2.10 |
| 70h | MMR_LOCK4 | The MMR_LOCKx Register is used allow/block writes to the registers in address region Start_Addess_x - End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x - End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of MMR_LOCKx registers. | Section 3.2.11 |
| 100h | CONTROL_SEC_CTRL | Security Control Register | Section 3.2.12 |
| 468h | DEVOSC | Oscillator 0 Control. Note for deepsleep: - 1st program bit16 (DSPOLARITY) of register DEEPSLEEP_CTRL to the desired polarity - To enter deepsleep mode, set bit 17 (DSENABLE) of DEEPSLEEP_CTRL to 1. - To exit, toggle the external pin DEEPSLEEPZ. | Section 3.2.13 |
| 46Ch | AUXOSC | Oscillator 1 Control | Section 3.2.14 |
| 480h | PCIE_CFG | 5.6.20. PCIE Configuration Register | Section 3.2.15 |
| 600h | DEVICE_ID | The DEVICE_ID register contains a software readable version of the device JTAG ID. Software can use this register to determine the version of the device on which it is executing. | Section 3.2.16 |
| 604h | DEV_FEATURE | The DEV_FEATURE register reflects the eFuse settings enabling or disabling certain device functions and capabilities. Note that DEV_FEATURE bits are all implemented as feature enabled bits. This may require inverting the value of those eFuses which are implemented as disable signals. | Section 3.2.17 |
| 608h | INIT_PRIORITY_0 | The INIT_PRIORITY_0 register configures the infrastructure pressure input for L3 initiators to provide dynamic priority escalation. | Section 3.2.18 |
| 60Ch | INIT_PRIORITY_1 | The INIT_PRIORITY_1 register configures the infrastructure pressure input for L3 initiators to provide dynamic priority escalation. | Section 3.2.19 |
| 610h | MMU_CFG | The MMU_CFG register is used to optionally route certain initiators through the System MMU. The C674x_DSP MDMA port accesses will always be routed through the System MMU so no configuration bit is instantiated for C674x_DSP. Note that when TPTC0 is routed through the MMU accesses to the C674x_DSP SDMA port will still by-pass the MMU for increased performance. | Section 3.2.20 |
| 614h | TPTC_CFG | The TPTC_CFG register configures the default burst size for TPTC0, 1, 2, and 3. | Section 3.2.21 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-3. CONTROL_MODULE REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 61Ch | DSP_IDLE_CFG | The C674x_DSP_IDLE_CFG register controls the state manaC674x_DSPent of the C674x_DSP initiator and target port interfaces. | Section 3.2.22 |
| 620h | USB_CTRL0 | USB Control Register 0 controls the PHY, CM and OTG modules for USB0. | Section 3.2.23 |
| 624h | USB_STS0 | USB Status Register 0 reflects the status of the USB0. | Section 3.2.24 |
| 628h | USB_CTRL1 | USB Control Register 1 controls the PHY, CM and OTG modules for USB1. | Section 3.2.25 |
| 62Ch | USB_STS1 | USB Status Register 1 reflects the status of the USB0. | Section 3.2.26 |
| 630h | MAC_ID0_LO | The MAC_ID0_LO register contains the lower 2 bytes of the 48-bit ID for MAC0. | Section 3.2.27 |
| 634h | MAC_ID0_HI | The MAC_ID0_HI register contains the upper 4 bytes of the 48-bit ID for MAC0. | Section 3.2.28 |
| 638h | MAC_ID1_LO | The MAC_ID1_LO register contains the lower 2 bytes of the 48-bit ID for MAC1. | Section 3.2.29 |
| 63Ch | MAC_ID1_HI | The MAC_ID1_HI register contains the upper 4 bytes of the 48-bit ID for MAC1. | Section 3.2.30 |
| 640h | SW_REVISION | The SW_REVISION register reads the Paper Spin Variant 32bit efuse. The SW can read this bit and decide what to do with a specific SW package. | Section 3.2.31 |
| 644h | DCAN_RAMINIT | The DCAN_RAMINIT register is used to start/check status of the autoinitialization process of DCAN RAM. | Section 3.2.32 |
| 64Ch | AUD_CTRL | The register controls loopback of McBSP data for testing purposes. | Section 3.2.33 |
| 650h | GMII_SEL | Ethernet GMII Select | Section 3.2.34 |
| 654h | OCMEM_PWRDN | The OCMEM_PWRDN Register determines the powerdown behavior of the OCMC RAM memories. | Section 3.2.35 |
| 65Ch | MEDIA_CONTROLLER_MEM_PWRDN | The MEDIA_CONTROLLER_MEM_PWRDN Register determines the powerdown behavior of the Media Controller memories. | Section 3.2.36 |
| 670h | SD_DAC_CTRL | The SD_DAC_CTRL Register controls operation of the Standard Definition video DACs. | Section 3.2.37 |
| 674h | SD_DAC0_CAL | The SD_DAC0_CAL Register is used to calibrate the output level of SD DAC 0 (SD Y). Users can use this register to drive a known value to the DAC input for performing system video level calibration. | Section 3.2.38 |
| 678h | SD_DAC1_CAL | The SD_DAC1_CAL Register is used to calibrate the output level of SD DAC 1 (SD C). Users can use this register to drive a known value to the DAC input for performing system video level calibration. | Section 3.2.39 |
| 67Ch | SD_DAC0_REGCTRL | The SD_DAC0_REGCTRL Register is used to provide an alternate method to program the AVDAC s CTL interface (to be used only when the internal registers of the AVDAC needs to be tweaked). | Section 3.2.40 |
| 680h | SD_DAC0_REGSTATUS | The SD_DAC0_REGSTATUS Register is used to provide to know the status of the last write to the AVDAC s internal registers (to be used only when the internal registers of the AVDAC needs to be tweaked). | Section 3.2.41 |
| 684h | SD_DAC1_REGCTRL | The SD_DAC1_REGCTRL Register is used to provide an alternate method to program the AVDAC s CTL interface (to be used only when the internal registers of the AVDAC needs to be tweaked). | Section 3.2.42 |
| 688h | SD_DAC1_REGSTATUS | The SD_DAC1_REGSTATUS Register is used to provide to know the status of the last write to the AVDAC s internal registers (to be used only when the internal registers of the AVDAC needs to be tweaked). | Section 3.2.43 |
| 694h | EMIF_CLK_GATE | This register provides a mechanism to control clock gating of DDR PHYs and see status of Self Refresh. | Section 3.2.44 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-3. CONTROL_MODULE REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 6A0h | SMRT_CTRL | The SMRT_CTRL Register reflects the control for the srsleep signal for the SRs . | Section 3.2.45 |
| 6A4h | ARM_CORTEX_A8_HW_DBG_SEL | This register controls what kind of debug info is returned by ARM Cortex-A8. | Section 3.2.46 |
| 6A8h | ARM_CORTEX_A8_HW_DBG_INFO | This register shows the debug info bus from ARM Cortex-A8. Please see below for definition. The selection bits are HW_DBG_SEL (3:0) in the MMR ARM_CORTEX_A8_HW_DBG_SEL | Section 3.2.47 |
| 6B0h | PRCM_DEBUG_ALWON_DEFAULT | This register is for PRCM debug. | Section 3.2.48 |
| 6B4h | PRCM_DEBUG_PD_DOMAIN_STATUS | This register is for PRCM debug. | Section 3.2.49 |
| 6D8h | PCIE_PLLCFG0 | This is connected to "CF_APLL_0" on SERDES ANA Initial Settings required prior to power up sequence: PCIE_100M: 4000 7000 ; SATA_20M: C000 7060 ; SATA_100M: 4000 7060 | Section 3.2.50 |
| 6DCh | PCIE_PLLCFG1 | This is connected to "CF_APLL_1" on SERDES ANA Initial Settings required prior to power up sequence: PCIE_100M: 0064 0000 ; SATA_100MHz: 912C 0000 ; SATA_20MHz: C02C 0000 | Section 3.2.51 |
| 6E0h | PCIE_PLLCFG2 | This is connected to "CF_APLL_2" on SERDES ANA Initial Settings required prior to power up sequence: PCIE: 0000 0000 -- SATA: 0000 0000 (both modes) | Section 3.2.52 |
| 6E4h | PCIE_PLLCFG3 | This is connected to "CF_APLL_3" on SERDES ANA Initial Settings required prior to power up sequence: 0040 08E0 | Section 3.2.53 |
| 6E8h | PCIE_PLLCFG4 | This is connected to "CF_APLL_4" on SERDES ANA Initial Settings required prior to power up sequence: PCIE: 0000 609C -- SATA: 0000 0000 (both modes) | Section 3.2.54 |
| 6ECh | PCIE_PLLSTATUS | Initial Settings required prior to power up sequence: | Section 3.2.55 |
| 6F0h | PCIE_RXSTATUS | PCIE SerDes Receive Status register | Section 3.2.56 |
| 6F4h | PCIE_TXSTATUS | PCIE SerDes Transmit Status register | Section 3.2.57 |
| 720h | SATA_PLLCFG0 | SATA PLL Configuration 0. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings | Section 21.5.1 |
| 724h | SATA_PLLCFG1 | SATA PLL Configuration 1. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings. | Section 21.5.2 |
| 728h | SATA_PLLCFG2 | SATA PLL Configuration 2. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings. | Section 21.5.3 |
| 72Ch | SATA_PLLCFG3 | SATA_PLLCFG3 Description. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings. | Section 21.5.4 |
| 730h | SATA_PLLCFG4 | 5PCIe PLL Configuration 4. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings. | Section 21.5.5 |
| 734h | SATA_PLLSTATUS | SATA PLL Status | Section 21.5.6 |
| 738h | SATA_RXSTATUS | SATA RX Status. | Section 21.5.7 |
| 73Ch | SATA_TXSTATUS | SATA TX Status. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings. | Section 21.5.8 |
| 740h | SATA_TESTCFG | SATA TEST Config. | Section 21.5.9 |
| 770h | VDD_MPU_OPP_050 | Ntarget value for OPP50. | Section 3.2.65 |
| 774h | VDD_MPU_OPP_100 | Ntarget value for OPP100. | Section 3.2.66 |
| 778h | VDD_MPU_OPP_120 | Ntarget value for OPP120. | Section 3.2.67 |
| 77Ch | VDD_MPU_OPP_166 | Ntarget value for OPP166. | Section 3.2.68 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-3. CONTROL_MODULE REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 788h | VDD_C674x_DSP_OPP_050 | Ntarget value for OPP50. | Section 3.2.69 |
| 78Ch | VDD_C674x_DSP_OPP_100 | Ntarget value for OPP100. | Section 3.2.70 |
| 790h | VDD_C674x_DSP_OPP_120 | Ntarget value for OPP120. | Section 3.2.71 |
| 794h | VDD_C674x_DSP_OPP_166 | Ntarget value for OPP166. | Section 3.2.72 |
| 7A0h | VDD_HDVICP_OPP_050 | Ntarget value for OPP50. | Section 3.2.73 |
| 7A4h | VDD_HDVICP_OPP_100 | Ntarget value for OPP100. | Section 3.2.74 |
| 7A8h | VDD_HDVICP_OPP_120 | Ntarget value for OPP120. | Section 3.2.75 |
| 7ACh | VDD_HDVICP_OPP_166 | Ntarget value for OPP166. | Section 3.2.76 |
| 7B8h | VDD_CORE_OPP_050 | Ntarget value for OPP50. | Section 3.2.77 |
| 7BCh | VDD_CORE_OPP_100 | Ntarget value for OPP100. | Section 3.2.78 |
| 7C0h | VDD_CORE_OPP_120 | Ntarget value for OPP120. | Section 3.2.79 |
| 7C4h | VDD_CORE_OPP_166 | Ntarget value for OPP166. | Section 3.2.80 |
| 7D0h | BB_SCALE | Back Bias and Dynamic Core Voltage Scaling Register | Section 3.2.81 |
| 7F4h | USB_VID_PID | From TI Efuse, use only for TI defined USB VID/PID option. | Section 3.2.82 |
| 7F8h | PCIE_VID_PID | From TI Efuse, use only for TI defined USB VID/PID option. | Section 3.2.83 |
| 800h | PINCNTL[270:1] | Pin Control | Section 3.2.84 |
| E18h | MLBP_SIG_IO_CTRL | MLBP SIG IO control | Section 3.2.85 |
| E1Ch | MLBP_DAT_IO_CTRL | MLBP DAT IO control | Section 3.2.86 |
| E24h | SERDES_REFCLK_CTL | SerDes Refclk powerdown Control | Section 3.2.87 |
| F00h | DSP_INTMUX_15_18 | DSP INTMUX 15 thru 18 | Section 3.2.88 |
| F04h | DSP_INTMUX_19_22 | DSP INTMUX 19 thru 22 | Section 3.2.89 |
| F08h | DSP_INTMUX_23_26 | DSP INTMUX 23 thru 26 | Section 3.2.90 |
| F0Ch | DSP_INTMUX_27_30 | DSP INTMUX 27 thru 30 | Section 3.2.91 |
| F10h | DSP_INTMUX_31_34 | DSP INTMUX 31 thru 34 | Section 3.2.92 |
| F14h | DSP_INTMUX_35_38 | DSP INTMUX 35 thru 38 | Section 3.2.93 |
| F18h | DSP_INTMUX_39_42 | DSP INTMUX 39 thru 42 | Section 3.2.94 |
| F1Ch | DSP_INTMUX_43_46 | DSP INTMUX 43 thru 46 | Section 3.2.95 |
| F20h | DSP_INTMUX_47_50 | DSP INTMUX 47 thru 50 | Section 3.2.96 |
| F24h | DSP_INTMUX_51_54 | DSP INTMUX 51 thru 54 | Section 3.2.97 |
| F28h | DSP_INTMUX_55_58 | DSP INTMUX 55 thru 58 | Section 3.2.98 |
| F2Ch | DSP_INTMUX_59_62 | DSP INTMUX 59 thru 62 | Section 3.2.99 |
| F30h | DSP_INTMUX_63_66 | DSP INTMUX 63 thru 66 | Section 3.2.100 |
| F34h | DSP_INTMUX_67_70 | DSP INTMUX 67 thru 70 | Section 3.2.101 |
| F38h | DSP_INTMUX_71_74 | DSP INTMUX 71 thru 74 | Section 3.2.102 |
| F3Ch | DSP_INTMUX_75_78 | DSP INTMUX 75 thru 78 | Section 3.2.103 |
| F40h | DSP_INTMUX_79_82 | DSP INTMUX 79 thru 82 | Section 3.2.104 |
| F44h | DSP_INTMUX_83_86 | DSP INTMUX 83 thru 86 | Section 3.2.105 |
| F48h | DSP_INTMUX_87_90 | DSP INTMUX 87 thru 90 | Section 3.2.106 |
| F4Ch | DSP_INTMUX_91_94 | DSP INTMUX 91 thru 94 | Section 3.2.107 |
| F50h | DSP_INTMUX_95 | DSP INTMUX 95 | Section 3.2.108 |
| F54h | Media_Controller_INTMUX_0_3 | Media Controller INTMUX 0 thru 3 | Section 3.2.109 |
| F58h | Media_Controller_INTMUX_4_7 | Media Controller INTMUX 4 thru 7 | Section 3.2.110 |
| F5Ch | Media_Controller_INTMUX_8_11 | Media Controller INTMUX 8 thru 11 | Section 3.2.111 |
| F60h | Media_Controller_INTMUX_12_15 | Media Controller INTMUX 12 thru 15 | Section 3.2.112 |
| F64h | Media_Controller_INTMUX_16_19 | Media Controller INTMUX 16 thru 19 | Section 3.2.113 |
| F68h | Media_Controller_INTMUX_20_23 | Media Controller INTMUX 20 thru 23 | Section 3.2.114 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-3. CONTROL_MODULE REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| F6Ch | Media_Controller_INTMUX_24_27 | Media Controller INTMUX 24 thru 27 | Section 3.2.115 |
| F70h | Media_Controller_INTMUX_28_31 | Media Controller INTMUX 28 thru 31 | Section 3.2.116 |
| F74h | Media_Controller_INTMUX_32_35 | Media Controller INTMUX 32 thru 35 | Section 3.2.117 |
| F78h | Media_Controller_INTMUX_36_39 | Media Controller INTMUX 36 thru 39 | Section 3.2.118 |
| F7Ch | Media_Controller_INTMUX_40_43 | Media Controller INTMUX 40 thru 43 | Section 3.2.119 |
| F80h | Media_Controller_INTMUX_44_47 | Media Controller INTMUX 44 thru 47 | Section 3.2.120 |
| F84h | Media_Controller_INTMUX_48_51 | Media Controller INTMUX 48 thru 51 | Section 3.2.121 |
| F88h | Media_Controller_INTMUX_52_55 | Media Controller INTMUX 52 thru 55 | Section 3.2.122 |
| F8Ch | Media_Controller_INTMUX_56 | Media Controller INTMUX 56 | Section 3.2.123 |
| F90h | EDMA3CC_EVTMUX_0_3 | EDMA3CC EVTMUX 0 thru 3 | Section 3.2.124 |
| F94h | EDMA3CC_EVTMUX_4_7 | EDMA3CC EVTMUX 4 thru 7 | Section 3.2.125 |
| F98h | EDMA3CC_EVTMUX_8_11 | EDMA3CC EVTMUX 8 thru 11 | Section 3.2.126 |
| F9Ch | EDMA3CC_EVTMUX_12_15 | EDMA3CC EVTMUX 12 thru 15 | Section 3.2.127 |
| FA0h | EDMA3CC_EVTMUX_16_19 | EDMA3CC EVTMUX 16 thru 19 | Section 3.2.128 |
| FA4h | EDMA3CC_EVTMUX_20_23 | EDMA3CC EVTMUX 20 thru 23 | Section 3.2.129 |
| FA8h | EDMA3CC_EVTMUX_24_27 | EDMA3CC EVTMUX 24 thru 27 | Section 3.2.130 |
| FACh | EDMA3CC_EVTMUX_28_31 | EDMA3CC EVTMUX 28 thru 31 | Section 3.2.131 |
| FB0h | EDMA3CC_EVTMUX_32_35 | EDMA3CC EVTMUX 32 thru 35 | Section 3.2.132 |
| FB4h | EDMA3CC_EVTMUX_36_39 | EDMA3CC EVTMUX 36 thru 39 | Section 3.2.133 |
| FB8h | EDMA3CC_EVTMUX_40_43 | EDMA3CC EVTMUX 40 thru 43 | Section 3.2.134 |
| FBCh | EDMA3CC_EVTMUX_44_47 | EDMA3CC EVTMUX 44 thru 47 | Section 3.2.135 |
| FC0h | EDMA3CC_EVTMUX_48_51 | EDMA3CC EVTMUX 48 thru 51 | Section 3.2.136 |
| FC4h | EDMA3CC_EVTMUX_52_55 | EDMA3CC EVTMUX 52 thru 55 | Section 3.2.137 |
| FC8h | EDMA3CC_EVTMUX_56_59 | EDMA3CC EVTMUX 56 thru 59 | Section 3.2.138 |
| FCCh | EDMA3CC_EVTMUX_60_63 | EDMA3CC EVTMUX 60 thru 63 | Section 3.2.139 |
| FD4h | GPIO_MUX | GPIO mux select | Section 3.2.140 |
| 1000h | RESET_ISO | RESET ISO | Section 3.2.141 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.1 CONTROL_REVISION Register (offset = 0h) [reset = 40000000h]

CONTROL_REVISION is shown in Figure 3-2 and described in Table 3-4.

The CONTROL_REVISION register is a read only register containing the revision number of the Control module. A write to this register has no effect.

**Figure 3-2. CONTROL_REVISION Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | Reserved | | IP_REV_FUNC | | | |
| R-1h | | R-0h | | R-0h | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| IP_REV_FUNC | | | | | | | |
| R-0h | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| IP_REV_RTL | | | | | IP_REV_MAJOR | | |
| R-0h | | | | | R-0h | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| IP_REV_CUSTOM | | IP_REV_MINOR | | | | | |
| R-0h | | R-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-4. CONTROL_REVISION Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 1h | 0x0 : Legacy (ASP or WTBU) scheme<br>0x1 : Highlander 0.8 scheme |
| 29-28 | Reserved | R | 0h | |
| 27-16 | IP_REV_FUNC | R | 0h | Function indicates a software compatible module family. If there is no level of software compatibility a new Func number (and hence REVISION) should be assigned. |
| 15-11 | IP_REV_RTL | R | 0h | RTL Version (R), maintained by IP design owner. |
| 10-8 | IP_REV_MAJOR | R | 0h | Major Revision (X), maintained by IP specification owner. |
| 7-6 | IP_REV_CUSTOM | R | 0h | Indicates a special version for a particular device. Consequence of use may avoid use of standard Chip Support Library (CSL) / Drivers |
| 5-0 | IP_REV_MINOR | R | 0h | Minor Revision (Y), maintained by IP specification owner. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.2 CONTROL_HWINFO Register (offset = 4h) [reset = 0h]

CONTROL_HWINFO is shown in Figure 3-3 and described in Table 3-5.

The CONTROL_HWINFO register contains information about the module hardware configuration (typically HDL generics, if any).

**Figure 3-3. CONTROL_HWINFO Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IP_HWINFO | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-5. CONTROL_HWINFO Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | IP_HWINFO | R | 0h | IP Module dependent |

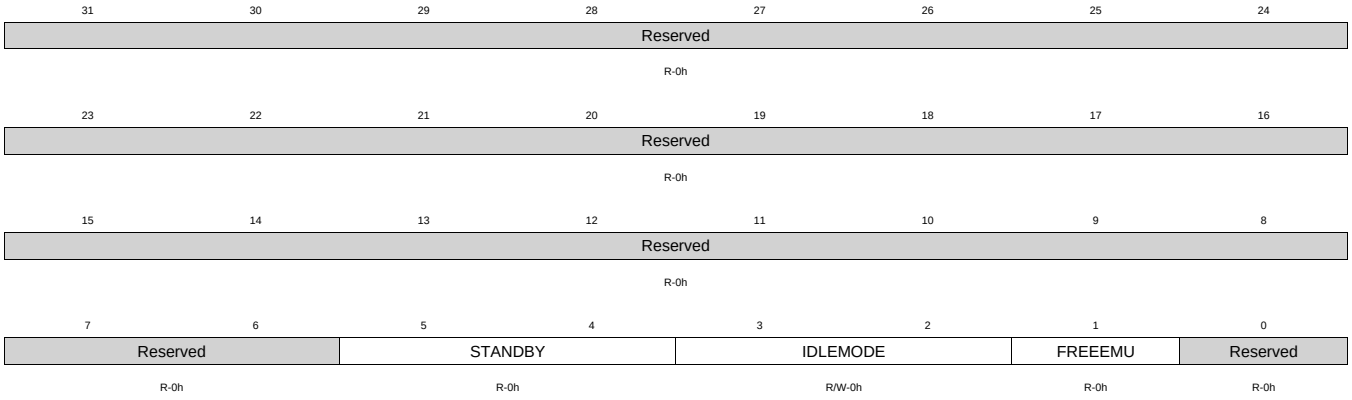

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.3 CONTROL_SYSCONFIG Register (offset = 10h) [reset = 0h]

CONTROL_SYSCONFIG is shown in Figure 3-4 and described in Table 3-6.

The CONTROL_SYSCONFIG register implements only a reduced set of the register defined in the register implements various standby and emulation control bits.

**Figure 3-4. CONTROL_SYSCONFIG Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-6. CONTROL_SYSCONFIG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-4 | STANDBY | R | 0h | Configure local initiator state manaC674x_DSPent.Reserved in Control Module since it has no local initiator.<br>0x0 : Force Standby<br>0x1 : No Standby Mode<br>0x2 : Smart Standby<br>0x3 : Smart Standby wakeup capable |
| 3-2 | IDLEMODE | R/W | 0h | Configure local target state manaC674x_DSPent<br>0x0 : Force Idle<br>0x1 : No Idle<br>0x2 : Smart Idle<br>0x3 : Smart Idle wakeup capable |
| 1 | FREEEMU | R | 0h | Sensitivity to Emulation suspend input.<br>0x0 : Module is sensitive to EMU suspend<br>0x1 : Module not sensitive to EMU suspend |
| 0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.4 CONTROL_STATUS Register (offset = 40h) [reset = 0h]

CONTROL_STATUS is shown in Figure 3-5 and described in Table 3-7.

The CONTROL_STATUS register reflects the system boot and the device type configuration values as sampled when the power-on reset (PORz) or warm reset (RESETz) signal goes high.

**Figure 3-5. CONTROL_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | BOOT_11 | BOOT_10 | BOOT_9_5 |
| R-0h | | | | | R-0h | R-0h | R-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| BOOT_9_5 | | | | ADMUX | | WAITEN | BW |
| R-0h | | | | R-0h | | R-0h | R-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | DEVTYPE | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | SYSBOOT | | | | |
| R-0h | | | R-0h | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-7. CONTROL_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26 | BOOT_11 | R | 0h | RSTOUT latch config.The default value comes from BOOT[11] pins<br>0x0 : RSTOUT only outputs OR of the 3 WD Timer reset output<br>0x1 : RSTOUT outputs combination of 3WD timer reset output + PRCM output |
| 25 | BOOT_10 | R | 0h | Read customer programmable IDs from efuse for PCIe/USB and NAND ROM.The default value comes from BOOT[10] pins |
| 24-20 | BOOT_9_5 | R | 0h | These registers are for future ROM code usage. The details will be published with the ROM code spec. From BOOT[9:5] |
| 19-18 | ADMUX | R | 0h | GPMC CS0 Default Address Muxing.The default value comes from BOOT[13:14] pins<br>0x0 : No Addr/Data Muxing<br>0x1 : Addr/Data Muxing<br>0x2 : Addr/Addr/Data Muxing<br>0x3 : Reserved |
| 17 | WAITEN | R | 0h | GPMC CS0 Default Wait Enable.The default value comes from BOOT[15] pins<br>0x0 : Ignore WAIT input<br>0x1 : Use WAIT input |
| 16 | BW | R | 0h | GPMC CS0 Default Bus Width.The default value comes from BOOT[12] pins<br>0x0 : 8-bit data bus<br>0x1 : 16-bit data bus |
| 15-11 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-7. CONTROL_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10-8 | DEVTYPE | R | 0h | eFuse Device Type Decoded from PRODUCTION_ID eFuse<br>0x0 : Test<br>0x1 : EMUL<br>0x2 : HSL<br>0x3 : GP<br>0x4 : Bad<br>0x5 : HS<br>0x6 : EMU<br>0x7 : Bad |
| 7-5 | Reserved | R | 0h | |
| 4-0 | SYSBOOT | R | 0h | System Boot Type.The default value comes from BOOT[4:0] pins |

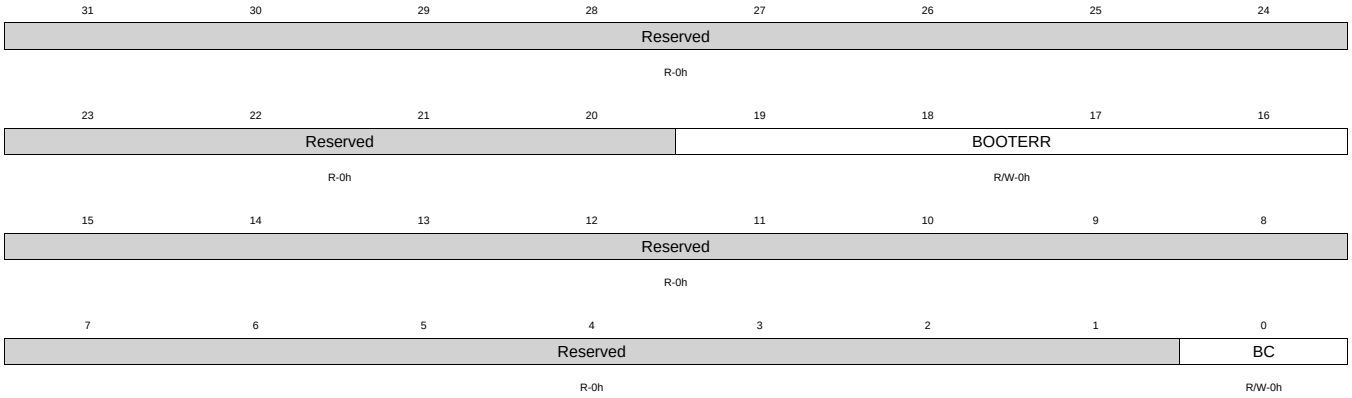

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.5 BOOTSTAT Register (offset = 44h) [reset = 0h]

BOOTSTAT is shown in Figure 3-6 and described in Table 3-8.

The Boot Status (BOOTSTAT) register indicates the status of the device boot process

**Figure 3-6. BOOTSTAT Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-8. BOOTSTAT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-16 | BOOTERR | R/W | 0h | Boot Error The exact meaning of the various error codes will be determined by the bootloader software.<br>0x0: No Boot Error<br>0x1: Others. Bootloader detected boot error |
| 15-1 | Reserved | R | 0h | |
| 0 | BC | R/W | 0h | Boot Complete.This bit may be optionally set by a host boot device to indicate that it has finished loading code. The Host ARM can poll this bit to determine whether to continue the boot process<br>0x0 : Host hasn't complete boot seq.<br>0x1 : Host boot sequence is complete |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.6 DSPBOOTADDR Register (offset = 48h) [reset = 800000h]

DSPBOOTADDR is shown in Figure 3-7 and described in Table 3-9.

The DSP boot address defaults to 0x00800000 (C674x_DSP L2 space). The Host ARM must place DSP boot code into C674x_DSP L2 prior to releasing C674x_DSP reset or change the DSPBOOTADDR to point to the code location. The DSPBOOTADDR register is reset to its default value by a hard reset only and retains its current value over a soft reset.

**Figure 3-7. DSPBOOTADDR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BOOTADDR | | | | | | | |
| R/W-2000h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| BOOTADDR | | | | | | | |
| R/W-2000h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| BOOTADDR | | | | | | Reserved | |
| R/W-2000h | | | | | | R-0h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | RSTDONE |
| R-0h | | | | | | | R-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-9. DSPBOOTADDR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | BOOTADDR | R/W | 2000h | DSP Boot Address (upper 22 bits) |
| 9-1 | Reserved | R | 0h | |
| 0 | RSTDONE | R | 0h | DSP Reset Done (reset_out from C674x_DSP) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.7 MMR_LOCK0 Register (offset = 60h) [reset = 1A1C8144h]

MMR_LOCK0 is shown in Figure 3-8 and described in Table 3-10.

The MMR_LOCKx Register is used allow/block writes to the registers in address region Start_Addess_x - End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x - End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of MMR_LOCKx registers.

**Figure 3-8. MMR_LOCK0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MMR_LOCK0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-1A1C8144h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-10. MMR_LOCK0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | MMR_LOCK0 | R/W | 1A1C8144h | Lock/Unlock register for region 0x0400 - 0x05FF<br>438075716 = LOCKED : All writes to 0x0400 - 0x05FF is blocked.<br>804367403 = UNLOCKED : All writes to 0x0400 - 0x05FF is allowed. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.8 MMR_LOCK1 Register (offset = 64h) [reset = FDF45530h]

MMR_LOCK1 is shown in Figure 3-9 and described in Table 3-11.

The MMR_LOCKx Register is used allow/block writes to the registers in address region Start_Addess_x - End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x - End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of MMR_LOCKx registers.

**Figure 3-9. MMR_LOCK1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MMR_LOCK1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FDF45530h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-11. MMR_LOCK1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | MMR_LOCK1 | R/W | FDF45530h | Lock/Unlock register for region 0x0600 - 0x07FF<br>4149738944 = UNLOCKED : All writes to 0x0600 - 0x07FF is allowed.<br>4260648240 = LOCKED : All writes to 0x0600 - 0x07FF is blocked. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.9 MMR_LOCK2 Register (offset = 68h) [reset = 1AE6E320h]

MMR_LOCK2 is shown in Figure 3-10 and described in Table 3-12.

The MMR_LOCKx Register is used allow/block writes to the registers in address region Start_Addess_x - End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x - End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of MMR_LOCKx registers.

**Figure 3-10. MMR_LOCK2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MMR_LOCK2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-1AE6E320h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-12. MMR_LOCK2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | MMR_LOCK2 | R/W | 1AE6E320h | Lock/Unlock register for region 0x0800 - 0x0FFF<br>3803986541 = UNLOCKED : All writes to 0x0800 - 0x0FFF is allowed.<br>451339040 = LOCKED : All writes to 0x0800 - 0x0FFF is blocked. |

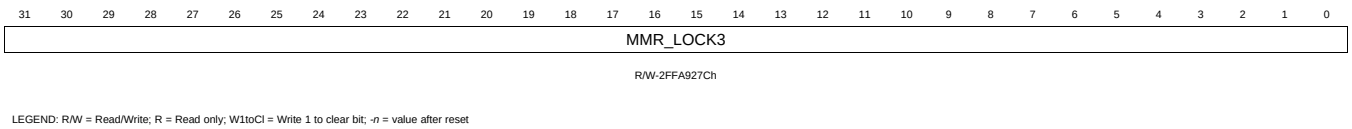

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.10 MMR_LOCK3 Register (offset = 6Ch) [reset = 2FFA927Ch]

MMR_LOCK3 is shown in Figure 3-11 and described in Table 3-13.

The MMR_LOCKx Register is used allow/block writes to the registers in address region Start_Addess_x - End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x - End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of MMR_LOCKx registers.

**Figure 3-11. MMR_LOCK3 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-13. MMR_LOCK3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | MMR_LOCK3 | R/W | 2FFA927Ch | Lock/Unlock register for region 0x1000 - 0x12FF<br>515838749 = UNLOCKED : All writes to 0x1000 - 0x12FF is allowed.<br>804950652 = LOCKED : All writes to 0x1000 - 0x12FF is blocked. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.11 MMR_LOCK4 Register (offset = 70h) [reset = 143F832Ch]

MMR_LOCK4 is shown in Figure 3-12 and described in Table 3-14.

The MMR_LOCKx Register is used allow/block writes to the registers in address region Start_Addess_x - End_Address_x. This register has 2 states LOCKED and UNLOCKED When in LOCKED state all writes to Start_Addess_x -End_Address_x is blocked. When in UNLOCKED state all writes to Start_Addess_x - End_Address_x is allowed. All reads to Start_Addess_x -End_Address_x should not be blocked by the state of MMR_LOCKx registers.

**Figure 3-12. MMR_LOCK4 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MMR_LOCK4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-143F832Ch | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-14. MMR_LOCK4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | MMR_LOCK4 | R/W | 143F832Ch | Lock/Unlock register for region 0x1300 - 0x17FF<br>1865817605 = UNLOCKED : All writes to 0x1300 - 0x17FF is allowed.<br>339706668 = LOCKED : All writes to 0x1300 - 0x17FF is blocked. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.12 CONTROL_SEC_CTRL Register (offset = 100h) [reset = 1Bh]

CONTROL_SEC_CTRL is shown in Figure 3-13 and described in Table 3-15.

Security Control Register

**Figure 3-13. CONTROL_SEC_CTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | Reserved | | | | | |
| R/W-0h | R-0h | R-0h | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | Reserved | WD0OpDisable | Reserved | Reserved | Reserved |
| R-0h | | | R/W-1h | R/W-1h | R-0h | R-1h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-15. CONTROL_SEC_CTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Reserved | R/W | 0h | |
| 30 | Reserved | R | 0h | |
| 29-5 | Reserved | R | 0h | |
| 4 | Reserved | R/W | 1h | |
| 3 | WD0OpDisable | R/W | 1h | The general purpose watchdog timer (WDT 0) is not running by default after reset. This bit must be written to 0 to make the WDT 0 watchdog start running. Once a 0 is written to this bit, it cannot be changed again. Reads will always return 0 and the watchdog timer will continue running. MPU watchdog (WDT 0) operation enable control:<br>0 = Enabled : Watchdog timer is running.<br>1 = Disabled : Watchdog timer is frozen. |
| 2 | Reserved | R | 0h | |
| 1 | Reserved | R | 1h | |
| 0 | Reserved | R | 1h | |

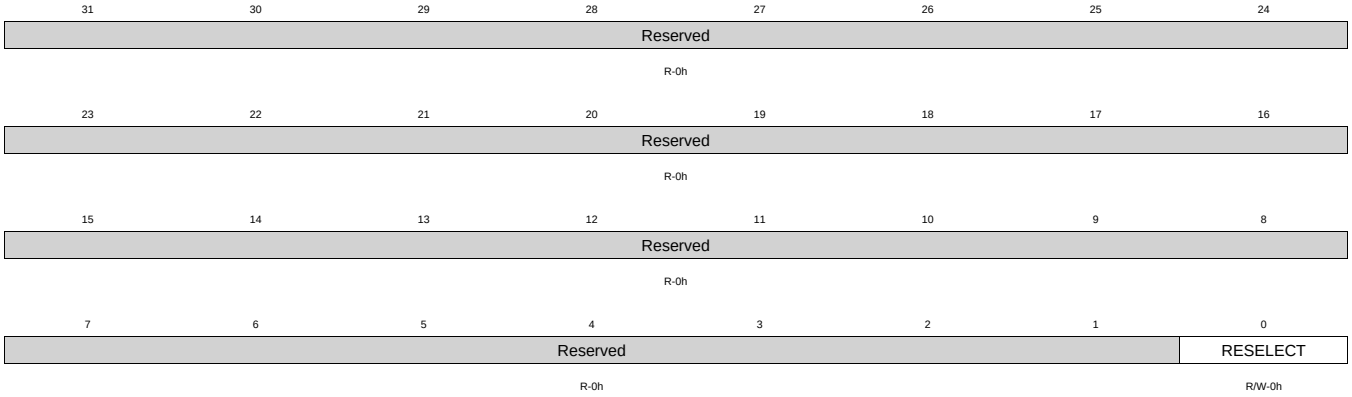

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.13 DEVOSC Register (offset = 468h) [reset = 0h]

DEVOSC is shown in Figure 3-14 and described in Table 3-16.

Oscillator 0 Control. Note for deepsleep: - 1st program bit16 (DSPOLARITY) of register DEEPSLEEP_CTRL to the desired polarity - To enter deepsleep mode, set bit 17 (DSENABLE) of DEEPSLEEP_CTRL to 1. - To exit, toggle the external pin DEEPSLEEPZ.

**Figure 3-14. DEVOSC Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-16. DEVOSC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | RESELECT | R/W | 0h | Internal feedback resistor usage |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.14 AUXOSC Register (offset = 46Ch) [reset = 2h]

AUXOSC is shown in Figure 3-15 and described in Table 3-17.

Oscillator 1 Control

**Figure 3-15. AUXOSC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | GZ | SW2 | SW1 | RESELECT |
| R-0h | | | | R/W-0h | R/W-0h | R/W-1h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-17. AUXOSC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | Reserved | R | 0h | |
| 3 | GZ | R/W | 0h | Setting this to 1 will disable the oscillator. Note: - To enter into deepsleep, set this bit to 1 - To exit deepsleep, set this bit 0 |
| 2 | SW2 | R/W | 0h | 0 (select 15-35 MHz range) |
| 1 | SW1 | R/W | 1h | 1 (select 15-35 MHz range) |
| 0 | RESELECT | R/W | 0h | Internal feedback resistor usage |

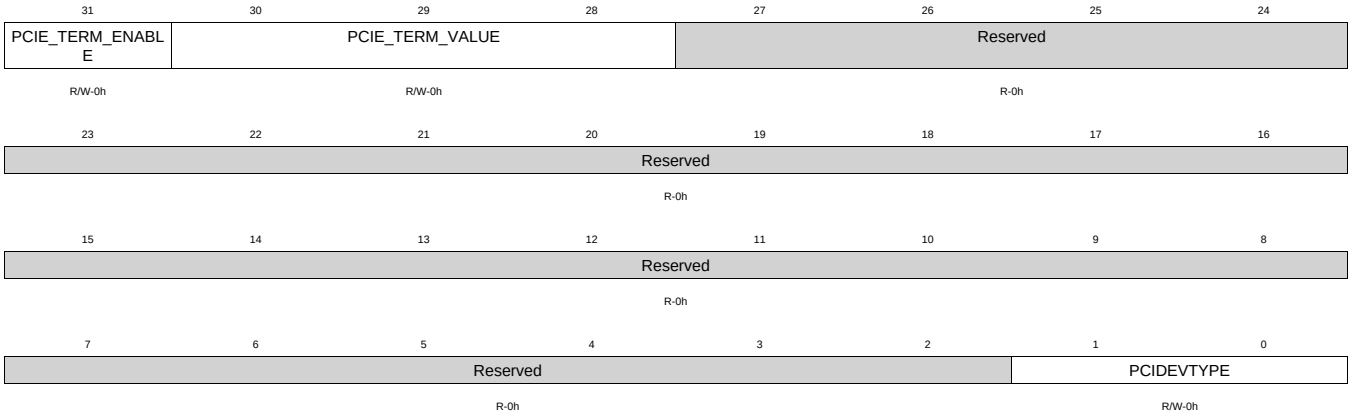

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.15 PCIE_CFG Register (offset = 480h) [reset = 0h]

PCIE_CFG is shown in Figure 3-16 and described in Table 3-18.

5.6.20. PCIE Configuration Register

**Figure 3-16. PCIE_CFG Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-18. PCIE_CFG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PCIE_TERM_ENABLE | R/W | 0h | PCIE Termination enable (override for SERDES termination control, see PCIESS/Serdes documentation). |
| 30-28 | PCIE_TERM_VALUE | R/W | 0h | PCIE Termination value (override for SERDES termination control, see PCIESS/Serdes documentation). |
| 27-2 | Reserved | R | 0h | |
| 1-0 | PCIDEVTYPE | R/W | 0h | Directly controls the DEVTYPE ports on the PCIESS. Used to select PCI device type. Should be valid before releasing reset to PCIESS. Full definition in PCIESS IP spec. |

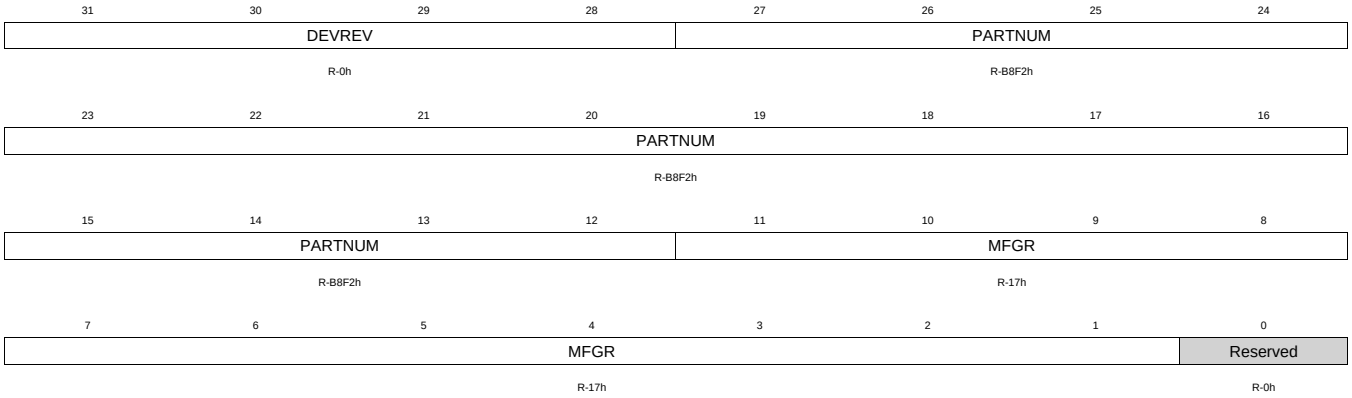

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.16 DEVICE_ID Register (offset = 600h) [reset = B8F202Eh]

DEVICE_ID is shown in Figure 3-17 and described in Table 3-19.

The DEVICE_ID register contains a software readable version of the device JTAG ID. Software can use this register to determine the version of the device on which it is executing.

**Figure 3-17. DEVICE_ID Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-19. DEVICE_ID Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | DEVREV | R | 0h | Device revision.Default value comes from Efuse. |
| 27-12 | PARTNUM | R | B8F2h | Device part number (unique JTAG ID) |
| 11-1 | MFGR | R | 17h | Manufacturer s JTAG ID |
| 0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.17 DEV_FEATURE Register (offset = 604h) [reset = 0h]

DEV_FEATURE is shown in Figure 3-18 and described in Table 3-20.

The DEV_FEATURE register reflects the eFuse settings enabling or disabling certain device functions and capabilities. Note that DEV_FEATURE bits are all implemented as feature enabled bits. This may require inverting the value of those eFuses which are implemented as disable signals.

**Figure 3-18. DEV_FEATURE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DSP | Reserved | SGX530 | DDR1 | HDVICP0 | FDIF | ISP5 | Reserved |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PCIe | SATA | HDMI | CUST_1500 | VCP | MLB | ATL | Reserved |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | Reserved | Reserved | Reserved | ThreeCC | VIP2 | GMII1_RMII1 | MACROV |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-20. DEV_FEATURE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DSP | R | 0h | C674x DSP (C674x_DSP) Enabled |
| 30 | Reserved | R | 0h | |
| 29 | SGX530 | R | 0h | 3D-Graphics Module (SGX530) Enabled |
| 28 | DDR1 | R | 0h | 2nd DDR Interface (DDR1) Enabled |
| 27 | HDVICP0 | R | 0h | HD-VICP0 Co-Processing Engine (HDVICP) Enabled |
| 26 | FDIF | R | 0h | Face Detect Enable |
| 25 | ISP5 | R | 0h | ISP5 Pin Enable (called cam_disable in Ch3 and ISP5 Pin Disable in efuse xls) |
| 24 | Reserved | R | 0h | |
| 23 | PCIe | R | 0h | PCIe Enabled |
| 22 | SATA | R | 0h | SATA Enabled |
| 21 | HDMI | R | 0h | HDMI Enabled |
| 20 | CUST_1500 | R | 0h | Customer Efuse 1500 Enable |
| 19 | VCP | R | 0h | VCP2 Enabled |
| 18 | MLB | R | 0h | MLB Enabled |
| 17 | ATL | R | 0h | ATL Enabled |
| 16 | Reserved | R | 0h | |
| 15 | Reserved | R | 0h | |
| 14 | Reserved | R | 0h | |
| 13 | Reserved | R | 0h | |
| 12 | Reserved | R | 0h | |
| 11 | Reserved | R | 0h | |
| 10 | Reserved | R | 0h | |
| 9 | Reserved | R | 0h | |
| 8 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-20. DEV_FEATURE Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | Reserved | R | 0h | |
| 6 | Reserved | R | 0h | |
| 5 | Reserved | R | 0h | |
| 4 | Reserved | R | 0h | |
| 3 | ThreeCC | R | 0h | 3CC mode Enabled |
| 2 | VIP2 | R | 0h | Enable VIP2 pins (called vin1_disable in ch3 and VIP2 Pin Disable in efuse xls |
| 1 | GMII1_RMII1 | R | 0h | Enable GMII1/RMII1 pins (called mii1_disable in Ch3 and Ethernet Pin disable in efuse xls) |
| 0 | MACROV | R | 0h | Macrovision Enabled |

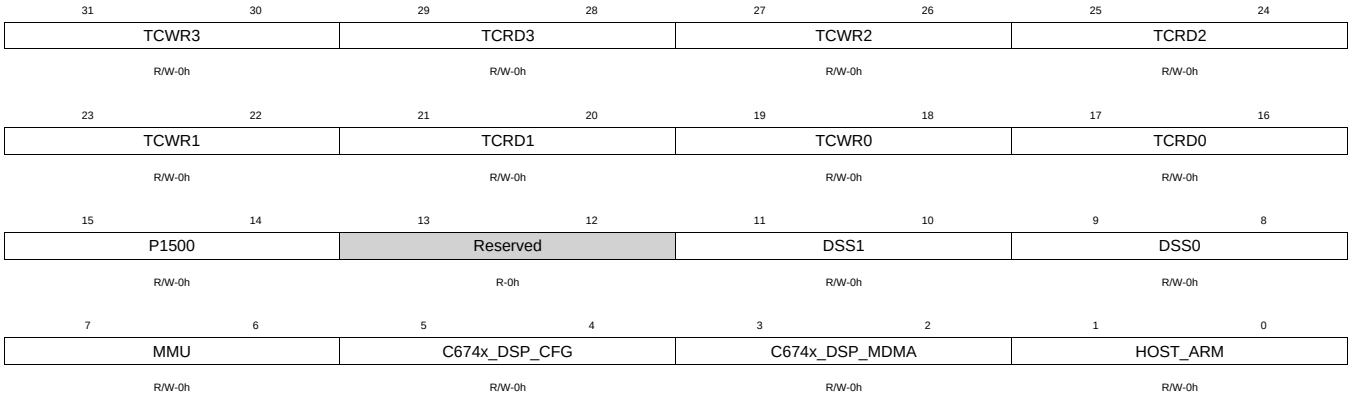

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.18 INIT_PRIORITY_0 Register (offset = 608h) [reset = 0h]

INIT_PRIORITY_0 is shown in Figure 3-19 and described in Table 3-21.

The INIT_PRIORITY_0 register configures the infrastructure pressure input for L3 initiators to provide dynamic priority escalation.

**Figure 3-19. INIT_PRIORITY_0 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-21. INIT_PRIORITY_0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | TCWR3 | R/W | 0h | TPTC 3 Write Port initiator priority |
| 29-28 | TCRD3 | R/W | 0h | TPTC 3 Read Port initiator priority |
| 27-26 | TCWR2 | R/W | 0h | TPTC 2 Write Port initiator priority |
| 25-24 | TCRD2 | R/W | 0h | TPTC 2 Read Port initiator priority |
| 23-22 | TCWR1 | R/W | 0h | TPTC 1 Write Port initiator priority |
| 21-20 | TCRD1 | R/W | 0h | TPTC 1 Read Port initiator priority |
| 19-18 | TCWR0 | R/W | 0h | TPTC 0 Write Port initiator priority |
| 17-16 | TCRD0 | R/W | 0h | TPTC 0 Read Port initiator priority |
| 15-14 | P1500 | R/W | 0h | P1500 Port Initiator priority |
| 13-12 | Reserved | R | 0h | |
| 11-10 | DSS1 | R/W | 0h | Display Subsystem Port 1 initiator priority |
| 9-8 | DSS0 | R/W | 0h | Display Subsystem Port 0 initiator priority |
| 7-6 | MMU | R/W | 0h | System MMU initiator priority |
| 5-4 | C674x_DSP_CFG | R/W | 0h | C674x_DSP CFG port initiator priority |
| 3-2 | C674x_DSP_MDMA | R/W | 0h | C674x_DSP DMA port initiator priority |
| 1-0 | HOST_ARM | R/W | 0h | Host Cortex A8 initiator priority |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.19 INIT_PRIORITY_1 Register (offset = 60Ch) [reset = 0h]

INIT_PRIORITY_1 is shown in Figure 3-20 and described in Table 3-22.

The INIT_PRIORITY_1 register configures the infrastructure pressure input for L3 initiators to provide dynamic priority escalation.

**Figure 3-20. INIT_PRIORITY_1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| FDIF | | PATA | | HDVICP | | DEBUG | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EXP | | GRFX | | Reserved | | PCIE | |
| R/W-0h | | R/W-0h | | R-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DSS_CTLR | | Reserved | | Reserved | | SATA | |
| R/W-0h | | R/W-0h | | R-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USB_QMGR | | USB_DMA | | Reserved | | ThreePGSW | |
| R/W-0h | | R/W-0h | | R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-22. INIT_PRIORITY_1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | FDIF | R/W | 0h | FDIF initiator priority |
| 29-28 | PATA | R/W | 0h | PATA initiator priority |
| 27-26 | HDVICP | R/W | 0h | HDVICP initiator priority |
| 25-24 | DEBUG | R/W | 0h | Debug Subsystem initiator priority |
| 23-22 | EXP | R/W | 0h | Expansion Slot Port initiator priority |
| 21-20 | GRFX | R/W | 0h | 3D Graphics (SGX530) initiator priority |
| 19-18 | Reserved | R | 0h | |
| 17-16 | PCIE | R/W | 0h | PCIe initiator priority |
| 15-14 | DSS_CTLR | R/W | 0h | DSS Controller (Media Controller) initiator priority |
| 13-12 | Reserved | R | 0h | |
| 11-10 | Reserved | R | 0h | |
| 9-8 | SATA | R/W | 0h | SATA initiator priority |
| 7-6 | USB_QMGR | R/W | 0h | USB Queue Manager initiator priority |
| 5-4 | USB_DMA | R/W | 0h | USB DMA port initiator priority |
| 3-2 | Reserved | R | 0h | |
| 1-0 | ThreePGSW | R/W | 0h | 3PGSW initiator priority |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.20 MMU_CFG Register (offset = 610h) [reset = 80h]

MMU_CFG is shown in Figure 3-21 and described in Table 3-23.

The MMU_CFG register is used to optionally route certain initiators through the System MMU. The C674x_DSP MDMA port accesses will always be routed through the System MMU so no configuration bit is instantiated for C674x_DSP. Note that when TPTC0 is routed through the MMU accesses to the C674x_DSP SDMA port will still by-pass the MMU for increased performance.

**Figure 3-21. MMU_CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| MMU_ABORT | Reserved | | | | | | |
| R/W-0h | R-0h | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MMU_EN | Reserved | | | | | TC1MMU | TC0MMU |
| R/W-1h | R-0h | | | | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-23. MMU_CFG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15 | MMU_ABORT | R/W | 0h | MMU abort operation This bit causes the MMU to abort the current operation in case of lockup. |
| 14-8 | Reserved | R | 0h | |
| 7 | MMU_EN | R/W | 1h | MMU Enable Clearing this bit disables MMU table lookup and causes accesses to use the non-translated address. This bit defaults to enabled but an identical bit within an MMU configuration register defaults to disabled and must be set after the page tables are programmed for MMU operation |
| 6-2 | Reserved | R | 0h | |
| 1 | TC1MMU | R/W | 0h | TPTC1 Uses MMU |
| 0 | TC0MMU | R/W | 0h | TPTC0 Uses MMU |

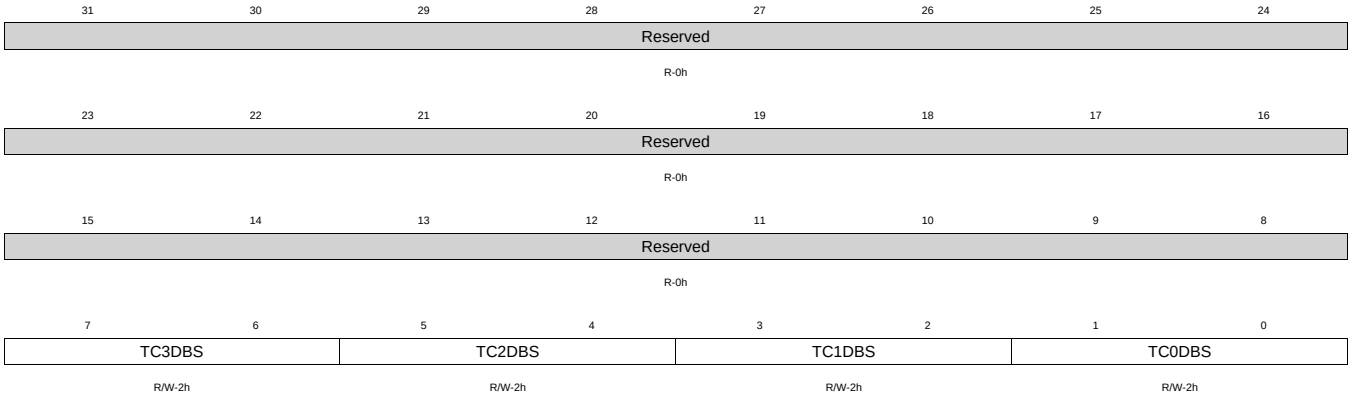

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.21 TPTC_CFG Register (offset = 614h) [reset = AAh]

TPTC_CFG is shown in Figure 3-22 and described in Table 3-24.

The TPTC_CFG register configures the default burst size for TPTC0, 1, 2, and 3.

**Figure 3-22. TPTC_CFG Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-24. TPTC_CFG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | Reserved | R | 0h | |
| 7-6 | TC3DBS | R/W | 2h | TC3 Default Burst Size<br>0x0 : 16 byte<br>0x1 : 32 byte<br>0x2 : 64 byte<br>0x3 : 128 byte |
| 5-4 | TC2DBS | R/W | 2h | TC2 Default Burst Size<br>0x0 : 16 byte<br>0x1 : 32 byte<br>0x2 : 64 byte<br>0x3 : 128 byte |
| 3-2 | TC1DBS | R/W | 2h | TC1 Default Burst Size<br>0x0 : 16 byte<br>0x1 : 32 byte<br>0x2 : 64 byte<br>0x3 : 128 byte |
| 1-0 | TC0DBS | R/W | 2h | TC0 Default Burst Size<br>0x0 : 16 byte<br>0x1 : 32 byte<br>0x2 : 64 byte<br>0x3 : 128 byte |

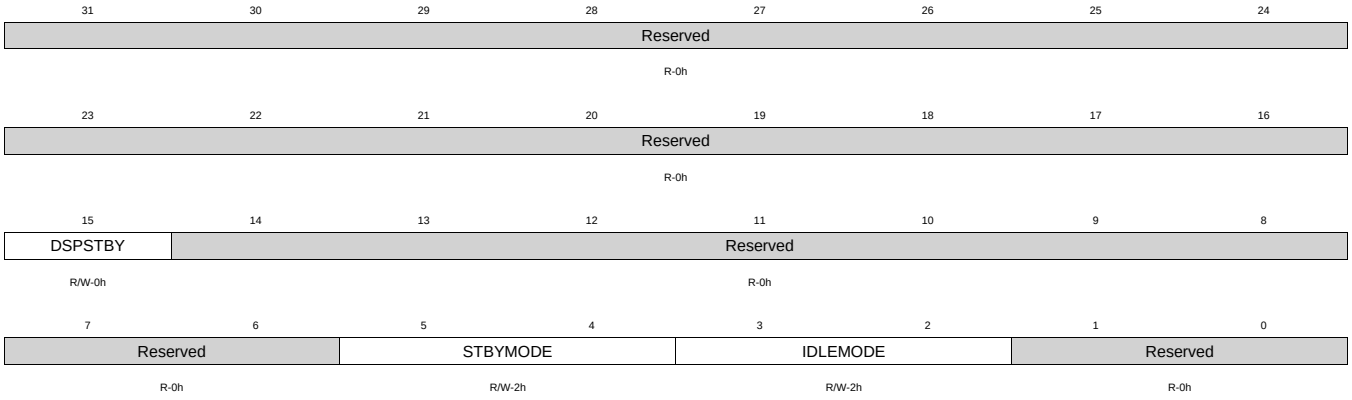

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.22 DSP_IDLE_CFG Register (offset = 61Ch) [reset = 28h]

DSP_IDLE_CFG is shown in Figure 3-23 and described in Table 3-25.

The C674x_DSP_IDLE_CFG register controls the state manaC674x_DSPent of the C674x_DSP initiator and target port interfaces.

**Figure 3-23. DSP_IDLE_CFG Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-25. DSP_IDLE_CFG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15 | DSPSTBY | R/W | 0h | Assert DSP Standby |
| 14-6 | Reserved | R | 0h | |
| 5-4 | STBYMODE | R/W | 2h | Initiator state manaC674x_DSPent mode<br>0x0 : Force-standby mode: Initiator is unconditionally placed in standby state. (for debug only)<br>0x1 : No-standby mode: Initiator is unconditionally removed from standby state. (for debug only)<br>0x2 : Smart-standby mode: Initiator standby status depends on local condition (module functional requirements of initiator). No initiator related wake-up events.<br>0x3 : Smart-standby wakeup-capable mode. : Initiator standby status depends on local condition (module functional requirements of initiator). Initiator related wake-up events may be generated when in standby. |
| 3-2 | IDLEMODE | R/W | 2h | Target state manaC674x_DSPent mode<br>0x0 : Force-idle mode: Target acknowledges idle requests unconditionally (debug only)<br>0x1 : No-idle mode: Target never enters idle state. (for debug only)<br>0x2 : Smart-idle mode: Target acknowledges idle requests after fulfilling IP internal requirements. No IRQ/DMA related wake-up events.<br>0x3 : Smart-idle wakeup-capable mode: Target acknowledges idle requests after fulfilling IP internal requirements. IRQ/DMA related wake-up events may be generated when in standby |
| 1-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.23 USB_CTRL0 Register (offset = 620h) [reset = 3C006007h]

USB_CTRL0 is shown in Figure 3-24 and described in Table 3-26.

USB Control Register 0 controls the PHY, CM and OTG modules for USB0.

**Figure 3-24. USB_CTRL0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SPAREIN7 | SPAREIN6 | SPAREIN5 | SPAREIN4 | SPAREIN3 | SPAREIN2 | SPAREIN1 | SPAREIN0 |
| R/W-0h | R/W-0h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DATAPOLARITY_INV | USB_PHY_SMA2 | USB_PHY_SMA1 | OTGSESSENDEN | OTGVDET_EN | DMGPIO_PD | DPGPIO_PD | DMINPUT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DPINPUT | DMOPBUFCTL | DPOPBUFCTL | GPIOMODE | Reserved | CDET_EXTCTL | DPPULLUP | DMPULLUP |
| R/W-0h | R/W-1h | R/W-1h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CHGVSRC_EN | CHGISINK_EN | SINKONDP | SRCONDM | CHGDET_RSTRT | CHGDET_DIS | OTG_PWRDN | CM_PWRDN |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R/W-1h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-26. USB_CTRL0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SPAREIN7 | R/W | 0h | To PHY SPAREINx ports |
| 30 | SPAREIN6 | R/W | 0h | To PHY SPAREINx ports |
| 29 | SPAREIN5 | R/W | 1h | To PHY SPAREINx ports |
| 28 | SPAREIN4 | R/W | 1h | To PHY SPAREINx ports |
| 27 | SPAREIN3 | R/W | 1h | To PHY SPAREINx ports |
| 26 | SPAREIN2 | R/W | 1h | To PHY SPAREINx ports |
| 25 | SPAREIN1 | R/W | 0h | To PHY SPAREINx ports |
| 24 | SPAREIN0 | R/W | 0h | To PHY SPAREINx ports |
| 23 | DATAPOLARITY_INV | R/W | 0h | Data Polarity Invert:<br>0x0 : DP/DM (normal polarity matching port definition)<br>0x1 : DM/DP (inverted polarity of port definition) |
| 22 | USB_PHY_SMA2 | R/W | 0h | Reserved: Write to 0. |
| 21 | USB_PHY_SMA1 | R/W | 0h | Reserved: Write to 0. |
| 20 | OTGSESSENDEN | R/W | 0h | Session End Detect Enable<br>0x0 : Disable Session End Comparator<br>0x1 : Turns on Session End Comparator |
| 19 | OTGVDET_EN | R/W | 0h | VBUS Detect Enable<br>0x0 : Disable VBUS Detect Enable<br>0x1 : Turns on all comparators except Session End comparator |
| 18 | DMGPIO_PD | R/W | 0h | Pull-down on DM in GPIO Mode<br>0x0 : Enables pull-down<br>0x1 : Disables pull-down |
| 17 | DPGPIO_PD | R/W | 0h | Pull-down on DP in GPIO Mode<br>0x0 : Enables pull-down<br>0x1 : Disables pull-down |
| 16 | DMINPUT | R/W | 0h | DM Input in GPIO Mode |
| 15 | DPINPUT | R/W | 0h | DP Input in GPIO Mode |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-26. USB_CTRL0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | DMOPBUFCTL | R/W | 1h | DM Output Buffer Control<br>0x0 : Enables Output Buffer<br>0x1 : Disables Output Buffer |
| 13 | DPOPBUFCTL | R/W | 1h | DP Output Buffer Control<br>0x0 : Enables Output Buffer<br>0x1 : Disables Output Buffer |
| 12 | GPIOMODE | R/W | 0h | GPIO Mode<br>0x0 : USB Mode<br>0x1 : GPIO Mode |
| 11 | Reserved | R | 0h | |
| 10 | CDET_EXTCTL | R/W | 0h | Bypass the charger detection state machine<br>0x0 : Charger detection on<br>0x1 : Charger detection is bypassed |
| 9 | DPPULLUP | R/W | 0h | Pull-up on DP line<br>0x0 : No effect<br>0x1 : Enable pull-down on DP line |
| 8 | DMPULLUP | R/W | 0h | Pull-up on DM line<br>0x0 : No effect<br>0x1 : Enable pull-up on DM line |
| 7 | CHGVSRC_EN | R/W | 0h | Enable VSRC on DP line (Host Charger case) |
| 6 | CHGISINK_EN | R/W | 0h | Enable ISINK on DM line (Host Charger case) |
| 5 | SINKONDP | R/W | 0h | Sink on DP<br>0x0 : Sink on DM<br>0x1 : Sink on DP |
| 4 | SRCONDM | R/W | 0h | Charger Detect Disable<br>0x0 : Source on DP<br>0x1 : Source on DM |
| 3 | CHGDET_RSTRT | R/W | 0h | Restart Charger Detect |
| 2 | CHGDET_DIS | R/W | 1h | Charger Detect Disable<br>0x0 : Enable<br>0x1 : Disable |
| 1 | OTG_PWRDN | R/W | 1h | Power down the USB OTG PHY<br>0x0 : PHY in normal mode<br>0x1 : PHY Powered down |
| 0 | CM_PWRDN | R/W | 1h | Power down the USB CM PHY<br>0x0 : PHY in normal mode<br>0x1 : PHY Powered down |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.24 USB_STS0 Register (offset = 624h) [reset = 0h]

USB_STS0 is shown in Figure 3-25 and described in Table 3-27.

USB Status Register 0 reflects the status of the USB0.

**Figure 3-25. USB_STS0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | GPIO_DMOUT | GPIO_DPOUT | Reserved | | | |
| R-0h | | R/W-0h | R/W-0h | R-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CHGDETSTS | | | CDET_DMDET | CDET_DPDET | CDET_DATADET | CHGDETECT | CHGDETDONE |
| R/W-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-27. USB_STS0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-14 | Reserved | R | 0h | |
| 13 | GPIO_DMOUT | R/W | 0h | Data on DM in GPIO mode |
| 12 | GPIO_DPOUT | R/W | 0h | Data on DP in GPIO mode |
| 11-8 | Reserved | R | 0h | |
| 7-5 | CHGDETSTS | R/W | 0h | Charge Detection Status<br>0x0 : Wait State (When a D+WPU and D-15K are connected, it enters into this state and will remain in this state unless it enters into other state)<br>0x1 : No Contact<br>0x2 : PS/2<br>0x3 : Unknown error<br>0x4 : Dedicated charger(valid if CE is HIGH)<br>0x5 : HOST charger (valid if CE is HIGH)<br>0x6 : PC<br>0x7 : Interrupt (if any of the pull-up is enabled, charger detect routine gets interrupted and will restart from the beginning if the same is disabled) |
| 4 | CDET_DMDET | R/W | 0h | DM Comparator Output |
| 3 | CDET_DPDET | R/W | 0h | DP Comparator Output |
| 2 | CDET_DATADET | R/W | 0h | Charger Comparator Output |
| 1 | CHGDETECT | R/W | 0h | Charger Detection Status<br>0x0 : Charger was NOT detected<br>0x1 : Charger was detected |
| 0 | CHGDETDONE | R/W | 0h | Charger Detection Protocol Done |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.25 USB_CTRL1 Register (offset = 628h) [reset = 3C006007h]

USB_CTRL1 is shown in Figure 3-26 and described in Table 3-28.

USB Control Register 1 controls the PHY, CM and OTG modules for USB1.

**Figure 3-26. USB_CTRL1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SPAREIN7 | SPAREIN6 | SPAREIN5 | SPAREIN4 | SPAREIN3 | SPAREIN2 | SPAREIN1 | SPAREIN0 |
| R/W-0h | R/W-0h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DATAPOLARITY_INV | USB_PHY_SMA2 | USB_PHY_SMA1 | OTGSESSENDEN | OTGVDET_EN | DMGPIO_PD | DPGPIO_PD | DMINPUT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DPINPUT | DMOPBUFCTL | DPOPBUFCTL | GPIOMODE | Reserved | CDET_EXTCTL | DPPULLUP | DMPULLUP |
| R/W-0h | R/W-1h | R/W-1h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CHGVSRC_EN | CHGISINK_EN | SINKONDP | SRCONDM | CHGDET_RSTRT | CHGDET_DIS | OTG_PWRDN | CM_PWRDN |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R/W-1h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-28. USB_CTRL1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SPAREIN7 | R/W | 0h | To PHY SPAREINx ports |
| 30 | SPAREIN6 | R/W | 0h | To PHY SPAREINx ports |
| 29 | SPAREIN5 | R/W | 1h | To PHY SPAREINx ports |
| 28 | SPAREIN4 | R/W | 1h | To PHY SPAREINx ports |
| 27 | SPAREIN3 | R/W | 1h | To PHY SPAREINx ports |
| 26 | SPAREIN2 | R/W | 1h | To PHY SPAREINx ports |
| 25 | SPAREIN1 | R/W | 0h | To PHY SPAREINx ports |
| 24 | SPAREIN0 | R/W | 0h | To PHY SPAREINx ports |
| 23 | DATAPOLARITY_INV | R/W | 0h | Data Polarity Invert:<br>0x0 : DP/DM (normal polarity matching port definition)<br>0x1 : DM/DP (inverted polarity of port definition) |
| 22 | USB_PHY_SMA2 | R/W | 0h | Reserved: Write to 0. |
| 21 | USB_PHY_SMA1 | R/W | 0h | Reserved: Write to 0. |
| 20 | OTGSESSENDEN | R/W | 0h | Session End Detect Enable<br>0x0 : Disable Session End Comparator<br>0x1 : Turns on Session End Comparator |
| 19 | OTGVDET_EN | R/W | 0h | VBUS Detect Enable<br>0x0 : Disable "VBUS Detect Enable"<br>0x1 : Turns on all comparators except Session End comparator |
| 18 | DMGPIO_PD | R/W | 0h | Pull-down on DM in GPIO Mode<br>0x0 : Enables pull-down<br>0x1 : Disables pull-down |
| 17 | DPGPIO_PD | R/W | 0h | Pull-down on DP in GPIO Mode<br>0x0 : Enables pull-down<br>0x1 : Disables pull-down |
| 16 | DMINPUT | R/W | 0h | DM Input in GPIO Mode |
| 15 | DPINPUT | R/W | 0h | DP Input in GPIO Mode |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-28. USB_CTRL1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | DMOPBUFCTL | R/W | 1h | DM Output Buffer Control<br>0x0 : Enables Output Buffer<br>0x1 : Disables Output Buffer |
| 13 | DPOPBUFCTL | R/W | 1h | GPIO Mode<br>0x0 : Enables Output Buffer<br>0x1 : Disables Output Buffer |
| 12 | GPIOMODE | R/W | 0h | GPIO Mode<br>0x0 : USB Mode<br>0x1 : GPIO Mode |
| 11 | Reserved | R | 0h | |
| 10 | CDET_EXTCTL | R/W | 0h | Bypass the charger detection state machine<br>0x0 : Charger detection on<br>0x1 : Charger detection is bypassed |
| 9 | DPPULLUP | R/W | 0h | Pull-up on DP line<br>0x0 : No effect<br>0x1 : Enable pull-down on DP line |
| 8 | DMPULLUP | R/W | 0h | Pull-up on DM line<br>0x0 : No effect<br>0x1 : Enable pull-up on DM line |
| 7 | CHGVSRC_EN | R/W | 0h | Enable VSRC on DP line (Host Charger case) |
| 6 | CHGISINK_EN | R/W | 0h | Enable ISINK on DM line (Host Charger case) |
| 5 | SINKONDP | R/W | 0h | Sink on DP<br>0x0 : Sink on DM<br>0x1 : Sink on DP |
| 4 | SRCONDM | R/W | 0h | Charger Detect Disable<br>0x0 : Source on DP<br>0x1 : Source on DM |
| 3 | CHGDET_RSTRT | R/W | 0h | Restart Charger Detect |
| 2 | CHGDET_DIS | R/W | 1h | Charger Detect Disable<br>0x0 : Enable<br>0x1 : Disable |
| 1 | OTG_PWRDN | R/W | 1h | Power down the USB OTG PHY<br>0x0 : PHY in normal mode<br>0x1 : PHY Powered down |
| 0 | CM_PWRDN | R/W | 1h | Power down the USB CM PHY<br>0x0 : PHY in normal mode<br>0x1 : PHY Powered down |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.26 USB_STS1 Register (offset = 62Ch) [reset = 0h]

USB_STS1 is shown in Figure 3-27 and described in Table 3-29.

USB Status Register 1 reflects the status of the USB0.

**Figure 3-27. USB_STS1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | GPIO_DMOUT | GPIO_DPOUT | Reserved | | | |
| R-0h | | R/W-0h | R/W-0h | R-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CHGDETSTS | | | CDET_DMDET | CDET_DPDET | CDET_DATADET | CHGDETECT | CHGDETDONE |
| R/W-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-29. USB_STS1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-14 | Reserved | R | 0h | |
| 13 | GPIO_DMOUT | R/W | 0h | Data on DM in GPIO mode |
| 12 | GPIO_DPOUT | R/W | 0h | Data on DP in GPIO mode |
| 11-8 | Reserved | R | 0h | |
| 7-5 | CHGDETSTS | R/W | 0h | Charge Detection Status<br>0x0 : Wait State (When a D+WPU and D-15K are connected, it enters into this state and will remain in this state unless it enters into other state)<br>0x1 : No Contact<br>0x2 : PS/2<br>0x3 : Unknown error<br>0x4 : Dedicated charger(valid if CE is HIGH)<br>0x5 : HOST charger (valid if CE is HIGH)<br>0x6 : PC<br>0x7 : Interrupt (if any of the pull-up is enabled, charger detect routine gets interrupted and will restart from the beginning if the same is disabled) |
| 4 | CDET_DMDET | R/W | 0h | DM Comparator Output |
| 3 | CDET_DPDET | R/W | 0h | DP Comparator Output |
| 2 | CDET_DATADET | R/W | 0h | Charger Comparator Output |
| 1 | CHGDETECT | R/W | 0h | Charger Detection Status<br>0x0 : Charger was NOT detected<br>0x1 : Charger was detected |
| 0 | CHGDETDONE | R/W | 0h | Charger Detection Protocol Done |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.27 MAC_ID0_LO Register (offset = 630h) [reset = 0h]

MAC_ID0_LO is shown in Figure 3-28 and described in Table 3-30.

The MAC_ID0_LO register contains the lower 2 bytes of the 48-bit ID for MAC0.

**Figure 3-28. MAC_ID0_LO Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| MACADDR_7_0 | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MACADDR_15_8 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-30. MAC_ID0_LO Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-8 | MACADDR_7_0 | R/W | 0h | MAC0 Address -Byte 0.Value comes from Efuse |
| 7-0 | MACADDR_15_8 | R/W | 0h | MAC0 Address -Byte 1.Value comes from Efuse |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.28 MAC_ID0_HI Register (offset = 634h) [reset = 0h]

MAC_ID0_HI is shown in Figure 3-29 and described in Table 3-31.

The MAC_ID0_HI register contains the upper 4 bytes of the 48-bit ID for MAC0.

**Figure 3-29. MAC_ID0_HI Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MACADDR_23_16 | | | | | | | |
| R/W-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| MACADDR_31_24 | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| MACADDR_39_32 | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MACADDR_47_40 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-31. MAC_ID0_HI Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | MACADDR_23_16 | R/W | 0h | MAC0 Address -Byte 5. Value comes from Efuse |
| 23-16 | MACADDR_31_24 | R/W | 0h | MAC0 Address -Byte 3. Value comes from Efuse |
| 15-8 | MACADDR_39_32 | R/W | 0h | MAC0 Address -Byte 4. Value comes from Efuse |
| 7-0 | MACADDR_47_40 | R/W | 0h | MAC0 Address -Byte 5. Value comes from Efuse |

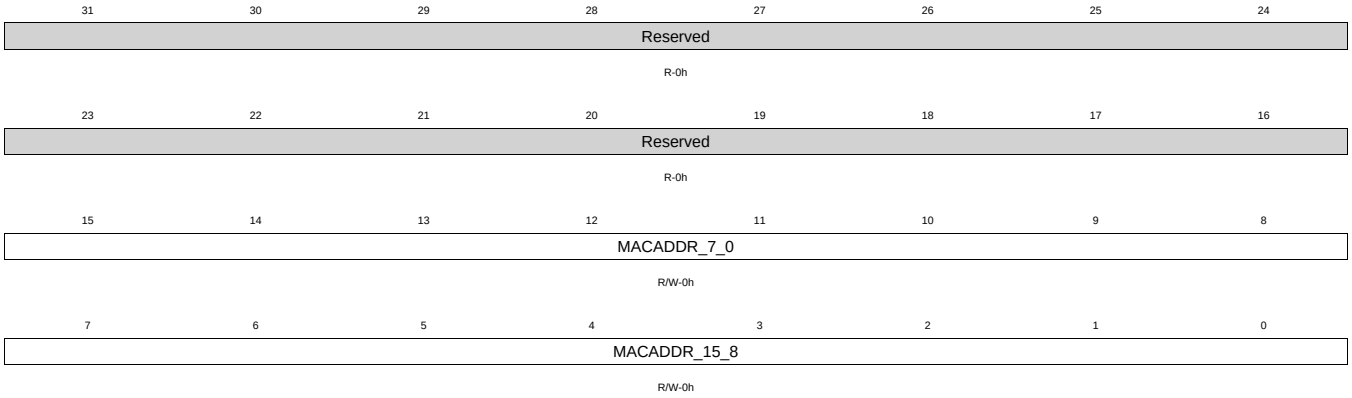

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.29 MAC_ID1_LO Register (offset = 638h) [reset = 0h]

MAC_ID1_LO is shown in Figure 3-30 and described in Table 3-32.

The MAC_ID1_LO register contains the lower 2 bytes of the 48-bit ID for MAC1.

**Figure 3-30. MAC_ID1_LO Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-32. MAC_ID1_LO Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-8 | MACADDR_7_0 | R/W | 0h | MAC1 Address -Byte 0.Value comes from Efuse |
| 7-0 | MACADDR_15_8 | R/W | 0h | MAC1 Address -Byte 1.Value comes from Efuse |

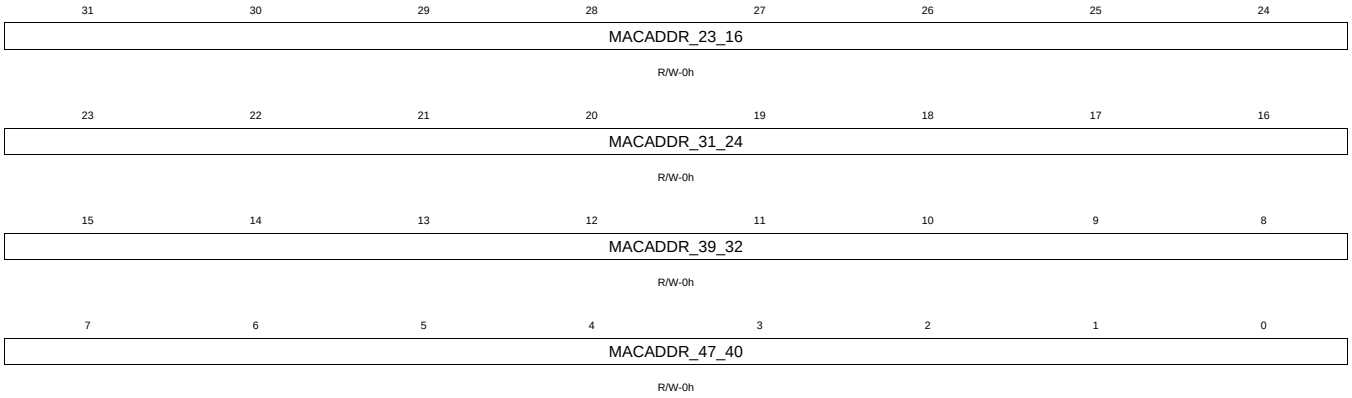

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.30 MAC_ID1_HI Register (offset = 63Ch) [reset = 0h]

MAC_ID1_HI is shown in Figure 3-31 and described in Table 3-33.

The MAC_ID1_HI register contains the upper 4 bytes of the 48-bit ID for MAC1.

**Figure 3-31. MAC_ID1_HI Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-33. MAC_ID1_HI Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | MACADDR_23_16 | R/W | 0h | MAC1 Address -Byte 5. Value comes from Efuse |
| 23-16 | MACADDR_31_24 | R/W | 0h | MAC1 Address -Byte 3. Value comes from Efuse |
| 15-8 | MACADDR_39_32 | R/W | 0h | MAC1 Address -Byte 4. Value comes from Efuse |
| 7-0 | MACADDR_47_40 | R/W | 0h | MAC1 Address -Byte 5. Value comes from Efuse |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.31 SW_REVISION Register (offset = 640h) [reset = 0h]

SW_REVISION is shown in Figure 3-32 and described in Table 3-34.

The SW_REVISION register reads the Paper Spin Variant 32bit efuse. The SW can read this bit and decide what to do with a specific SW package.

**Figure 3-32. SW_REVISION Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REV | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-34. SW_REVISION Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | REV | R | 0h | SW revision field from efuse |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.32 DCAN_RAMINIT Register (offset = 644h) [reset = 0h]

DCAN_RAMINIT is shown in Figure 3-33 and described in Table 3-35.

The DCAN_RAMINIT register is used to start/check status of the autoinitialization process of DCAN RAM.

**Figure 3-33. DCAN_RAMINIT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | DCAN1_RAMINIT_DONE | DCAN0_RAMINIT_DONE |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | DCAN1_RAMINIT_START | DCAN0_RAMINIT_START |
| R-0h | | | | | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-35. DCAN_RAMINIT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | Reserved | R | 0h | |
| 9 | DCAN1_RAMINIT_DONE | R/W | 0h | RAM initialization status<br>0x0 : DCAN1 RAM Initialization NOT complete<br>0x1 : DCAN1 RAM Initialization complete |
| 8 | DCAN0_RAMINIT_DONE | R/W | 0h | RAM initialization status<br>0x0 : DCAN0 RAM Initialization NOT complete. W1toClr<br>0x1 : DCAN0 RAM Initialization complete.W1toClr |
| 7-2 | Reserved | R | 0h | |
| 1 | DCAN1_RAMINIT_START | R/W | 0h | A transition from 0 to 1 will start DCAN1 RAM initialization sequence. |
| 0 | DCAN0_RAMINIT_START | R/W | 0h | A transition from 0 to 1 will start DCAN0 RAM initialization sequence. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.33 AUD_CTRL Register (offset = 64Ch) [reset = 0h]

AUD_CTRL is shown in Figure 3-34 and described in Table 3-36.

The register controls loopback of McBSP data for testing purposes.

**Figure 3-34. AUD_CTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | MCB_LBFSX | MCB_LBCLKX |
| R-0h | | | | | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-36. AUD_CTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0h | |
| 1 | MCB_LBFSX | R/W | 0h | McBSP FSX Loopback enable -Drives McBSP PIALBCTRLRX[1]<br>0x0 : No loopback<br>0x1 : Enable loopback |
| 0 | MCB_LBCLKX | R/W | 0h | McBSP CLKX Loopback enable -Drives McBSP PIALBCTRLRX[0]<br>0x0 : No loopback<br>0x1 : Enable loopback |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.34 GMII_SEL Register (offset = 650h) [reset = 0h]

GMII_SEL is shown in Figure 3-35 and described in Table 3-37.

**Figure 3-35. GMII_SEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | RGMII1_EN | RGMII0_EN |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | RGMII1_ID_MODE | RGMII0_ID_MODE | GMII1_SEL | | GMII0_SEL | |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-37. GMII_SEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | Reserved | R | 0h | |
| 9 | RGMII1_EN | R/W | 0h | RGMII vs GMII pin mux control for **Port 1**. For this bit to take effect, the PINCNTL settings for the device must select the GMII related function. This register is a secondary control that selects GMII vs RGMII.<br>0x0 : GMII / MII mode<br>0x1 : RGMII mode |
| 8 | RGMII0_EN | R/W | 0h | RGMII vs GMII pin mux control for **Port 0**. For this bit to take effect, the PINCNTL settings for the device must select the GMII related function. This register is a secondary control that selects GMII vs RGMII.<br>0x0 : GMII / MII mode<br>0x1 : RGMII mode |
| 7-6 | Reserved | R | 0h | |
| 5 | RGMII1_ID_MODE | R/W | 0h | Port 1 CPRGMII Internal Delay Mode<br>0x0 : Internal delay<br>0x1 : No Internal delay |
| 4 | RGMII0_ID_MODE | R/W | 0h | Port 0 CPRGMII Internal Delay Mode<br>0x0 : Internal delay<br>0x1 : No Internal delay |
| 3-2 | GMII1_SEL | R/W | 0h | Port 1 functionality<br>0x0 : GMII/MII<br>0x1 : RMII<br>0x2 : RGMII |
| 1-0 | GMII0_SEL | R/W | 0h | Port 0 functionality<br>0x0 : GMII/MII<br>0x1 : RMII<br>0x2 : RGMII |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.35 OCMEM_PWRDN Register (offset = 654h) [reset = 0h]

OCMEM_PWRDN is shown in Figure 3-36 and described in Table 3-38.

The OCMEM_PWRDN Register determines the powerdown behavior of the OCMC RAM memories.

**Figure 3-36. OCMEM_PWRDN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MEM_PWRDN_STATUS | MEM_PWRDN |
| R-0h | | | | | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-38. OCMEM_PWRDN Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0h | |
| 1 | MEM_PWRDN_STATUS | R/W | 0h | Power down status from the OCMEM<br>0x0 : memory not powered down<br>0x1 : memory powered down |
| 0 | MEM_PWRDN | R/W | 0h | Power down request to the OCMEM<br>0x0 : No memory power down requested<br>0x1 : memory power down requested |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.36 MEDIA_CONTROLLER_MEM_PWRDN Register (offset = 65Ch) [reset = 0h]

MEDIA_CONTROLLER_MEM_PWRDN is shown in Figure 3-37 and described in Table 3-39.

The MEDIA_CONTROLLER_MEM_PWRDN Register determines the powerdown behavior of the Media Controller memories.

**Figure 3-37. MEDIA_CONTROLLER_MEM_PWRDN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | MEM_PWRDN_STATUS | MEM_PWRDN |
| R-0h | | | | | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-39. MEDIA_CONTROLLER_MEM_PWRDN Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0h | |
| 1 | MEM_PWRDN_STATUS | R/W | 0h | Power down status from the Media Controller MEM<br>0x0 : memory not powered down<br>0x1 : memory powered down |
| 0 | MEM_PWRDN | R/W | 0h | Power down request to the Media Controller MEM<br>0x0 : No memory power down requested<br>0x1 : memory power down requested |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.37 SD_DAC_CTRL Register (offset = 670h) [reset = 18800C6h]

SD_DAC_CTRL is shown in Figure 3-38 and described in Table 3-40.

The SD_DAC_CTRL Register controls operation of the Standard Definition video DACs.

**Figure 3-38. SD_DAC_CTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | DAC1_cm4 | DAC1_cm3 |
| R-0h | | | | | | R/W-0h | R/W-1h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DAC1_cm2 | DAC1_cm1 | DAC1_cm0 | DAC0_cm4 | DAC0_cm3 | DAC0_cm2 | DAC0_cm1 | DAC0_cm0 |
| R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | TVOUTBYPASS |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACEN | INPUTINV | DEMEN | PBNDACz | PBNBGz | OFFMODE | SD_CALSEL | RESETz |
| R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R/W-1h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-40. SD_DAC_CTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | DAC1_cm4 | R/W | 0h | It should be low by default . |
| 24 | DAC1_cm3 | R/W | 1h | Optional control used only for dual-channel configuration. It should be low by default (single-channel configuration).<br>0x0 : Single channel<br>0x1 : Dual channel configuration (see section 7.5.1) |
| 23 | DAC1_cm2 | R/W | 1h | Optional control used only for dual-channel configuration. It should be low by default (single-channel configuration).<br>0x0 : Module configured as Luma video channel (dual-channel configuration) or Composite video channel (single-channel configuration)<br>0x1 : Module configured as Chroma video channel (dual-channel configuration, see section 7.5.1) |
| 22 | DAC1_cm1 | R/W | 0h | Optional control for lower output swing. It should be low by default (high output swing).<br>0x0 : High full-scale output swing<br>0x1 : Low full-scale output swing. |
| 21 | DAC1_cm0 | R/W | 0h | Optional control for internal current reference. It should be low by default (external current reference).<br>0x0 : External current reference (external resistor connected to rset)<br>0x1 : DO NOT USE. The switched-capacitor resistor has been removed. |
| 20 | DAC0_cm4 | R/W | 0h | It should be low by default . |
| 19 | DAC0_cm3 | R/W | 1h | Optional control used only for dual-channel configuration. It should be low by default (single-channel configuration).<br>0x0 : Single channel<br>0x1 : Dual channel configuration (see section 7.5.1) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-40. SD_DAC_CTRL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | DAC0_cm2 | R/W | 0h | Optional control used only for dual-channel configuration. It should be low by default (single-channel configuration).<br>0x0 : Module configured as Luma video channel (dual-channel configuration) or Composite video channel (single-channel configuration)<br>0x1 : Module configured as Chroma video channel (dual-channel configuration, see section 7.5.1) |
| 17 | DAC0_cm1 | R/W | 0h | Optional control for lower output swing. It should be low by default (high output swing).<br>0x0 : High full-scale output swing<br>0x1 : Low full-scale output swing. |
| 16 | DAC0_cm0 | R/W | 0h | Optional control for internal current reference. It should be low by default (external current reference).<br>0x0 : External current reference (external resistor connected to rset)<br>0x1 : DO NOT USE. The switched-capacitor resistor has been removed. |
| 15-9 | Reserved | R | 0h | |
| 8 | TVOUTBYPASS | R/W | 0h | TVOUT Bypass Signal. signal. Active high. Video Buffer is bypassed when tvoutbypass is high. Static Signal |
| 7 | ACEN | R/W | 1h | AC coupling enable. Active high. Must be set low for DC coupling, high for AC coupling. Static signal. |
| 6 | INPUTINV | R/W | 1h | Inversion of din[9:0]. Active high. It allows polarity alignment with Video Encoder data. It should be LOW for inverted video input data and HIGH for non-inverted video input data. Static signal. |
| 5 | DEMEN | R/W | 0h | Dynamic Element Matching (DEM) enable. signal. Active high. When high DEM logic is enabled. Static Signal |
| 4 | PBNDACz | R/W | 0h | DAC Power-Down Control Input. Active low. When low puts the entire DAC channel in power down mode, except the Bandgap and the internal LDO.<br>0x0 : DAC is powered down<br>0x1 : DAC is functional |
| 3 | PBNBGz | R/W | 0h | Bandgap (BG) Power-Down Control. internal LDO in power down mode. Active low. When low puts Bandgap and internal LDO in power down mode.<br>0x0 : DAC BG is powereddown<br>0x1 : DAC BG is functional |
| 2 | OFFMODE | R/W | 1h | Master power-down pin from a permanent (always-on) supply. Active high. When high it powers down all analog blocks regardless of the state of other power control pins. This pin must be used to power down if VDD is turned off.<br>0x0 : DAC normal operation<br>0x1 : DAC powered down |
| 1 | SD_CALSEL | R/W | 1h | SD DAC Calibration Select Controls the video path mux between the Display Subsystem SD output and calibration register values.<br>0x0 : Output calibration value<br>0x1 : Normal operation (DSS output) |
| 0 | RESETz | R/W | 0h | Reset signal. Asynchronous. Active low. When low, it resets data registers to mid-code ( 10000000 ) which is the default value. This signal can only be toggled when VDDA is ON. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.38 SD_DAC0_CAL Register (offset = 674h) [reset = 0h]

SD_DAC0_CAL is shown in Figure 3-39 and described in Table 3-41.

The SD_DAC0_CAL Register is used to calibrate the output level of SD DAC 0 (SD Y). Users can use this register to drive a known value to the DAC input for performing system video level calibration.

**Figure 3-39. SD_DAC0_CAL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | SDDAC_0_CAL | |
| R-0h | | | | | | R/W-0h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SDDAC_0_CAL | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-41. SD_DAC0_CAL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | Reserved | R | 0h | |
| 9-0 | SDDAC_0_CAL | R/W | 0h | SD DAC 0 calibration value This is driven as a constant value to the DAC when SD_CALSEL in the SD_DAC_CTRL register is set to 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.39 SD_DAC1_CAL Register (offset = 678h) [reset = 0h]

SD_DAC1_CAL is shown in Figure 3-40 and described in Table 3-42.

The SD_DAC1_CAL Register is used to calibrate the output level of SD DAC 1 (SD C). Users can use this register to drive a known value to the DAC input for performing system video level calibration.

**Figure 3-40. SD_DAC1_CAL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | SDDAC_1_CAL | |
| R-0h | | | | | | R/W-0h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SDDAC_1_CAL | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-42. SD_DAC1_CAL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | Reserved | R | 0h | |
| 9-0 | SDDAC_1_CAL | R/W | 0h | SD DAC 1 calibration value This is driven as a constant value to the DAC when SD_CALSEL in the SD_DAC_CTRL register is set to 1. |

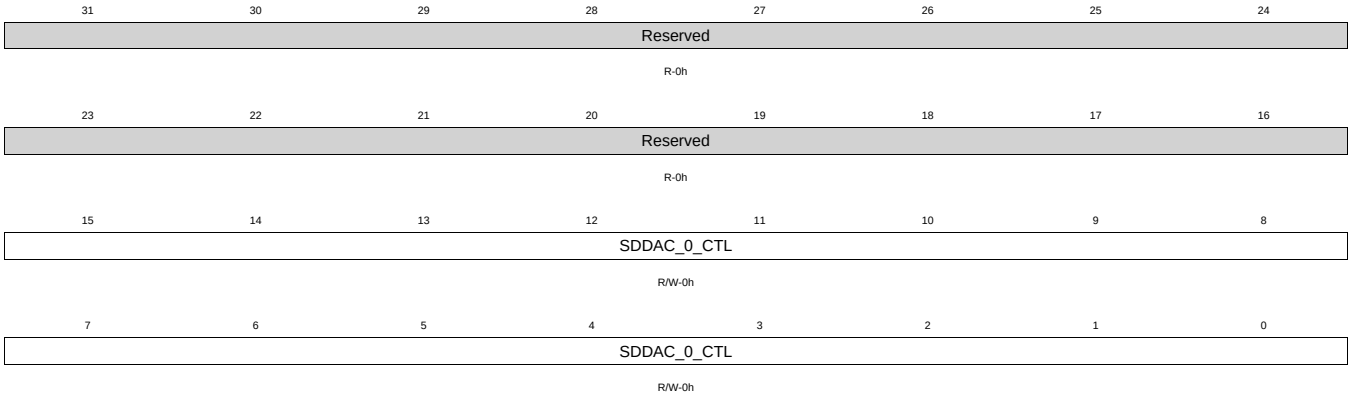

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.40 SD_DAC0_REGCTRL Register (offset = 67Ch) [reset = 0h]

SD_DAC0_REGCTRL is shown in Figure 3-41 and described in Table 3-43.

The SD_DAC0_REGCTRL Register is used to provide an alternate method to program the AVDAC s CTL interface (to be used only when the internal registers of the AVDAC needs to be tweaked).

**Figure 3-41. SD_DAC0_REGCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-43. SD_DAC0_REGCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-0 | SDDAC_0_CTL | R/W | 0h | CTL interface for the AVDAC |

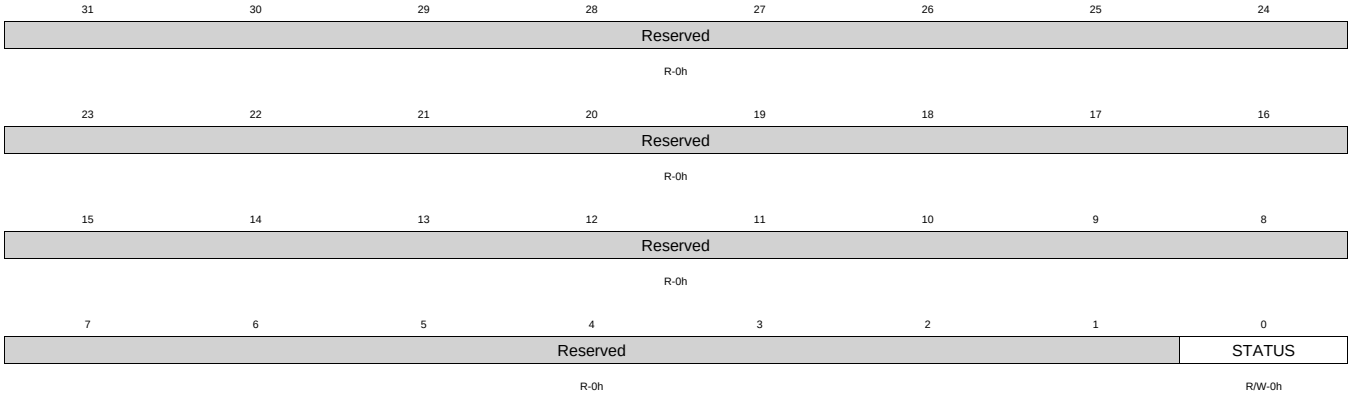

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.41 SD_DAC0_REGSTATUS Register (offset = 680h) [reset = 0h]

SD_DAC0_REGSTATUS is shown in Figure 3-42 and described in Table 3-44.

The SD_DAC0_REGSTATUS Register is used to provide to know the status of the last write to the AVDAC s internal registers (to be used only when the internal registers of the AVDAC needs to be tweaked).

**Figure 3-42. SD_DAC0_REGSTATUS Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-44. SD_DAC0_REGSTATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | STATUS | R/W | 0h | Write status for the AVDAC. Write 1 to clear. Output used only for Asynchronous mode of the CTL interface. It is by default low. It toggles high and then low when an asynchronous write on the ctl[15:0] bus has been acknowledged by the VDAC module (see section 5.3.4).<br>0x0 : Write pending<br>0x1 : Write acknowledged |

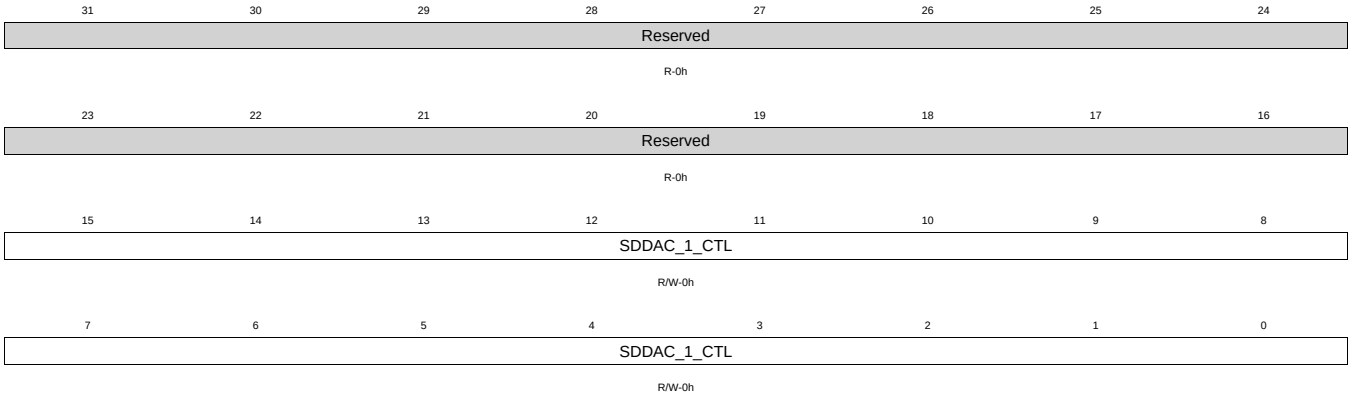

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.42 SD_DAC1_REGCTRL Register (offset = 684h) [reset = 0h]

SD_DAC1_REGCTRL is shown in Figure 3-43 and described in Table 3-45.

The SD_DAC1_REGCTRL Register is used to provide an alternate method to program the AVDAC s CTL interface (to be used only when the internal registers of the AVDAC needs to be tweaked).

**Figure 3-43. SD_DAC1_REGCTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-45. SD_DAC1_REGCTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-0 | SDDAC_1_CTL | R/W | 0h | CTL interface for the AVDAC |

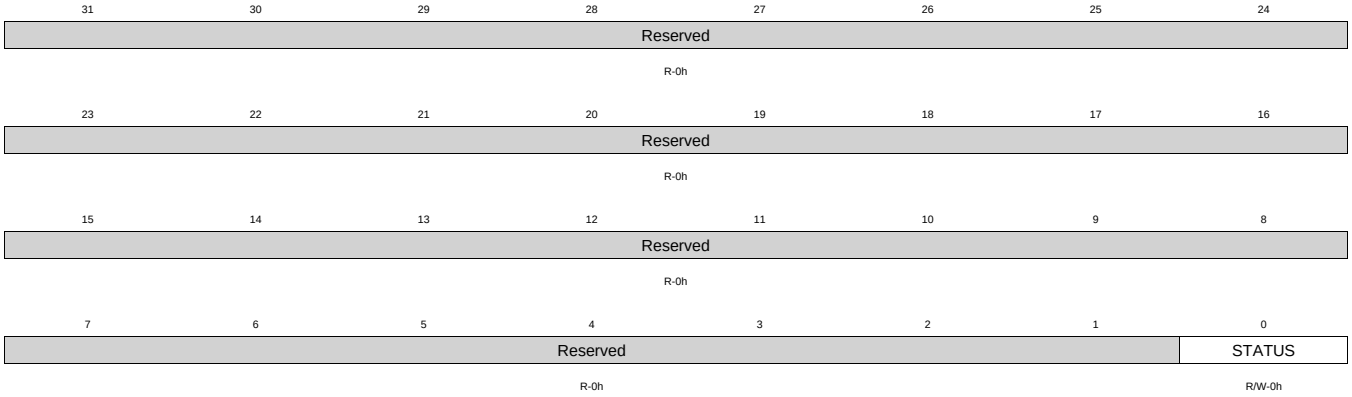

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.43 SD_DAC1_REGSTATUS Register (offset = 688h) [reset = 0h]

SD_DAC1_REGSTATUS is shown in Figure 3-44 and described in Table 3-46.

The SD_DAC1_REGSTATUS Register is used to provide to know the status of the last write to the AVDAC s internal registers (to be used only when the internal registers of the AVDAC needs to be tweaked).

**Figure 3-44. SD_DAC1_REGSTATUS Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-46. SD_DAC1_REGSTATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | STATUS | R/W | 0h | Write status for the AVDAC. Write 1 to clear. Output used only for Asynchronous mode of the CTL interface. It is by default low. It toggles high and then low when an asynchronous write on the ctl[15:0] bus has been acknowledged by the VDAC module (see section 5.3.4).<br>0x0 : Write pending<br>0x1 : Write acknowledged |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.44 EMIF_CLK_GATE Register (offset = 694h) [reset = 0h]

EMIF_CLK_GATE is shown in Figure 3-45 and described in Table 3-47.

This register provides a mechanism to control clock gating of DDR PHYs and see status of Self Refresh.

**Figure 3-45. EMIF_CLK_GATE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | DDR1_CKE_STATUS | DDR0_CKE_STATUS | DDRPHY1_CLK_GATE | DDRPHY0_CLK_GATE |
| R-0h | | | | R-0h | R-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-47. EMIF_CLK_GATE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | Reserved | R | 0h | |
| 3 | DDR1_CKE_STATUS | R | 0h | CKE ststus for DDR0<br>0x0 : Not in Self Refresh<br>0x1 : In Self Refresh |
| 2 | DDR0_CKE_STATUS | R | 0h | CKE ststus for DDR0<br>0x0 : Not in Self Refresh<br>0x1 : In Self Refresh |
| 1 | DDRPHY1_CLK_GATE | R/W | 0h | Enables/disables the clock to the DDR1 PHY<br>0x0 : DDR1 PHY clock enabled (running)<br>0x1 : DDR1 PHY clock disabled (stopped) |
| 0 | DDRPHY0_CLK_GATE | R/W | 0h | Enables/disables the clock to the DDR0 PHY<br>0x0 : DDR0 PHY clock enabled (running)<br>0x1 : DDR0 PHY clock disabled (stopped) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.45 SMRT_CTRL Register (offset = 6A0h) [reset = 0h]

SMRT_CTRL is shown in Figure 3-46 and described in Table 3-48.

The SMRT_CTRL Register reflects the control for the srsleep signal for the SRs .

**Figure 3-46. SMRT_CTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | SR3_SRSLLEP | SR2_SRSLLEP | SR1_SRSLLEP | SR0_SRSLLEP |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-48. SMRT_CTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | Reserved | R | 0h | |
| 3 | SR3_SRSLLEP | R/W | 0h | Disables or enables the sensor.<br>0x0 : disable sensor (SRSLEEP on sensor driven to 1)<br>0x1 : enable sensor (SRSLEEP on sensor driven to 0). |
| 2 | SR2_SRSLLEP | R/W | 0h | Disables or enables the sensor.<br>0x0 : disable sensor (SRSLEEP on sensor driven to 1)<br>0x1 : enable sensor (SRSLEEP on sensor driven to 0). |
| 1 | SR1_SRSLLEP | R/W | 0h | Disables or enables the sensor.<br>0x0 : disable sensor (SRSLEEP on sensor driven to 1)<br>0x1 : enable sensor (SRSLEEP on sensor driven to 0). |
| 0 | SR0_SRSLLEP | R/W | 0h | Disables or enables the sensor.<br>0x0 : disable sensor (SRSLEEP on sensor driven to 1)<br>0x1 : enable sensor (SRSLEEP on sensor driven to 0). |

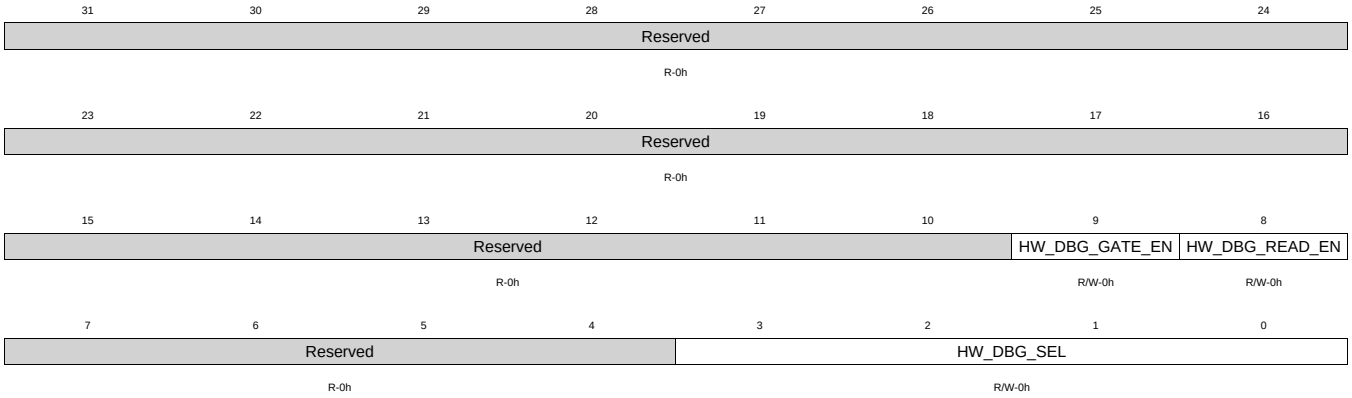

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.46 ARM_CORTEX_A8_HW_DBG_SEL Register (offset = 6A4h) [reset = 0h]

ARM_CORTEX_A8_HW_DBG_SEL is shown in Figure 3-47 and described in Table 3-49.

This register controls what kind of debug info is returned by ARM Cortex-A8.

**Figure 3-47. ARM_CORTEX_A8_HW_DBG_SEL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-49. ARM_CORTEX_A8_HW_DBG_SEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | Reserved | R | 0h | |
| 9 | HW_DBG_GATE_EN | R/W | 0h | To save power input to ARM_CORTEX_A8_HW_DBG_INFO is gated off to all zeros when HW_DBG_GATE_EN bit is low.<br>0x0 : Debug info gated off<br>0x1 : Debug info not gated off |
| 8 | HW_DBG_READ_EN | R/W | 0h | Enable disable reads of ARM_CORTEX_A8_HW_DBG_INFO<br>0x0 : Read of ARM_CORTEX_A8_HW_DBG_INFO is not enabled<br>0x1 : Read of ARM_CORTEX_A8_HW_DBG_INFO is enabled |
| 7-4 | Reserved | R | 0h | |
| 3-0 | HW_DBG_SEL | R/W | 0h | Selects which Group of signals are sent out to the ARM_CORTEX_A8_HW_DBG_INFO register<br>0x0 : Group 0<br>0x1 : Group 1<br>0x2 : Group 2<br>0x3 : Group 3<br>0x4 : Group 4<br>0x5 : Group 5<br>0x6 : Group 6<br>0x7 : Group 7<br>0x8 - 0xF : Reserved |

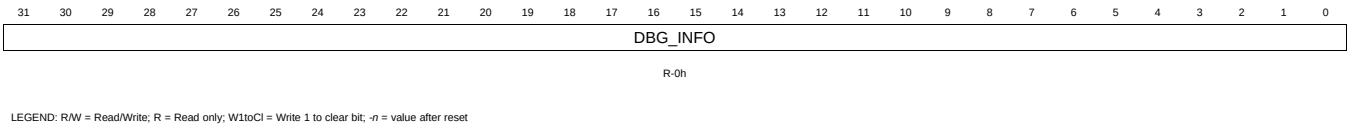

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.47 ARM_CORTEX_A8_HW_DBG_INFO Register (offset = 6A8h) [reset = 0h]

ARM_CORTEX_A8_HW_DBG_INFO is shown in Figure 3-48 and described in Table 3-50.

This register shows the debug info bus from ARM Cortex-A8. Please see below for definition. The selection bits are HW_DBG_SEL (3:0) in the MMR ARM_CORTEX_A8_HW_DBG_SEL

**Figure 3-48. ARM_CORTEX_A8_HW_DBG_INFO Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-50. ARM_CORTEX_A8_HW_DBG_INFO Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DBG_INFO | R | 0h | Debug info from ARM Cortex-A8. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.48 PRCM_DEBUG_ALWON_DEFAULT Register (offset = 6B0h) [reset = 3F7B0010h]

PRCM_DEBUG_ALWON_DEFAULT is shown in Figure 3-49 and described in Table 3-51.

This register is for PRCM debug.

**Figure 3-49. PRCM_DEBUG_ALWON_DEFAULT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | USB_CLK_OFF | Reserved | PCI_CLK_OFF | SATASS_CLK_OFF | DMM_EMIF_CLK_OFF | MEDIA_CONTROLLER_CLK_OFF |
| R-0h | | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | RTC_CLK_OFF | VCP_CLK_OFF | OCMC_RAM_CLK_OFF | MPU_CLK_OFF | Reserved | MMU_CLK_OFF | ETHERNET_CLK_OFF |
| R-0h | R-1h | R-1h | R-1h | R-1h | R-0h | R-1h | R-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | L3_SLOW_CLK_OFF | L3_MED_CLK_OFF | L3_FAST_CLK_OFF | Reserved | SYSCLK6_CLK_OFF | SYSCLK5_CLK_OFF | SYSCLK4_CLK_OFF |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | DEF_PWR_ON | HW_DBG_READ_EN | Reserved | | |
| R-0h | | | R-1h | R/W-0h | R-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-51. PRCM_DEBUG_ALWON_DEFAULT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29 | USB_CLK_OFF | R | 1h | USB_CLK status<br>0x0 : Usb clk is not idled<br>0x1 : Usb clk is idled |
| 28 | Reserved | R | 1h | Write 1 for future compatibility. Reads returns 1 |
| 27 | PCI_CLK_OFF | R | 1h | PCI_CLK status<br>0x0 : Pcie clk is not idled<br>0x1 : Pcie clk is idled |
| 26 | SATASS_CLK_OFF | R | 1h | SATASS_CLK status<br>0x0 : Satass clk is not idled<br>0x1 : Satass clk is idled |
| 25 | DMM_EMIF_CLK_OFF | R | 1h | DMM_EMIF_CLK status<br>0x0 : Dmm/Emif clk is not idled<br>0x1 : Dmm/Emif clk is idled |
| 24 | MEDIA_CONTROLLER_CLK_OFF | R | 1h | MEDIA_CONTROLLER_CLK status<br>0x0 : Media Controller clk is not idled<br>0x1 : Media Controller clk is idled |
| 23 | Reserved | R | 0h | |
| 22 | RTC_CLK_OFF | R | 1h | RTC_CLK status<br>0x0 : Rtc clk is not idled<br>0x1 : Rtc clk is idled |
| 21 | VCP_CLK_OFF | R | 1h | ETHERNET_CLK status<br>0x0 : Vcp clk is not idled<br>0x1 : Vcp clk is idled |
| 20 | OCMC_RAM_CLK_OFF | R | 1h | OCMC_RAM_CLK status<br>0x0 : OCMC_RAM clk is not idled<br>0x1 : OCMC_RAM clk is idled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-51. PRCM_DEBUG_ALWON_DEFAULT Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MPU_CLK_OFF | R | 1h | MPU clk status<br>0x0 : Mpu clk is not idled<br>0x1 : Mpu clk is idled |
| 18 | Reserved | R | 0h | |
| 17 | MMU_CLK_OFF | R | 1h | MMU CLK status<br>0x0 : MMU clk is not idled<br>0x1 : Ethernet clk is idled |
| 16 | ETHERNET_CLK_OFF | R | 1h | ETHERNET_CLK status<br>0x0 : Ethernet clk is not idled<br>0x1 : Ethernet clk is idled |
| 15 | Reserved | R | 0h | |
| 14 | L3_SLOW_CLK_OFF | R | 0h | L3_SLOW_CLK status<br>0x0 : L3_slow clk is not idled<br>0x1 : L3_slow clk is idled |
| 13 | L3_MED_CLK_OFF | R | 0h | L3_MED_CLK status<br>0x0 : L3_med clk is not idled<br>0x1 : L3_med clk is idled |
| 12 | L3_FAST_CLK_OFF | R | 0h | L3_FAST_CLK status<br>0x0 : L3_fast clk is not idled<br>0x1 : L3_fast clk is idled |
| 11 | Reserved | R | 0h | |
| 10 | SYSCLK6_CLK_OFF | R | 0h | SYSCLK6 status<br>0x0 : Sysclk6 clk is on<br>0x1 : Sysclk6 clk is off |
| 9 | SYSCLK5_CLK_OFF | R | 0h | SYSCLK5 status<br>0x0 : Sysclk5 clk is on<br>0x1 : Sysclk5 clk is off |
| 8 | SYSCLK4_CLK_OFF | R | 0h | SYSCLK4 status<br>0x0 : Sysclk4 clk is on<br>0x1 : Sysclk4 clk is off |
| 7-5 | Reserved | R | 0h | |
| 4 | DEF_PWR_ON | R | 1h | Default domain power status<br>0x0 : Default domain is not powered up<br>0x1 : Default domain is powered up |
| 3 | HW_DBG_READ_EN | R/W | 0h | Enables the read of the other bitfileds in this MMR.<br>0x0 : Read of PRCM_DEBUG_ALWON_DEFAULT is not enabled<br>0x1 : Read of PRCM_DEBUG_ALWON_DEFAULT is enabled |
| 2-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.49 PRCM_DEBUG_PD_DOMAIN_STATUS Register (offset = 6B4h) [reset = 871C7180h]

PRCM_DEBUG_PD_DOMAIN_STATUS is shown in Figure 3-50 and described in Table 3-52.

This register is for PRCM debug.

**Figure 3-50. PRCM_DEBUG_PD_DOMAIN_STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SGX_PD_SGX_CLK_STATUS_GLUE | SGX_PD_CLK_STATUS_PRCM | SGX_PD_MEM_ON_GLUE | SGX_PD_SWITCH_ON_GLUE | SGX_PD_ON_PRCM | DSS_PD_HDMI_CLK_STATUS_GLUE | DSS_PD_DSS_CLK_STATUS_GLUE | DSS_PD_CLK_STATUS_PRCM |
| R-1h | R-0h | R-0h | R-0h | R-0h | R-1h | R-1h | R-1h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DSS_PD_MEM_ON_GLUE | DSS_PD_SWITCH_ON_GLUE | DSS_PD_ON_PRCM | ISP_PD_FDIF_CLK_STATUS_GLUE | ISP_PD_ISS_CLK_STATUS_GLUE | ISP_PD_CLK_STATUS_PRCM | ISP_PD_MEM_ON_GLUE | ISP_PD_SWITCH_ON_GLUE |
| R-0h | R-0h | R-0h | R-1h | R-1h | R-1h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ISP_PD_ON_PRCM | HDVICP_PD_SLX_CLK_STATUS_GLUE | HDVICP_PD_HDVICP_CLK_STATUS_GLUE | HDVICP_PD_HDVICP_CLK_STATUS_PRCM | HDVICP_PD_MEM_ON_GLUE | HDVICP_PD_SWITCH_ON_GLUE | HDVICP_PD_ON_PRCM | ACTIVE_PD_C674x_DSP_CLK_STATUS_GLUE |
| R-0h | R-1h | R-1h | R-1h | R-0h | R-0h | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACTIVE_PD_C674x_DSP_CLK_STATUS_PRCM | ACTIVE_PD_MEM_ON_GLUE | ACTIVE_PD_SWITCH_ON_GLUE | ACTIVE_PD_ON_PRCM | HW_DBG_READ_EN | Reserved | | |
| R-1h | R-0h | R-0h | R-0h | R/W-0h | R-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-52. PRCM_DEBUG_PD_DOMAIN_STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SGX_PD_SGX_CLK_STATUS_GLUE | R | 1h | SGX domain SGX clk status<br>0x0 : SGX clk is not idled status from glue<br>0x1 : SGX clk is idled status from glue |
| 30 | SGX_PD_CLK_STATUS_PRCM | R | 0h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 29 | SGX_PD_MEM_ON_GLUE | R | 0h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 28 | SGX_PD_SWITCH_ON_GLUE | R | 0h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 27 | SGX_PD_ON_PRCM | R | 0h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 26 | DSS_PD_HDMI_CLK_STATUS_GLUE | R | 1h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 25 | DSS_PD_DSS_CLK_STATUS_GLUE | R | 1h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 24 | DSS_PD_CLK_STATUS_PRCM | R | 1h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-52. PRCM_DEBUG_PD_DOMAIN_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | DSS_PD_MEM_ON_GLUE | R | 0h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 22 | DSS_PD_SWITCH_ON_GLUE | R | 0h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 21 | DSS_PD_ON_PRCM | R | 0h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 20 | ISP_PD_FDIF_CLK_STATUS_GLUE | R | 1h | ISS Power domain FDIF clk status<br>0x0 : FDIF clk is not idled status from glue<br>0x1 : FDIF clk is idled status from glue |
| 19 | ISP_PD_ISS_CLK_STATUS_GLUE | R | 1h | Active domain power switch status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 18 | ISP_PD_CLK_STATUS_PRCM | R | 1h | ISS Power domain clk status<br>0x0 : ISP domain clks are not idled<br>0x1 : ISP domain clks are idled |
| 17 | ISP_PD_MEM_ON_GLUE | R | 0h | ISS power domain memory status<br>0x0 : ISP domain memories are off<br>0x1 : ISP domain memories are on |
| 16 | ISP_PD_SWITCH_ON_GLUE | R | 0h | ISS power domain switch status<br>0x0 : ISP domain switches are off<br>0x1 : ISP domain switches are on |
| 15 | ISP_PD_ON_PRCM | R | 0h | ISS Power domain status<br>0x0 : ISP domain is not powered up<br>0x1 : ISP domain is powered up |
| 14 | HDVICP_PD_SLX_CLK_STATUS_GLUE | R | 1h | HDVICP PD SLX clk status<br>0x0 : HDVICP slx clk is not idled status from glue<br>0x1 : HDVICP slx clk is idled status from glue |
| 13 | HDVICP_PD_HDVICP_CLK_STATUS_GLUE | R | 1h | HDVICP glue clk status<br>0x0 : HDVICP clk is not idled status from glue<br>0x1 : HDVICP clk is idled status from glue |
| 12 | HDVICP_PD_HDVICP_CLK_STATUS_PRCM | R | 1h | HDVICP clk status<br>0x0 : HDVICP domain clks are not idled<br>0x1 : HDVICP domain clks are idled |
| 11 | HDVICP_PD_MEM_ON_GLUE | R | 0h | HDVICP Power domain memory glue status<br>0x0 : HDVICP domain memories are off<br>0x1 : VA domain memories are on |
| 10 | HDVICP_PD_SWITCH_ON_GLUE | R | 0h | HDVICP power domain switch status<br>0x0 : HDVICP domain switches are off<br>0x1 : HDVICP domain switches are on |
| 9 | HDVICP_PD_ON_PRCM | R | 0h | HDVICP power domain status<br>0x0 : HDVICP domain is not powered up<br>0x1 : HDVICP domain is powered up |
| 8 | ACTIVE_PD_C674x_DSP_CLK_STATUS_GLUE | R | 1h | C674x_DSP clk status from the glue logic<br>0x0 : C674x_DSP clk is not idled status from glue<br>0x1 : C674x_DSP clk is idled status from glue |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-52. PRCM_DEBUG_PD_DOMAIN_STATUS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | ACTIVE_PD_C674x_DSP_CLK_STATUS_PRCM | R | 1h | C674x_DSP clk status<br>0x0 : C674x_DSP clk is not idled<br>0x1 : C674x_DSP clk is idled |
| 6 | ACTIVE_PD_MEM_ON_GLUE | R | 0h | Active domain memory status<br>0x0 : Active domain memories are on<br>0x1 : Active domain memories are off |
| 5 | ACTIVE_PD_SWITCH_ON_GLUE | R | 0h | Active domain power switch status<br>0x0 : Active domain switches are off<br>0x1 : Active domain switches are on |
| 4 | ACTIVE_PD_ON_PRCM | R | 0h | Active domain power status<br>0x0 : Active domain is not powered up<br>0x1 : Active domain is powered up |
| 3 | HW_DBG_READ_EN | R/W | 0h | Enables the read of the other bitfileds in this MMR.<br>0x0 : Read of PRCM_DEBUG_PD_DOMAIN_STATUS is not enabled<br>0x1 : Read of PRCM_DEBUG_PD_DOMAIN_STATUS is enabled |
| 2-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.50 PCIE_PLLCFG0 Register (offset = 6D8h) [reset = 40007000h]

PCIE_PLLCFG0 is shown in Figure 3-51 and described in Table 3-53.

This is connected to "CF_APLL_0" on SERDES ANA Initial Settings required prior to power up sequence: PCIE_100M: 4000 7000 ; SATA_20M: C000 7060 ; SATA_100M: 4000 7060

**Figure 3-51. PCIE_PLLCFG0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SEL_IN_FREQ | DIGCLRZ | Reserved | | AMUXSEL | TESTCLKMUXSEL | APLL_MISC_CTRL | |
| R/W-0h | R/W-1h | R/W-0h | | R/W-0h | R/W-0h | R/W-0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| APLL_MISC_CTRL | | | | DIS_REFCLK | EN_3P | PFD_CLR | CLK_FLIP |
| R/W-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EN_RTRIM | EN_MEAS | EN_LATCH | | CP_CTRL | | | |
| R/W-0h | R/W-1h | R/W-3h | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESVALUE | | C1_2X | ENDIGLDO | APLL_CP_CURR | ENBGSC_REF | ENPLLLDO | ENPLL |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-53. PCIE_PLLCFG0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SEL_IN_FREQ | R/W | 0h | Select input frequency:<br>0b0 = 100 M<br>0b1 = 20 M |
| 30 | DIGCLRZ | R/W | 1h | CLRZ for APLL DIG and DLL DIG. Used to reset flops and state machines |
| 29-28 | Reserved | R/W | 0h | Reserved |
| 27 | AMUXSEL | R/W | 0h | AMUX select :<br>0b0 = KVCO<br>0b1 = VRTRIM |
| 26 | TESTCLKMUXSEL | R/W | 0h | Selects test clock mux :<br>0b0=REF Clock<br>0b1=Divided clock |
| 25-20 | APLL_MISC_CTRL | R/W | 0h | Used by APLL DIG<br>**Bitfield [21:20]** encoding is :<br>0b00 = No step<br>0b01 = 75% to 100%<br>0b10 = 75% to 87.5% to 100% in 60, 20 , 20 cycle steps<br>0b11 = same as 0b10 but in 40, 40, 40 cycle steps<br>**Bit 22** specifies power saving :<br>0b0 = Default mode<br>0b1 = Used for debug<br>**Bit 23** function is:<br>0b0 = fixed divider<br>0b1 = programmable divider<br>**Bit 24** is for 50 MHz rtrim disable<br>**Bit 25** controls the lock speed :<br>0 = 128 clock<br>1 = 256 clock |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-53. PCIE_PLLCFG0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | DIS_REFCLK | R/W | 0h | Disable CML that control FREF output clock buffer. 0=Enable, 1= Disable |
| 18 | EN_3P | R/W | 0h | Post_Cent_PG1.0. When high, enables the SC circuit in the APLL LF Pre_Cent_PG1.0. Logic '1' generates single phase output and puts other phases at '0' using 3-phase ring oscillator 0 = 4-phase output 1 = single phase output |
| 17 | PFD_CLR | R/W | 0h | To open PLL loop; 0 = Closed loop, 1 = Open loop |
| 16 | CLK_FLIP | R/W | 0h | Logic '1' flips the output clock phase |
| 15 | EN_RTRIM | R/W | 0h | Enable resistor calibration 0=OFF, 1=ON |
| 14 | EN_MEAS | R/W | 1h | To enable measurement circuit inside APLL ANA to control AMUX output. 0 = OFF, 1 = ON |
| 13-12 | EN_LATCH | R/W | 3h | Enables output latch in differential ring to control power. Designer should be contacted while changing this value:<br>0b00 = All latch on<br>0b01 = Remove 33%<br>0b10 = Remove 16%<br>0b11 = Remove 50 % |
| 11-8 | CP_CTRL | R/W | 0h | Charge pump control bit +- 50 % :<br>0b0000 = Nom<br>0b0001 = +13%<br>0b0010 = +27%<br>0b0011 = +50%<br>0b0100 = -20%<br>0b1000 = -27%<br>0b1010 = -50%<br>All others are not valid. |
| 7-6 | RESVALUE | R/W | 0h | Sets loop filter resistor value:<br>0b00 = 1 Kohm for PCIe<br>0b01 = 1.67 kohm for eSata<br>0b10 = 2.5 kohm for SSC<br>0b11 is not valid |
| 5 | C1_2X | R/W | 0h | Increase filter capacitance by 2x in low reference (20 MHz) clock:<br>0b0 = 120 pF for PCIe<br>0b1 = 240 pF for eSATA and SSC |
| 4 | ENDIGLDO | R/W | 0h | Enable DIG LDO. Required to wake pll up for plll dig |
| 3 | APLL_CP_CURR | R/W | 0h | Post_Cent_PG1.0. When high, this bit increases the APLL CP current. Pre_Cent_PG1.0. Unused. |
| 2 | ENBGSC_REF | R/W | 0h | Enable switched cap bias current module/REFGEN |
| 1 | ENPLLLDO | R/W | 0h | Enable PLL LDO |
| 0 | ENPLL | R/W | 0h | Enable PLL |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.51 PCIE_PLLCFG1 Register (offset = 6DCh) [reset = 0h]

PCIE_PLLCFG1 is shown in Figure 3-52 and described in Table 3-54.

This is connected to "CF_APLL_1" on SERDES ANA Initial Settings required prior to power up sequence: PCIE_100M: 0064 0000 ; SATA_100MHz: 912C 0000 ; SATA_20MHz: C02C 0000

**Figure 3-52. PCIE_PLLCFG1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ENSATAMODE | PLLREFSEL | NP1_DIV_INT | | | | MDIVINT | |
| R/W-0h | R/W-0h | R/W-0h | | | | R/W-0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| MDIVINT | | | | | | MDIVFRAC | |
| R/W-0h | | | | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| MDIVFRAC | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MDIVFRAC | | EN_CLKAUX | EN_CLK125m | EN_CLK100m | EN_CLK50m | ENSSC | MDIVPULSE |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-54. PCIE_PLLCFG1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | ENSATAMODE | R/W | 0h | Mode select 0 = PCIe (2.5 GHz) / 1 = SATA (1.5 GHz). Used by APLLDIG and TRX_DIG for different purpose. Needs to configure this bit to get ethernet clocks |
| 30 | PLLREFSEL | R/W | 0h | Chose reference clock value 0 = 100 MHz / 1 = 20 MHz (Note: only needs to be set if using fixed FB divider)<br>For fixed divider mode [31:30] defines division ratio as:<br>0b00 = 25<br>0b01 = 125<br>0b10 = 15<br>0b11 = 75 |
| 29-26 | NP1_DIV_INT | R/W | 0h | Integer value of N+1 divider 0 value maps to 1 4 value maps to 5 |
| 25-18 | MDIVINT | R/W | 0h | 8-bit integer value for M divider (Osc/ref). Valid range is 4 to 255 |
| 17-6 | MDIVFRAC | R/W | 0h | 12-bit fractional value for M divider (Osc/ref) |
| 5 | EN_CLKAUX | R/W | 0h | Old -- Enable for auxilary clock (testclk) clock 0 = Enable / 1 = Disable Corrected on 11/30/2010 Updated -- Enable for auxilary clock (testclk) clock 0 = Disable / 1 = Enable |
| 4 | EN_CLK125m | R/W | 0h | Enable for 125MHz clock 0 = Disable / 1 = Enable. This is one of the Ethernet clocks |
| 3 | EN_CLK100m | R/W | 0h | Enable for 100MHz clock 0 = Disable / 1 = Enable |
| 2 | EN_CLK50m | R/W | 0h | Enable for 50MHz clock 0 = Disable / 1 = Enable. This is one of the Ethernet clocks |
| 1 | ENSSC | R/W | 0h | Enable spread specturm support 0 = Disable / 1 = Enable |
| 0 | MDIVPULSE | R/W | 0h | Update M div when pulse is high |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.52 PCIE_PLLCFG2 Register (offset = 6E0h) [reset = 0h]

PCIE_PLLCFG2 is shown in Figure 3-53 and described in Table 3-55.

This is connected to "CF_APLL_2" on SERDES ANA Initial Settings required prior to power up sequence: PCIE: 0000 0000 -- SATA: 0000 0000 (both modes)

**Figure 3-53. PCIE_PLLCFG2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SSCDNSPREAD | SSCMANT | | | | | | |
| R/W-0h | R/W-0h | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| SSCEXPO | | | SSCFRSPREAD | | | | |
| R/W-0h | | | R/W-0h | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SSCFRSPREAD | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SSCFRSPREAD | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-55. PCIE_PLLCFG2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SSCDNSPREAD | R/W | 0h | SSC downspread |
| 30-24 | SSCMANT | R/W | 0h | Mantesa portion of the modified frequency in SSC operation |
| 23-21 | SSCEXPO | R/W | 0h | Exponent portion of the modified frequency in SSC operation |
| 20-0 | SSCFRSPREAD | R/W | 0h | SSC frequency spread |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.53 PCIE_PLLCFG3 Register (offset = 6E4h) [reset = 4000000h]

PCIE_PLLCFG3 is shown in Figure 3-54 and described in Table 3-56.

This is connected to "CF_APLL_3" on SERDES ANA Initial Settings required prior to power up sequence: 0040 08E0

**Figure 3-54. PCIE_PLLCFG3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | DIGLDO_BYPASS | DIGLDO_PULLDOWNZ | DIGLDO_DIS_SC_PROT | DIGLDO_EN_SUB_REGULATION |
| R/W-0h | | | | R/W-0h | R/W-1h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIGLDO_EN_HP_CAPLESSMODE | DIGLDO_EN_CAPLESSMODE | DIGLDO_EN_LP_CAPLESSMODE | DIGLDO_VSET | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PLLLDO_CTRL_15_14_13 | | | PLLLDO_EN_RETENTION | PLLLDO_EN_LDO_STABLE | PLLLDO_EN_BYPASS | PLLLDO_EN_SC_PROT | PLLLDO_EN_EXT_CAP |
| R/W-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PLLLDO_EN_BUF_CUR | PLLLDO_EN_LP | PLLLDO_CTRL_TRIM | | | | | PLLLDO_CTRL_0_UNUSED |
| R/W-0h | R/W-0h | R/W-0h | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-56. PCIE_PLLCFG3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R/W | 0h | Reserved for future use |
| 27 | DIGLDO_BYPASS | R/W | 0h | Digital signal when high puts the DIG LDO in bypass mode. Need to be careful while exercising this mode, the 1.8V external supply must be lowered before asserting this bit. |
| 26 | DIGLDO_PULLDOWNZ | R/W | 1h | Digital signal when high disables all the pull down paths for LDO output |
| 25 | DIGLDO_DIS_SC_PROT | R/W | 0h | Digital signal when high disables short circuit protection |
| 24 | DIGLDO_EN_SUB_REGULATION | R/W | 0h | Digital signal when high disables sub regulation |
| 23 | DIGLDO_EN_HP_CAPLESSMODE | R/W | 0h | Digital signal when high enables high performance Capless mode. In this mode, quiescent LDO current increases by 240uA |
| 22 | DIGLDO_EN_CAPLESSMODE | R/W | 0h | Digital signal when high enables Capless Mode. In particular: (1) The 800fF capacitor between VDDAR and the gate of the power FET is bypassed (2) The output of the error amplifier connects to a 30pF capacitor and the other terminal of the capacitor connects to a 40k resistor whose other terminal goes to VSSA. This resistor is also bypassed. |
| 21 | DIGLDO_EN_LP_CAPLESSMODE | R/W | 0h | Digital signal when high enables Low Power Capless Mode. Quiescent LDO current increases by 240uA typical when this bit is set to low. |
| 20-16 | DIGLDO_VSET | R/W | 0h | Changed on Jan 12th 2011 Post PG1.0 Inversion in VSET bits, 1 and 0 swap Based upon Sami's measured data and DIGLDO spice sims VSET(4:0) VDDAR (V) 11111 1.20 11110 1.19 11101 1.18 11100 1.17 11011 1.16 11010 1.15 11001 1.14 11000 1.13 10111 1.12 10110 1.11 10101 1.10 10100 1.09 10011 1.08 10010 1.07 10001 1.06 10000 1.05 01111 1.36 01110 1.35 01101 1.34 01100 1.33 01011 1.32 01010 1.31 01001 1.30 01000 1.29 00111 1.28 00110 1.27 00101 1.26 00100 1.25 00011 1.24 00010 1.23 00001 1.22 00000 1.21 |
| 15-13 | PLLLDO_CTRL_15_14_13 | R/W | 0h | Reserved for future use |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-56. PCIE_PLLCFG3 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | PLLLDO_EN_RETENTION | R/W | 0h | Enable retention mode |
| 11 | PLLLDO_EN_LDO_STABLE | R/W | 0h | LDO Stable Signal |
| 10 | PLLLDO_EN_BYPASS | R/W | 0h | Enable bypass mode |
| 9 | PLLLDO_EN_SC_PROT | R/W | 0h | Enable short circuit protection |
| 8 | PLLLDO_EN_EXT_CAP | R/W | 0h | Enable external cap mode |
| 7 | PLLLDO_EN_BUF_CUR | R/W | 0h | Enable Increased Buffer Current |
| 6 | PLLLDO_EN_LP | R/W | 0h | Enable Low Power |
| 5-1 | PLLLDO_CTRL_TRIM | R/W | 0h | TRIM_BITS vout 00000 1.000 00001 1.025 00010 1.050 00011 1.075 00100 1.100 00101 1.125 00110 1.150 00111 1.175 01000 1.200 01001 1.225 01010 1.250 01011 1.275 01100 1.300 01101 1.325 01110 1.350 01111 1.375 10000 1.400 10001 1.425 10010 1.450 10011 1.475 10100 1.500 10101 1.525 10110 1.550 10111 1.575 11000 1.600 |
| 0 | PLLLDO_CTRL_0_UNUSED | R/W | 0h | No Modelling |

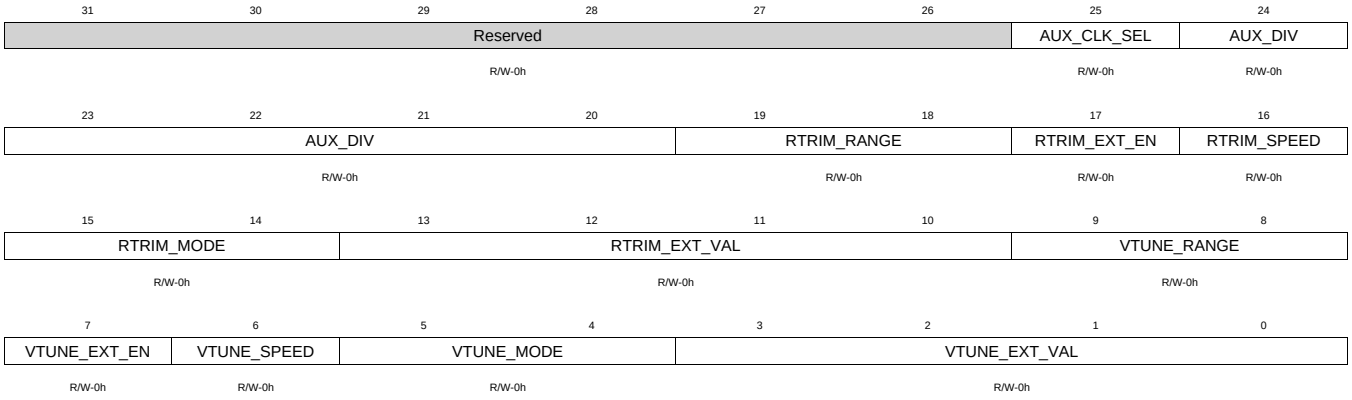

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.54 PCIE_PLLCFG4 Register (offset = 6E8h) [reset = 0h]

PCIE_PLLCFG4 is shown in Figure 3-55 and described in Table 3-57.

This is connected to "CF_APLL_4" on SERDES ANA Initial Settings required prior to power up sequence: PCIE: 0000 609C -- SATA: 0000 0000 (both modes)

**Figure 3-55. PCIE_PLLCFG4 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-57. PCIE_PLLCFG4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R/W | 0h | Spare |
| 25 | AUX_CLK_SEL | R/W | 0h | 0=Divided, 1=ref-clk |
| 24-20 | AUX_DIV | R/W | 0h | Integer value of aux divider; Valid range is 2-31 |
| 19-18 | RTRIM_RANGE | R/W | 0h | Range for rtrim:<br>0b00 = 0.9-0.8<br>0b01 = 1.0-0.9<br>0b10 = 1.1-1.0<br>0b11 = 1.1-0.9 |
| 17 | RTRIM_EXT_EN | R/W | 0h | Select loop starting point. 0= mid-code 0000 / 1 = external input RTRIM_EXT_VAL |
| 16 | RTRIM_SPEED | R/W | 0h | Selection on wait for # REFCLKs after previous update; 0 = 128 / 1 = 256 |
| 15-14 | RTRIM_MODE | R/W | 0h | Loop mode:<br>"0b00" = off<br>"0b10" = on - freeze (after lock freeze loop)<br>"0bx1" = on - continuous (after lock keep lock running) |
| 13-10 | RTRIM_EXT_VAL | R/W | 0h | 2 s compliment rtrim control loop value. Valid range : 0111 (max) to 0000 (mid) to 1000 (min) |
| 9-8 | VTUNE_RANGE | R/W | 0h | Range for Vtune:<br>0b00 = 0.9-0.8<br>0b01 = 1.0-0.9<br>0b10 = 1.1-1.0<br>0b11 = 1.1-0.9 |
| 7 | VTUNE_EXT_EN | R/W | 0h | Select loop starting point. 0= mid-code 0000 / 1 = external input VTUNE_EXT_VAL |
| 6 | VTUNE_SPEED | R/W | 0h | Selection on wait for # REFCLKs after previous update; 0 = 128 / 1 = 256 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-57. PCIE_PLLCFG4 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5-4 | VTUNE_MODE | R/W | 0h | Loop mode :<br>"0b00" = off<br>"0b10" = on - freeze (after lock freeze loop)<br>"0bx1" = on - continuous (after lock keep lock running) |
| 3-0 | VTUNE_EXT_VAL | R/W | 0h | 2 s compliment vtune control loop value. Valid ranges : 0100 (max) to 0000 (mid) to 1100 (min) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.55 PCIE_PLLSTATUS Register (offset = 6ECh) [reset = 0h]

PCIE_PLLSTATUS is shown in Figure 3-56 and described in Table 3-58.

Initial Settings required prior to power up sequence:

**Figure 3-56. PCIE_PLLSTATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R/W-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | RTRIMSTS | | | |
| R/W-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VTUNESTS | | | | PLLDIG_EN | PLLANA_EN | PLLSSC_EN | PLL_LOCK |
| R/W-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-58. PCIE_PLLSTATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R/W | 0h | Not used |
| 15-12 | Reserved | R/W | 0h | Not used |
| 11-8 | RTRIMSTS | R/W | 0h | Value of the rtrim loop (may not be static in continuous mode of operation) |
| 7-4 | VTUNESTS | R/W | 0h | Value of the vtune control loop (may not be static in continuous mode of operation) |
| 3 | PLLDIG_EN | R/W | 0h | Digital EN. Same as APLL DIG STS 2 internally |
| 2 | PLLANA_EN | R/W | 0h | Analog EN. Same as APLL DIG STS 4 internally |
| 1 | PLLSSC_EN | R/W | 0h | SSC is engaged |
| 0 | PLL_LOCK | R/W | 0h | APLL locked |

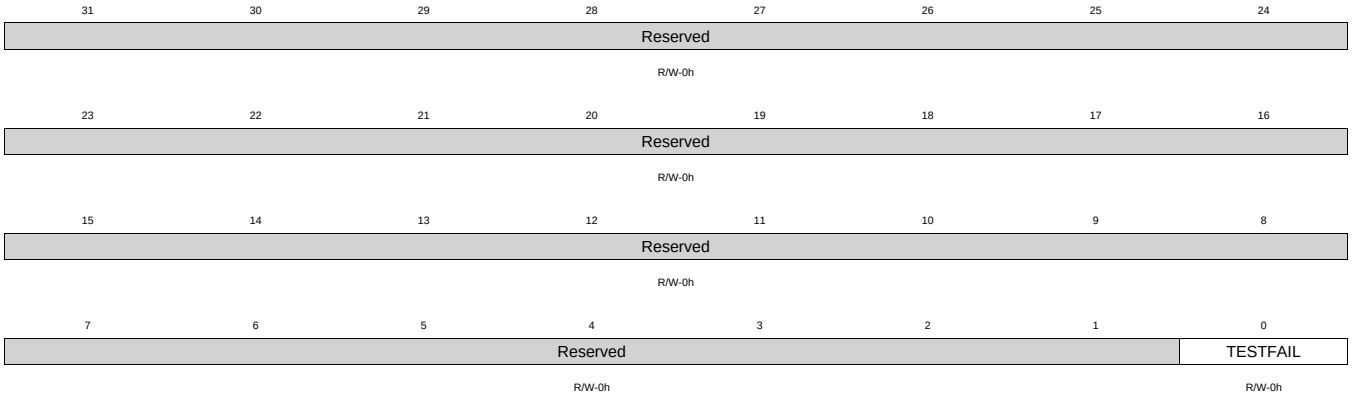

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.56 PCIE_RXSTATUS Register (offset = 6F0h) [reset = 0h]

PCIE_RXSTATUS is shown in Figure 3-57 and described in Table 3-59.

PCIE SerDes Receive Status register

**Figure 3-57. PCIE_RXSTATUS Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-59. PCIE_RXSTATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R/W | 0h | Reserved, driven low. |
| 0 | TESTFAIL | R/W | 0h | Test failure. Driven high when an error is encountered during a test sequence executed on channel i. Synchronous to RXBCLKIN[i]. |

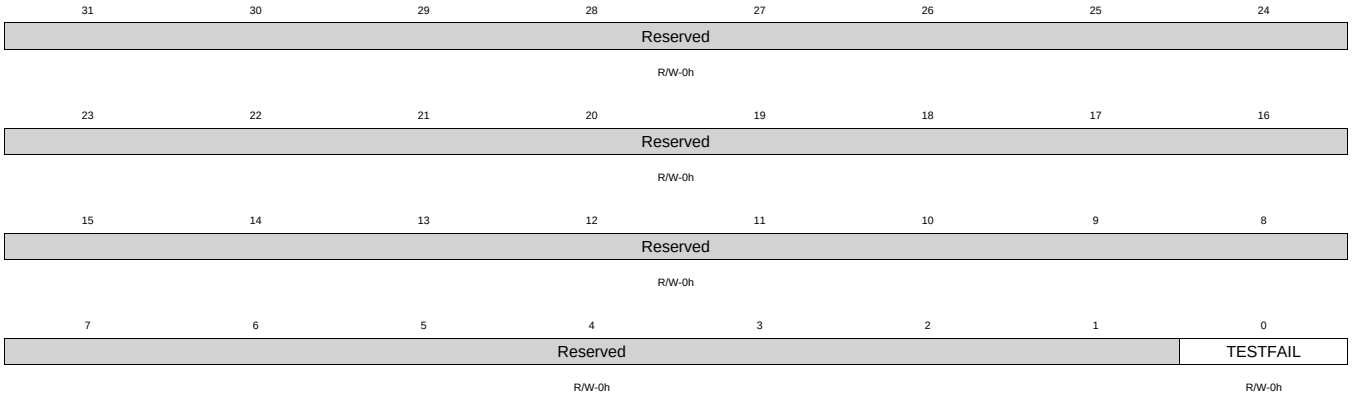

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.57 PCIE_TXSTATUS Register (offset = 6F4h) [reset = 0h]

PCIE_TXSTATUS is shown in Figure 3-58 and described in Table 3-60.

This register reports PCIE SerDes Transmit Status.

**Figure 3-58. PCIE_TXSTATUS Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-60. PCIE_TXSTATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R/W | 0h | Reserved, driven low. |
| 0 | TESTFAIL | R/W | 0h | Test failure. Driven high when an error is encountered during a test sequence executed on channel i, or if an over or underflow occurs in the Tx serialiser when ENFTP = 0. Synchronous to TXBCLKIN[i]. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.58 SATA_PLLCFG0 Register (offset = 720h) [reset = C0007160h]

SATA_PLLCFG0 is shown in Figure 3-59 and described in Table 3-61.

SATA PLL Configuration 0. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings

**Figure 3-59. SATA_PLLCFG0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SEL_IN_FREQ | DIGCLRZ | Reserved | | AMUXSEL | TESTCLKMUXSEL | APLL_MISC_CTRL | |
| R/W-1h | R/W-1h | R-0h | | R/W-0h | R/W-0h | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| APLL_MISC_CTRL | | | | DIS_REFCLK | EN_3P | PFD_CLR | CLK_FLIP |
| R/W-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| EN_RTRIM | EN_MEAS | EN_LATCH | | CP_CTRL | | | |
| R/W-0h | R/W-1h | R/W-3h | | R/W-1h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESVALUE | | C1_2X | ENDIGLDO | APLL_CP_CURR | ENBGSC_REF | ENPLLLDO | ENPLL |
| R/W-1h | | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-61. SATA_PLLCFG0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SEL_IN_FREQ | R/W | 1h | Select input frequency. 0=100M, 1=20M |
| 30 | DIGCLRZ | R/W | 1h | CLRZ for APLL DIG and DLL DIG. Used to reset flops and state machines |
| 29-28 | Reserved | R | 0h | |
| 27 | AMUXSEL | R/W | 0h | AMUX select; 0=KVCO, 1=VRTRIM |
| 26 | TESTCLKMUXSEL | R/W | 0h | Selects test clock mux; 0=REF Clock, 1=Divided clock |
| 25-20 | APLL_MISC_CTRL | R/W | 0h | Used by APLL DIG |
| 19 | DIS_REFCLK | R/W | 0h | 0=Disable, 1= Enable |
| 18 | EN_3P | R/W | 0h | Logic 1 enables the SC circuit in the APLL LF |
| 17 | PFD_CLR | R/W | 0h | Reset for phase-frequency detector |
| 16 | CLK_FLIP | R/W | 0h | Logic 1 flips the output clock phase |
| 15 | EN_RTRIM | R/W | 0h | Enable resistor calibration 0=OFF, 1=ON |
| 14 | EN_MEAS | R/W | 1h | To enable measurement circuit inside APLL ANA |
| 13-12 | EN_LATCH | R/W | 3h | Enables output latch in differential ring |
| 11-8 | CP_CTRL | R/W | 1h | Charge pump control bit +- 50 % |
| 7-6 | RESVALUE | R/W | 1h | Sets loop filter resistor value |
| 5 | C1_2X | R/W | 1h | Increase filter capacitance by 2x in low reference (20 MHz) clock |
| 4 | ENDIGLDO | R/W | 0h | Enable DIG LDO. Required to wake pll up for plll dig |
| 3 | APLL_CP_CURR | R/W | 0h | When high, this bit increases the APLL CP current |
| 2 | ENBGSC_REF | R/W | 0h | ENBGSC_REF |
| 1 | ENPLLLDO | R/W | 0h | Enable PLL LDO |
| 0 | ENPLL | R/W | 0h | Enable PLL |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.59 SATA_PLLCFG1 Register (offset = 724h) [reset = 812C0000h]

SATA_PLLCFG1 is shown in Figure 3-60 and described in Table 3-62.

SATA PLL Configuration 1. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings.

**Figure 3-60. SATA_PLLCFG1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ENSATAMODE | PLLREFSEL | NP1_DIV_INT | | | | MDIVINT | |
| R/W-1h | R/W-0h | R/W-0h | | | | R/W-4Bh | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| MDIVINT | | | | | | MDIVFRAC | |
| R/W-4Bh | | | | | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| MDIVFRAC | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MDIVFRAC | | EN_CLKAUX | EN_CLK125M | EN_CLK100M | EN_CLK50M | ENSSC | MDIVPULSE |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-62. SATA_PLLCFG1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | ENSATAMODE | R/W | 1h | Mode select 0 = PCIe (2.5 GHz) / 1 = SATA (1.5 GHz). Used by APLLDIG and TRX_DIG for different purpose. Needs to configure this bit to get Ethernet clocks. |
| 30 | PLLREFSEL | R/W | 0h | Chose reference clock value 0 = 100 MHz / 1 = 20 MHz (Note: only needs to be set if using DLL outputs) |
| 29-26 | NP1_DIV_INT | R/W | 0h | Integer value of N+1 divider |
| 25-18 | MDIVINT | R/W | 4Bh | 8-bit value, which is used as integer portion of feedback divider |
| 17-6 | MDIVFRAC | R/W | 0h | 12-bit value, which is used as fractional portion of feedback divider |
| 5 | EN_CLKAUX | R/W | 0h | Enable for Auxilary clock 0 = Disable / 1 = Enable |
| 4 | EN_CLK125M | R/W | 0h | Enable for 125MHz clock 0 = Disable / 1 = Enable |
| 3 | EN_CLK100M | R/W | 0h | Enable for 100MHz clock 0 = Disable / 1 = Enable |
| 2 | EN_CLK50M | R/W | 0h | EN_CLK50M Enable for 50MHz clock 0 = Disable / 1 = Enable |
| 1 | ENSSC | R/W | 0h | Enable spread pectrum support |
| 0 | MDIVPULSE | R/W | 0h | Update M div when pulse is high |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.60 SATA_PLLCFG2 Register (offset = 728h) [reset = 0h]

SATA_PLLCFG2 is shown in Figure 3-61 and described in Table 3-63.

SATA PLL Configuration 2. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings.

**Figure 3-61. SATA_PLLCFG2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SSCDNSPREAD | SSCMANT | | | | | | |
| R/W-0h | R/W-0h | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| SSCEXPO | | | SSCFRSPREAD | | | | |
| R/W-0h | | | R/W-0h | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SSCFRSPREAD | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SSCFRSPREAD | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-63. SATA_PLLCFG2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SSCDNSPREAD | R/W | 0h | SSC downspread |
| 30-24 | SSCMANT | R/W | 0h | Mantesa portion of the modified frequency in SSC operation |
| 23-21 | SSCEXPO | R/W | 0h | Exponent portion of the modified frequency in SSC operation |
| 20-0 | SSCFRSPREAD | R/W | 0h | SSC frequency spread |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.61 SATA_PLLCFG3 Register (offset = 72Ch) [reset = 4000000h]

SATA_PLLCFG3 is shown in Figure 3-62 and described in Table 3-64.

SATA_PLLCFG3 Description. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings.

**Figure 3-62. SATA_PLLCFG3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | DIGLDO_BYPASS | DIGLDO_PULLDOWNZ | DIGLDO_DIS_SC_PROT | DIGLDO_EN_SUB_REGULATION |
| R/W-0h | | | | R/W-0h | R/W-1h | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| DIGLDO_EN_HP_CAPLESSMODE | DIGLDO_EN_CAPLESSMODE | DIGLDO_EN_LP_CAPLESSMODE | DIGLDO_VSET | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | PLLLDO_EN_RETENTION | PLLLDO_EN_LDO_STABLE | PLLLDO_EN_BYPASS | PLLLDO_EN_SC_PROT | PLLLDO_EN_EXT_CAP |
| R/W-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PLLLDO_EN_BUF_CUR | PLLLDO_EN_LP | PLLLDO_CTRL_TRIM | | | | | PLLLDO_CTRL |
| R/W-0h | R/W-0h | R/W-0h | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-64. SATA_PLLCFG3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R/W | 0h | Reserved |
| 27 | DIGLDO_BYPASS | R/W | 0h | Digital signal when high puts the DIG LDO in bypass mode. Caution Need to be careful while exercising this mode, the 1.8V external supply must be lowered before asserting this bit. |
| 26 | DIGLDO_PULLDOWNZ | R/W | 1h | Digital signal when high disables all the pull down paths for LDO output |
| 25 | DIGLDO_DIS_SC_PROT | R/W | 0h | Digital signal when high disables short circuit protection |
| 24 | DIGLDO_EN_SUB_REGULATION | R/W | 0h | Digital signal when high disables sub regulation |
| 23 | DIGLDO_EN_HP_CAPLESSMODE | R/W | 0h | Digital signal when high enables high performance Capless mode. In this mode, quiescent LDO current increases by 240uA |
| 22 | DIGLDO_EN_CAPLESSMODE | R/W | 0h | Digital signal when high enables Capless Mode. In particular: (1) The 800fF capacitor between VDDAR and the gate of the power FET is bypassed (2) The output of the error amplifier connects to a 30pF capacitor and the other terminal of the capacitor connects to a 40k resistor whose other terminal goes to VSSA. This resistor is also bypassed. |
| 21 | DIGLDO_EN_LP_CAPLESSMODE | R/W | 0h | Digital signal when high enables Low Power Capless Mode. Quiescent LDO current increases by 240uA typical when this bit is set to low. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-64. SATA_PLLCFG3 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20-16 | DIGLDO_VSET | R/W | 0h | VSET(4:0) VDDAR(V)<br>0b11111 = 1.20 V<br>0b11110 = 1.19 V<br>0b11101 = 1.18 V<br>0b11100 = 1.17 V<br>0b11011 = 1.16 V<br>0b11010 = 1.15 V<br>0b11001 = 1.14 V<br>0b11000 = 1.13 V<br>0b10111 = 1.12 V<br>0b10110 = 1.11 V<br>0b10101 = 1.10 V<br>0b10100 = 1.09 V<br>0b10011 = 1.08 V<br>0b10010 = 1.07 V<br>0b10001 = 1.06 V<br>0b10000 = 1.05 V<br>0b01111 = 1.36 V<br>0b01110 = 1.35 V<br>0b01101 = 1.34 V<br>0b01100 = 1.33 V<br>0b01011 = 1.32 V<br>0b01010 = 1.31 V<br>0b01001 = 1.30 V<br>0b01000 = 1.29 V<br>0b00111 = 1.28 V<br>0b00110 = 1.27 V<br>0b00101 = 1.26 V<br>0b00100 = 1.25 V<br>0b00011 = 1.24 V<br>0b00010 = 1.23 V<br>0b00001 = 1.22 V<br>0b00000 = 1.21 V |
| 15-13 | Reserved | R/W | 0h | Reserved |
| 12 | PLLLDO_EN_RETENTION | R/W | 0h | Enable retention mode |
| 11 | PLLLDO_EN_LDO_STABLE | R/W | 0h | LDO Stable Signal |
| 10 | PLLLDO_EN_BYPASS | R/W | 0h | Enable bypass mode |
| 9 | PLLLDO_EN_SC_PROT | R/W | 0h | Enable bypass mode |
| 8 | PLLLDO_EN_EXT_CAP | R/W | 0h | Enable external cap mode |
| 7 | PLLLDO_EN_BUF_CUR | R/W | 0h | Enable Increased Buffer Current |
| 6 | PLLLDO_EN_LP | R/W | 0h | Enable Low Power |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-64. SATA_PLLCFG3 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5-1 | PLLLDO_CTRL_TRIM | R/W | 0h | TRIM_BITS vout<br>0b00000 = 1.000 V<br>0b00001 = 1.025 V<br>0b00010 = 1.050 V<br>0b00011 = 1.075 V<br>0b00100 = 1.100 V<br>0b00101 = 1.125 V<br>0b00110 = 1.150 V<br>0b00111 = 1.175 V<br>0b01000 = 1.200 V<br>0b01001 = 1.225 V<br>0b01010 = 1.250 V<br>0b01011 = 1.275 V<br>0b01100 = 1.300 V<br>0b01101 = 1.325 V<br>0b01110 = 1.350 V<br>0b01111 = 1.375 V<br>0b10000 = 1.400 V<br>0b10001 = 1.425 V<br>0b10010 = 1.450 V<br>0b10011 = 1.475 V<br>0b10100 = 1.500 V<br>0b10101 = 1.525 V<br>0b10110 = 1.550 V<br>0b10111 = 1.575 V<br>0b11000 = 1.600 V |
| 0 | PLLLDO_CTRL | R/W | 0h | PLLLDO_CTRL_0_UNUSED |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.62 SATA_PLLCFG4 Register (offset = 730h) [reset = 0h]

SATA_PLLCFG4 is shown in Figure 3-63 and described in Table 3-65.

5PCIe PLL Configuration 4. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings.

**Figure 3-63. SATA_PLLCFG4 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | AUX_CLK_SEL | AUX_DIV |
| R/W-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| AUX_DIV | | | | RTRIM_RANGE | | RTRIM_EXT_EN | RTRIM_SPEED |
| R/W-0h | | | | R/W-0h | | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RTRIM_MODE | | RTRIM_EXT_VAL | | | | VTUNE_RANGE | |
| R/W-0h | | R/W-0h | | | | R/W-0h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VTUNE_EXT_EN | VTUNE_SPEED | VTUNE_MODE | | VTUNE_EXT_VAL | | | |
| R/W-0h | R/W-0h | R/W-0h | | R/W-0h | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-65. SATA_PLLCFG4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R/W | 0h | |
| 25 | AUX_CLK_SEL | R/W | 0h | 0=Divided, 1=ref-clk |
| 24-20 | AUX_DIV | R/W | 0h | Aux divider control. Valid range is 2-31 |
| 19-18 | RTRIM_RANGE | R/W | 0h | Range for rtrim :<br>0b00 = 0.9-0.8<br>0b01 = 1.0-0.9<br>0b10 = 1.1-1.0<br>0b11 = 1.1-0.9 |
| 17 | RTRIM_EXT_EN | R/W | 0h | Select loop starting point. 0= mid-code 0000 / 1 = external input RTRIM_EXT_VAL |
| 16 | RTRIM_SPEED | R/W | 0h | Selection on wait for # REFCLKs after previous update; 0 = 128 / 1 = 256. |
| 15-14 | RTRIM_MODE | R/W | 0h | Loop mode: 0b0x = off, 0b10 = on freeze (after lock freeze loop), 0b11 = on continuous (after lock keep lock running) |
| 13-10 | RTRIM_EXT_VAL | R/W | 0h | 2 s compliment vtune control loop value. Valid range : 0100 (max) to 0000 (mid) to 1100 (min) |
| 9-8 | VTUNE_RANGE | R/W | 0h | Range for Vtune:<br>0b00=0.9-0.8<br>0b01=1.0-0.9<br>0b10=1.1-1.0<br>0b11= 1.1-0.9 |
| 7 | VTUNE_EXT_EN | R/W | 0h | Select loop starting point. 0= mid-code 0000 / 1 = external input VTUNE_EXT_VAL |
| 6 | VTUNE_SPEED | R/W | 0h | Selection on wait for # REFCLKs after previous update; 0 = 128 / 1 = 256 |
| 5-4 | VTUNE_MODE | R/W | 0h | Loop mode: 0b0x = off, 0b10 = on freeze (after lock freeze loop), 0b11 = on continuous (after lock keep lock running) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 3-65. SATA_PLLCFG4 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3-0 | VTUNE_EXT_VAL | R/W | 0h | 2 s compliment vtune control loop value. Valid range : 0100 (max) to 0000 (mid) to 1100 (min) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.63 SATA_TXSTATUS Register (offset = 73Ch) [reset = 0h]

SATA_TXSTATUS is shown in Figure 3-64 and described in Table 3-66.

SATA TX Status. Note: Please note that this registers s reset behavior depends upon the Reset Isolation Register (RESET_ISO) settings.

**Figure 3-64. SATA_TXSTATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | TESTFAIL |
| R-0h | | | | | | | R-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-66. SATA_TXSTATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | TESTFAIL | R | 0h | Test failure. Driven high when an error is encountered during a test sequence executed on channel i, or if an over or under ow occurs in the Tx serialiser when ENFTP = 0. Synchronous to TXBCLKIN[i]. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.64 SATA_TESTCFG Register (offset = 740h) [reset = 0h]

SATA_TESTCFG is shown in Figure 3-65 and described in Table 3-67.

SATA TEST Config.

**Figure 3-65. SATA_TESTCFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R/W-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | RX_TESTPATT | | |
| R/W-0h | | | | | R/W-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-67. SATA_TESTCFG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R/W | 0h | |
| 2-0 | RX_TESTPATT | R/W | 0h | Enables and selects test patterns.<br>0b000 = Reserved<br>0b100 = Uses a 31-bit LFSR with feedback polynomial x31+x28+1.<br>0b001 = An alternating 0/1 pattern with a period of 2 UI.<br>0b010 = Uses a 7-bit LFSR with feedback polynomial x7+x6+1.<br>0b011 = Uses a 23-bit LFSR with feedback polynomial x23+x18+1.<br>0b101 - 0b111 = Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.65 VDD_MPU_OPP_050 Register (offset = 770h) [reset = 0h]

VDD_MPU_OPP_050 is shown in Figure 3-66 and described in Table 3-68.

Ntarget value for OPP50.

**Figure 3-66. VDD_MPU_OPP_050 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-68. VDD_MPU_OPP_050 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for MPU Voltage domain s OPP50.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.66 VDD_MPU_OPP_100 Register (offset = 774h) [reset = 0h]

VDD_MPU_OPP_100 is shown in Figure 3-67 and described in Table 3-69.

Ntarget value for OPP100.

**Figure 3-67. VDD_MPU_OPP_100 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-69. VDD_MPU_OPP_100 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for MPU Voltage domain s OPP100.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.67 VDD_MPU_OPP_120 Register (offset = 778h) [reset = 0h]

VDD_MPU_OPP_120 is shown in Figure 3-68 and described in Table 3-70.

Ntarget value for OPP120.

**Figure 3-68. VDD_MPU_OPP_120 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-70. VDD_MPU_OPP_120 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for MPU Voltage domain s OPP119.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.68 VDD_MPU_OPP_166 Register (offset = 77Ch) [reset = 0h]

VDD_MPU_OPP_166 is shown in Figure 3-69 and described in Table 3-71.

Ntarget value for OPP166.

**Figure 3-69. VDD_MPU_OPP_166 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-71. VDD_MPU_OPP_166 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for MPU Voltage domain s OPPTURBO.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.69 VDD_C674x_DSP_OPP_050 Register (offset = 788h) [reset = 0h]

VDD_C674x_DSP_OPP_050 is shown in Figure 3-70 and described in Table 3-72.

Ntarget value for OPP50.

**Figure 3-70. VDD_C674x_DSP_OPP_050 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-72. VDD_C674x_DSP_OPP_050 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for C674x_DSP Voltage domain s OPP50.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.70 VDD_C674x_DSP_OPP_100 Register (offset = 78Ch) [reset = 0h]

VDD_C674x_DSP_OPP_100 is shown in Figure 3-71 and described in Table 3-73.

Ntarget value for OPP100.

**Figure 3-71. VDD_C674x_DSP_OPP_100 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-73. VDD_C674x_DSP_OPP_100 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for C674x_DSP Voltage domain s OPP100.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.71 VDD_C674x_DSP_OPP_120 Register (offset = 790h) [reset = 0h]

VDD_C674x_DSP_OPP_120 is shown in Figure 3-72 and described in Table 3-74.

Ntarget value for OPP120.

**Figure 3-72. VDD_C674x_DSP_OPP_120 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-74. VDD_C674x_DSP_OPP_120 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for C674x_DSP Voltage domain s OPP119.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.72 VDD_C674x_DSP_OPP_166 Register (offset = 794h) [reset = 0h]

VDD_C674x_DSP_OPP_166 is shown in Figure 3-73 and described in Table 3-75.

Ntarget value for OPP166.

**Figure 3-73. VDD_C674x_DSP_OPP_166 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-75. VDD_C674x_DSP_OPP_166 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for C674x_DSP Voltage domain s OPPTURBO.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.73 VDD_HDVICP_OPP_050 Register (offset = 7A0h) [reset = 0h]

VDD_HDVICP_OPP_050 is shown in Figure 3-74 and described in Table 3-76.

Ntarget value for OPP50.

**Figure 3-74. VDD_HDVICP_OPP_050 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-76. VDD_HDVICP_OPP_050 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for HDVICP Voltage domain s OPP50.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.74 VDD_HDVICP_OPP_100 Register (offset = 7A4h) [reset = 0h]

VDD_HDVICP_OPP_100 is shown in Figure 3-75 and described in Table 3-77.

Ntarget value for OPP100.

**Figure 3-75. VDD_HDVICP_OPP_100 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-77. VDD_HDVICP_OPP_100 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for HDVICP Voltage domain s OPP100.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.75 VDD_HDVICP_OPP_120 Register (offset = 7A8h) [reset = 0h]

VDD_HDVICP_OPP_120 is shown in Figure 3-76 and described in Table 3-78.

Ntarget value for OPP120.

**Figure 3-76. VDD_HDVICP_OPP_120 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-78. VDD_HDVICP_OPP_120 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for HDVICP Voltage domain s OPP119.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.76 VDD_HDVICP_OPP_166 Register (offset = 7ACh) [reset = 0h]

VDD_HDVICP_OPP_166 is shown in Figure 3-77 and described in Table 3-79.

Ntarget value for OPP166.

**Figure 3-77. VDD_HDVICP_OPP_166 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-79. VDD_HDVICP_OPP_166 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for HDVICP Voltage domain s OPPTURBO.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.77 VDD_CORE_OPP_050 Register (offset = 7B8h) [reset = 0h]

VDD_CORE_OPP_050 is shown in Figure 3-78 and described in Table 3-80.

Ntarget value for OPP50.

**Figure 3-78. VDD_CORE_OPP_050 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-80. VDD_CORE_OPP_050 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for CORE Voltage domain s OPP50.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.78 VDD_CORE_OPP_100 Register (offset = 7BCh) [reset = 0h]

VDD_CORE_OPP_100 is shown in Figure 3-79 and described in Table 3-81.

Ntarget value for OPP100.

**Figure 3-79. VDD_CORE_OPP_100 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-81. VDD_CORE_OPP_100 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for CORE Voltage domain s OPP100.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.79 VDD_CORE_OPP_120 Register (offset = 7C0h) [reset = 0h]

VDD_CORE_OPP_120 is shown in Figure 3-80 and described in Table 3-82.

Ntarget value for OPP120.

**Figure 3-80. VDD_CORE_OPP_120 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-82. VDD_CORE_OPP_120 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for CORE Voltage domain s OPP119.Default comes from Efuse. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.80 VDD_CORE_OPP_166 Register (offset = 7C4h) [reset = 0h]

VDD_CORE_OPP_166 is shown in Figure 3-81 and described in Table 3-83.

Ntarget value for OPP166.

**Figure 3-81. VDD_CORE_OPP_166 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NTARGET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-83. VDD_CORE_OPP_166 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-0 | NTARGET | R/W | 0h | Ntarget value for CORE Voltage domain s OPPTURBO.Default comes from Efuse. |

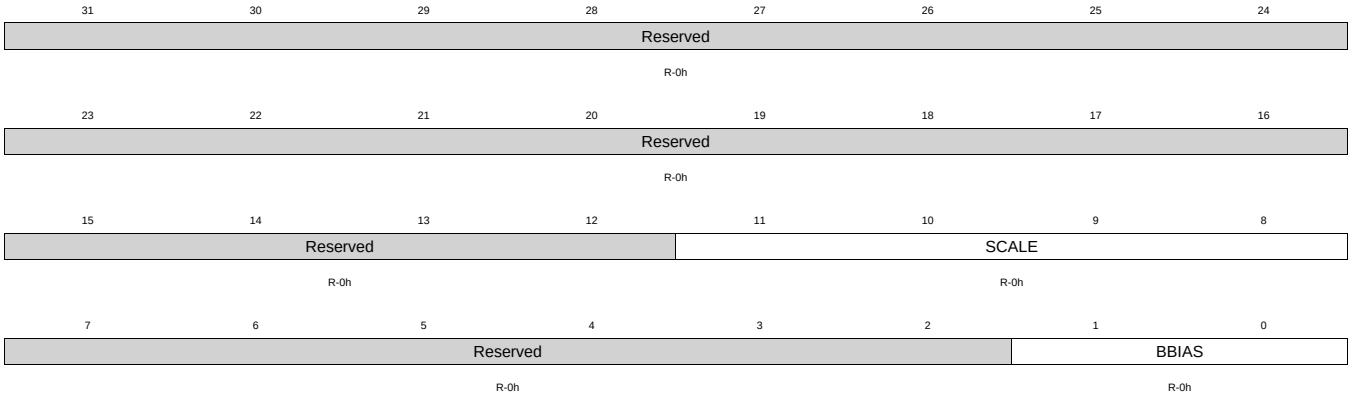

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.81 BB_SCALE Register (offset = 7D0h) [reset = 0h]

BB_SCALE is shown in Figure 3-82 and described in Table 3-84.

Back Bias and Dynamic Core Voltage Scaling Register

**Figure 3-82. BB_SCALE Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-84. BB_SCALE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | Reserved | R | 0h | |
| 11-8 | SCALE | R | 0h | Dynamic Core Voltage Scaling For class 0 smart reflex. |
| 7-2 | Reserved | R | 0h | |
| 1-0 | BBIAS | R | 0h | BBIAS value from Efuse |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.82 USB_VID_PID Register (offset = 7F4h) [reset = 0h]

USB_VID_PID is shown in Figure 3-83 and described in Table 3-85.

From TI Efuse, use only for TI defined USB VID/PID option.

**Figure 3-83. USB_VID_PID Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| USB_VID | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| USB_VID | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| USB_PID | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USB_PID | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-85. USB_VID_PID Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | USB_VID | R | 0h | USB Vendor ID (TI Programmed Efuse) |
| 15-0 | USB_PID | R | 0h | USB Product ID (TI programmed Efuse) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.83 PCIE_VID_PID Register (offset = 7F8h) [reset = 0h]

PCIE_VID_PID is shown in Figure 3-84 and described in Table 3-86.

From TI Efuse, use only for TI defined USB VID/PID option.

**Figure 3-84. PCIE_VID_PID Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PCIE_VID | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PCIE_VID | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| PCIE_PID | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PCIE_PID | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-86. PCIE_VID_PID Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | PCIE_VID | R | 0h | PCIE Vendor ID (TI Programmed Efuse) |
| 15-0 | PCIE_PID | R | 0h | PCIEUSB Product ID (TI programmed Efuse) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.84 PINCNTL[270:1] Register (offset = 800h) [reset = 40000h]

PINCNTL[270:1] is shown in Figure 3-85 and described in Table 3-87.

This register is used to configure the device pin muxing and other IO characteristics. Refer to device *Data Manual* for full details.

**Figure 3-85. PINCNTL[270:1] Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | | | | | | |
| 0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | | | Rsvd3 | Rsvd2 | PULLTYPESEL | PULLUDEN |
| 0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Rsvd1 | | | | | | | |
| 0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MUXMODE | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-87. PINCNTL[270:1] Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Rsvd4 | | 0h | Reserved-read returns 0 |
| 19 | Rsvd3 | R/W | 0h | Reserved - This field must always be written with the reset (default) value. Reset value is Pad dependent |
| 18 | Rsvd2 | R/W | 0h | Reserved - This field must always be written as '1'. |
| 17 | PULLTYPESEL | R/W | 0h | Pad Pullup / Pulldown Type. Reset value is Pad dependent<br>0x0 : Pulldown (PD) selected<br>0x1 : Pullup (PU) selected |
| 16 | PULLUDEN | R/W | 0h | Pullup / Pulldown Enable. Reset value is Pad dependent<br>0x0 : PU/PD enabled<br>0x1 : PU/PD disabled |
| 15-8 | Rsvd1 | | 0h | Reserved Read returns 0 |
| 7-0 | MUXMODE | R/W | 0h | Pin Multiplexing Selection Ctonrols which multiplexed signal is assigned to the corresponding pin. Refer to device data manual for details. |

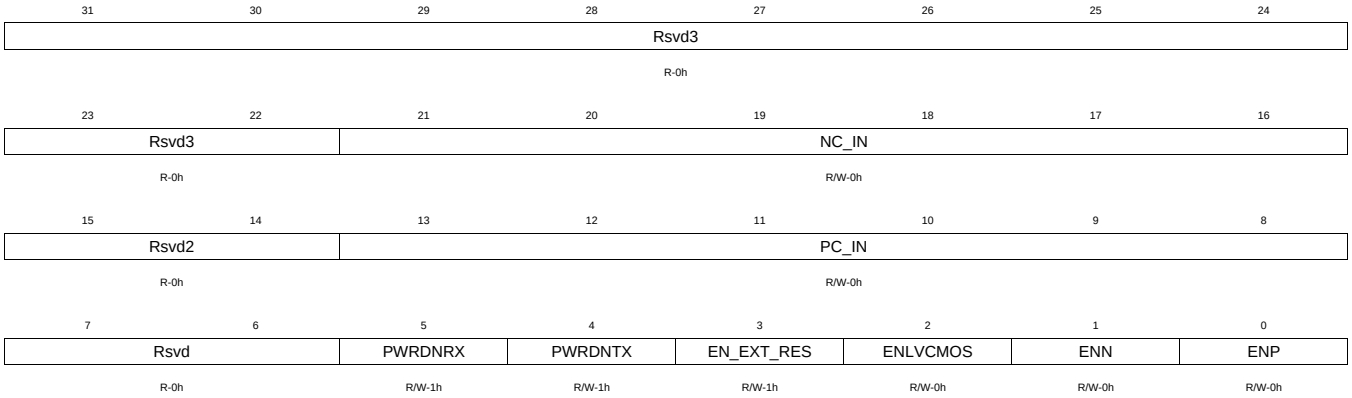

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.85 MLBP_SIG_IO_CTRL Register (offset = E18h) [reset = 38h]

MLBP_SIG_IO_CTRL is shown in Figure 3-86 and described in Table 3-88.

This register controls the MLBP SIG pins IO characteristics.

**Figure 3-86. MLBP_SIG_IO_CTRL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-88. MLBP_SIG_IO_CTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | Rsvd3 | R | 0h | Reserved-read returns 0 |
| 21-16 | NC_IN | R/W | 0h | MLB LVDS Output Drive Trim value. If register defaults to state of 0x0, then it should be set to the recommended state of 0x1A. If the register defaults to non-zero, that implies it was set to optimal value during manufacturing. |
| 15-14 | Rsvd2 | R | 0h | Reserved-read returns 0 |
| 13-8 | PC_IN | R/W | 0h | MLB LVDS Output Drive Trim value. If register defaults to state of 0x0, then it should be set to the recommended state of 0x24. If the register defaults to non-zero, that implies it was set to optimal value during manufacturing. |
| 7-6 | Rsvd | R | 0h | Reserved-read returns 0 |
| 5 | PWRDNRX | R/W | 1h | Power down control for MLB LVDS receiver. A 1 written to this register will power down the receiver. A 0 written to this register will power on the receiver. By default, the receiver is in power down mode. |
| 4 | PWRDNTX | R/W | 1h | Power down control for MLB LVDS transmitter. A 1 written to this register will power down the transmitter. A 0 written to this register will power on the transmitter. By default, the transmitter is in power down mode. |
| 3 | EN_EXT_RES | R/W | 1h | Disables the internal resistors required for proper MLB LVDS operation. This register should always be programmed to 0x0 |
| 2 | ENLVCMOS | R/W | 0h | This registers is used to disable the MLB LVDS mode and program the pins as GPIO. Each pad acts as 50ohm LVCMOS TX and LVCMOS RX when this bit is set to 1. Each pad is MLB LVDS when this bit is set to zero. |
| 1 | ENN | R/W | 0h | Receiver enable for LVCMOS GPIO mode. Programming this register to 1 enables the MLBP_SIGN pad to be a GPIO input. Programming this register to 0 disables the input and MLBP_SIGN pad is only an output. |
| 0 | ENP | R/W | 0h | Receiver enable for LVCMOS GPIO mode. Programming this register to 1 enables the MLBP_SIGP pad to be a GPIO input. Programming this register to 0 disables the input and MLBP_SIGP pad is only an output. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.86 MLBP_DAT_IO_CTRL Register (offset = E1Ch) [reset = 38h]

MLBP_DAT_IO_CTRL is shown in Figure 3-87 and described in Table 3-89.

This register controls the MLBP DAT pins IO characteristics.

**Figure 3-87. MLBP_DAT_IO_CTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd3 | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Rsvd3 | | NC_IN | | | | | |
| R-0h | | R/W-0h | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Rsvd2 | | PC_IN | | | | | |
| R-0h | | R/W-0h | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | | PWRDNRX | PWRDNTX | EN_EXT_RES | ENLVCMOS | ENN | ENP |
| R-0h | | R/W-1h | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-89. MLBP_DAT_IO_CTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | Rsvd3 | R | 0h | Reserved-read returns 0 |
| 21-16 | NC_IN | R/W | 0h | MLB LVDS Output Drive Trim value. If register defaults to state of 0x0, then it should be set to the recommended state of 0x1A. If the register defaults to non-zero, that implies it was set to optimal value during manufacturing. |
| 15-14 | Rsvd2 | R | 0h | Reserved-read returns 0 |
| 13-8 | PC_IN | R/W | 0h | MLB LVDS Output Drive Trim value. If register defaults to state of 0x0, then it should be set to the recommended state of 0x24. If the register defaults to non-zero, that implies it was set to optimal value during manufacturing. |
| 7-6 | Rsvd | R | 0h | Reserved-read returns 0 |
| 5 | PWRDNRX | R/W | 1h | Power down control for MLB LVDS receiver. A 1 written to this register will power down the receiver. A 0 written to this register will power on the receiver. By default, the receiver is in power down mode. |
| 4 | PWRDNTX | R/W | 1h | Power down control for MLB LVDS transmitter. A 1 written to this register will power down the transmitter. A 0 written to this register will power on the transmitter. By default, the transmitter is in power down mode. |
| 3 | EN_EXT_RES | R/W | 1h | Disables the internal resistors required for proper MLB LVDS operation. This register should always be programmed to 0x0 |
| 2 | ENLVCMOS | R/W | 0h | This registers is used to disable the MLB LVDS mode and program the pins as GPIO. Each pad acts as 50ohm LVCMOS TX and LVCMOS RX when this bit is set to 1. Each pad is MLB LVDS when this bit is set to zero. |
| 1 | ENN | R/W | 0h | Receiver enable for LVCMOS GPIO mode. Programming this register to 1 enables the MLBP_DATN pad to be a GPIO input. Programming this register to 0 disables the input and MLBP_DATN pad is only an output. |
| 0 | ENP | R | 0h | Receiver enable for LVCMOS GPIO mode. Programming this register to 1 enables the MLBP_DATP pad to be a GPIO input. Programming this register to 0 disables the input and MLBP_DATP pad is only an output. |

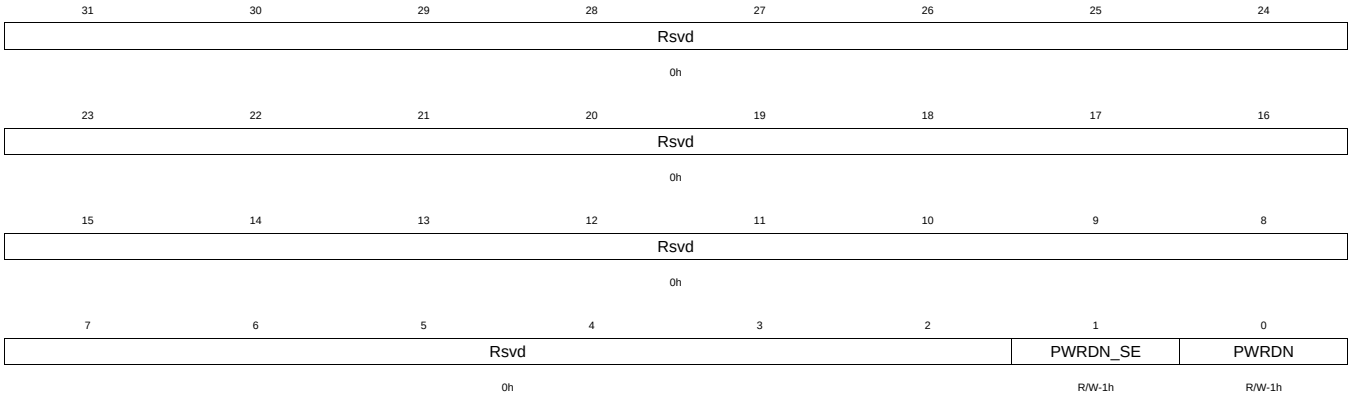

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.87 SERDES_REFCLK_CTL Register (offset = E24h) [reset = 3h]

SERDES_REFCLK_CTL is shown in Figure 3-88 and described in Table 3-90.

This register controls the refclk for the SerDes block s IO powerdown.

**Figure 3-88. SERDES_REFCLK_CTL Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-90. SERDES_REFCLK_CTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Rsvd | | 0h | Reserved-read returns 0 |
| 1 | PWRDN_SE | R/W | 1h | Powerdown both refclkp/n single ended receiver |
| 0 | PWRDN | R/W | 1h | Powerdown both refclkp/n receiver |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.88 DSP_INTMUX_15_18 Register (offset = F00h) [reset = 0h]

DSP_INTMUX_15_18 is shown in Figure 3-89 and described in Table 3-91.

This register controls interrupt mux assignment for Interrupt 15 to 18

**Figure 3-89. DSP_INTMUX_15_18 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_18 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_17 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_16 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_15 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-91. DSP_INTMUX_15_18 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_18 | R/W | 0h | Interrupt mux for GEM s interrupt number 18 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_17 | R/W | 0h | Interrupt mux for GEM s interrupt number 17 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_16 | R/W | 0h | Interrupt mux for GEM s interrupt number 16 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_15 | R/W | 0h | Interrupt mux for GEM s interrupt number 15 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.89 DSP_INTMUX_19_22 Register (offset = F04h) [reset = 0h]

DSP_INTMUX_19_22 is shown in Figure 3-90 and described in Table 3-92.

This register controls interrupt mux assignment for Interrupt 19 to 22

**Figure 3-90. DSP_INTMUX_19_22 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_22 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_21 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_20 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_19 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-92. DSP_INTMUX_19_22 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_22 | R/W | 0h | Interrupt mux for GEM s interrupt number 22 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_21 | R/W | 0h | Interrupt mux for GEM s interrupt number 21 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_20 | R/W | 0h | Interrupt mux for GEM s interrupt number 20 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_19 | R/W | 0h | Interrupt mux for GEM s interrupt number 19 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.90 DSP_INTMUX_23_26 Register (offset = F08h) [reset = 0h]

DSP_INTMUX_23_26 is shown in Figure 3-91 and described in Table 3-93.

This register controls interrupt mux assignment for Interrupt 23 to 26

**Figure 3-91. DSP_INTMUX_23_26 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_26 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_25 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_24 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_23 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-93. DSP_INTMUX_23_26 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_26 | R/W | 0h | Interrupt mux for GEM s interrupt number 26 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_25 | R/W | 0h | Interrupt mux for GEM s interrupt number 25 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_24 | R/W | 0h | Interrupt mux for GEM s interrupt number 24 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_23 | R/W | 0h | Interrupt mux for GEM s interrupt number 23 |

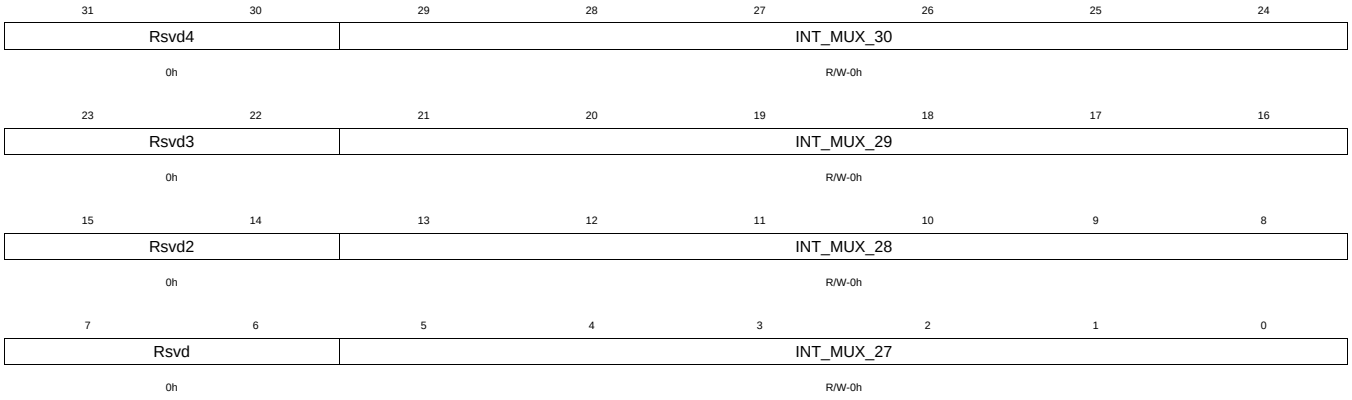

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.91 DSP_INTMUX_27_30 Register (offset = F0Ch) [reset = 0h]

DSP_INTMUX_27_30 is shown in Figure 3-92 and described in Table 3-94.

This register controls interrupt mux assignment for Interrupt 27 to 30

**Figure 3-92. DSP_INTMUX_27_30 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-94. DSP_INTMUX_27_30 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_30 | R/W | 0h | Interrupt mux for GEM s interrupt number 30 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_29 | R/W | 0h | Interrupt mux for GEM s interrupt number 29 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_28 | R/W | 0h | Interrupt mux for GEM s interrupt number 28 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_27 | R/W | 0h | Interrupt mux for GEM s interrupt number 27 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.92 DSP_INTMUX_31_34 Register (offset = F10h) [reset = 0h]

DSP_INTMUX_31_34 is shown in Figure 3-93 and described in Table 3-95.

This register controls interrupt mux assignment for Interrupt 31 to 34

**Figure 3-93. DSP_INTMUX_31_34 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_34 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_33 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_32 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_31 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-95. DSP_INTMUX_31_34 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_34 | R/W | 0h | Interrupt mux for GEM s interrupt number 34 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_33 | R/W | 0h | Interrupt mux for GEM s interrupt number 33 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_32 | R/W | 0h | Interrupt mux for GEM s interrupt number 32 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_31 | R/W | 0h | Interrupt mux for GEM s interrupt number 31 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.93 DSP_INTMUX_35_38 Register (offset = F14h) [reset = 0h]

DSP_INTMUX_35_38 is shown in Figure 3-94 and described in Table 3-96.

This register controls interrupt mux assignment for Interrupt 35 to 38

**Figure 3-94. DSP_INTMUX_35_38 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_38 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_37 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_36 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_35 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-96. DSP_INTMUX_35_38 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_38 | R/W | 0h | Interrupt mux for GEM s interrupt number 38 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_37 | R/W | 0h | Interrupt mux for GEM s interrupt number 37 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_36 | R/W | 0h | Interrupt mux for GEM s interrupt number 36 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_35 | R/W | 0h | Interrupt mux for GEM s interrupt number 35 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.94 DSP_INTMUX_39_42 Register (offset = F18h) [reset = 0h]

DSP_INTMUX_39_42 is shown in Figure 3-95 and described in Table 3-97.

This register controls interrupt mux assignment for Interrupt 39 to 42

**Figure 3-95. DSP_INTMUX_39_42 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_42 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_41 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_40 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_39 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-97. DSP_INTMUX_39_42 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_42 | R/W | 0h | Interrupt mux for GEM s interrupt number 42 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_41 | R/W | 0h | Interrupt mux for GEM s interrupt number 41 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_40 | R/W | 0h | Interrupt mux for GEM s interrupt number 40 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_39 | R/W | 0h | Interrupt mux for GEM s interrupt number 39 |

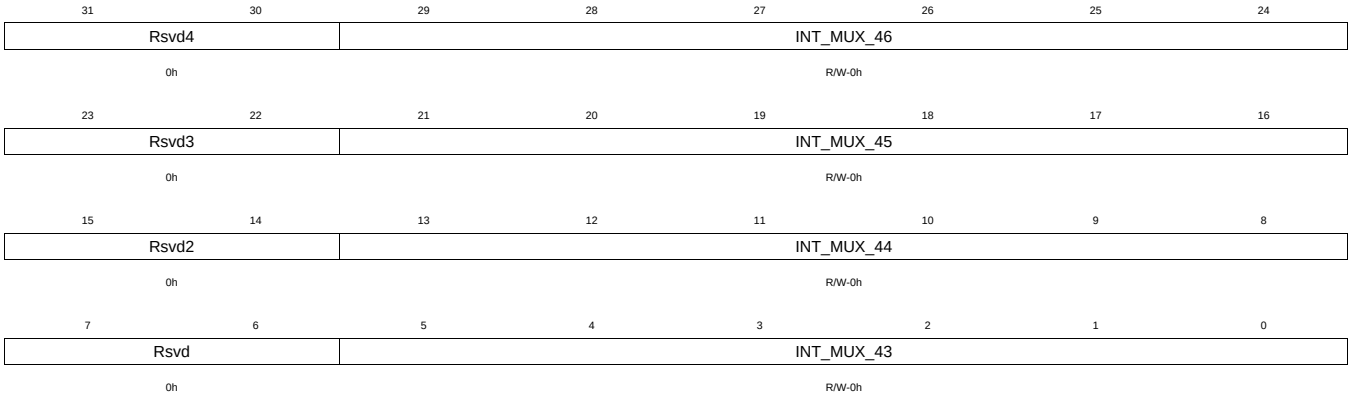

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.95 DSP_INTMUX_43_46 Register (offset = F1Ch) [reset = 0h]

DSP_INTMUX_43_46 is shown in Figure 3-96 and described in Table 3-98.

This register controls interrupt mux assignment for Interrupt 43 to 46

**Figure 3-96. DSP_INTMUX_43_46 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-98. DSP_INTMUX_43_46 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_46 | R/W | 0h | Interrupt mux for GEM s interrupt number 46 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_45 | R/W | 0h | Interrupt mux for GEM s interrupt number 45 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_44 | R/W | 0h | Interrupt mux for GEM s interrupt number 44 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_43 | R/W | 0h | Interrupt mux for GEM s interrupt number 43 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.96 DSP_INTMUX_47_50 Register (offset = F20h) [reset = 0h]

DSP_INTMUX_47_50 is shown in Figure 3-97 and described in Table 3-99.

This register controls interrupt mux assignment for Interrupt 47 to 50

**Figure 3-97. DSP_INTMUX_47_50 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_50 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_49 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_48 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_47 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-99. DSP_INTMUX_47_50 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_50 | R/W | 0h | Interrupt mux for GEM s interrupt number 50 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_49 | R/W | 0h | Interrupt mux for GEM s interrupt number 49 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_48 | R/W | 0h | Interrupt mux for GEM s interrupt number 48 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_47 | R/W | 0h | Interrupt mux for GEM s interrupt number 47 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.97 DSP_INTMUX_51_54 Register (offset = F24h) [reset = 0h]

DSP_INTMUX_51_54 is shown in Figure 3-98 and described in Table 3-100.

This register controls interrupt mux assignment for Interrupt 51 to 54

**Figure 3-98. DSP_INTMUX_51_54 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_54 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_53 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_52 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_51 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-100. DSP_INTMUX_51_54 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_54 | R/W | 0h | Interrupt mux for GEM s interrupt number 54 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_53 | R/W | 0h | Interrupt mux for GEM s interrupt number 53 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_52 | R/W | 0h | Interrupt mux for GEM s interrupt number 52 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_51 | R/W | 0h | Interrupt mux for GEM s interrupt number 51 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.98 DSP_INTMUX_55_58 Register (offset = F28h) [reset = 0h]

DSP_INTMUX_55_58 is shown in Figure 3-99 and described in Table 3-101.

This register controls interrupt mux assignment for Interrupt 55 to 58

**Figure 3-99. DSP_INTMUX_55_58 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_58 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_57 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_56 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_55 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-101. DSP_INTMUX_55_58 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_58 | R/W | 0h | Interrupt mux for GEM s interrupt number 58 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_57 | R/W | 0h | Interrupt mux for GEM s interrupt number 57 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_56 | R/W | 0h | Interrupt mux for GEM s interrupt number 56 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_55 | R/W | 0h | Interrupt mux for GEM s interrupt number 55 |

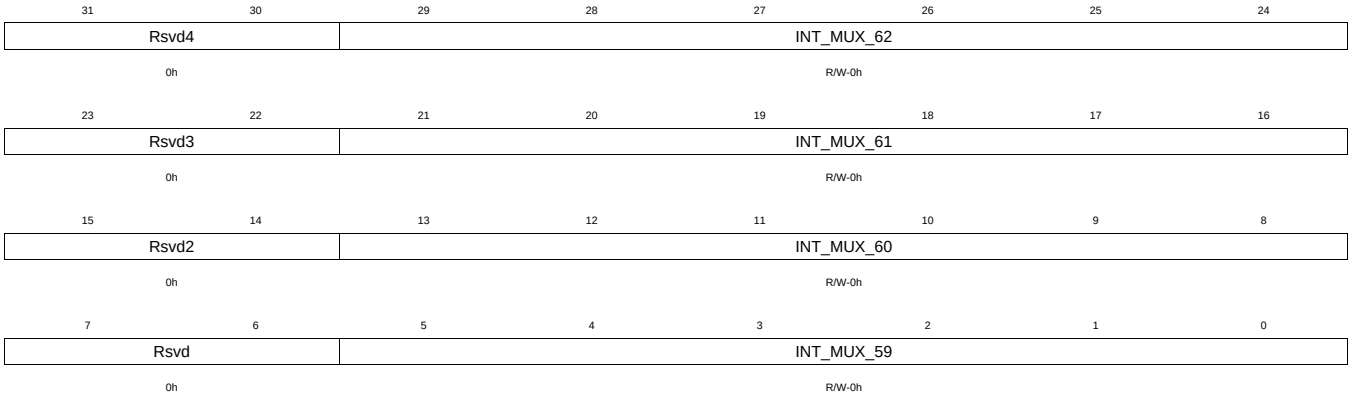

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.99 DSP_INTMUX_59_62 Register (offset = F2Ch) [reset = 0h]

DSP_INTMUX_59_62 is shown in Figure 3-100 and described in Table 3-102.

This register controls interrupt mux assignment for Interrupt 59 to 62

**Figure 3-100. DSP_INTMUX_59_62 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-102. DSP_INTMUX_59_62 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_62 | R/W | 0h | Interrupt mux for GEM s interrupt number 62 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_61 | R/W | 0h | Interrupt mux for GEM s interrupt number 61 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_60 | R/W | 0h | Interrupt mux for GEM s interrupt number 60 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_59 | R/W | 0h | Interrupt mux for GEM s interrupt number 59 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.100 DSP_INTMUX_63_66 Register (offset = F30h) [reset = 0h]

DSP_INTMUX_63_66 is shown in Figure 3-101 and described in Table 3-103.

This register controls interrupt mux assignment for Interrupt 63 to 66

**Figure 3-101. DSP_INTMUX_63_66 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_66 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_65 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_64 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_63 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-103. DSP_INTMUX_63_66 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_66 | R/W | 0h | Interrupt mux for GEM s interrupt number 66 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_65 | R/W | 0h | Interrupt mux for GEM s interrupt number 65 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_64 | R/W | 0h | Interrupt mux for GEM s interrupt number 64 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_63 | R/W | 0h | Interrupt mux for GEM s interrupt number 63 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.101 DSP_INTMUX_67_70 Register (offset = F34h) [reset = 0h]

DSP_INTMUX_67_70 is shown in Figure 3-102 and described in Table 3-104.

This register controls interrupt mux assignment for Interrupt 67 to 70

**Figure 3-102. DSP_INTMUX_67_70 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_70 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_69 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_68 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_67 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-104. DSP_INTMUX_67_70 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_70 | R/W | 0h | Interrupt mux for GEM s interrupt number 70 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_69 | R/W | 0h | Interrupt mux for GEM s interrupt number 69 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_68 | R/W | 0h | Interrupt mux for GEM s interrupt number 68 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_67 | R/W | 0h | Interrupt mux for GEM s interrupt number 67 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.102 DSP_INTMUX_71_74 Register (offset = F38h) [reset = 0h]

DSP_INTMUX_71_74 is shown in Figure 3-103 and described in Table 3-105.

This register controls interrupt mux assignment for Interrupt 71 to 74

**Figure 3-103. DSP_INTMUX_71_74 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_74 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_73 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_72 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_71 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-105. DSP_INTMUX_71_74 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_74 | R/W | 0h | Interrupt mux for GEM s interrupt number 74 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_73 | R/W | 0h | Interrupt mux for GEM s interrupt number 73 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_72 | R/W | 0h | Interrupt mux for GEM s interrupt number 72 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_71 | R/W | 0h | Interrupt mux for GEM s interrupt number 71 |

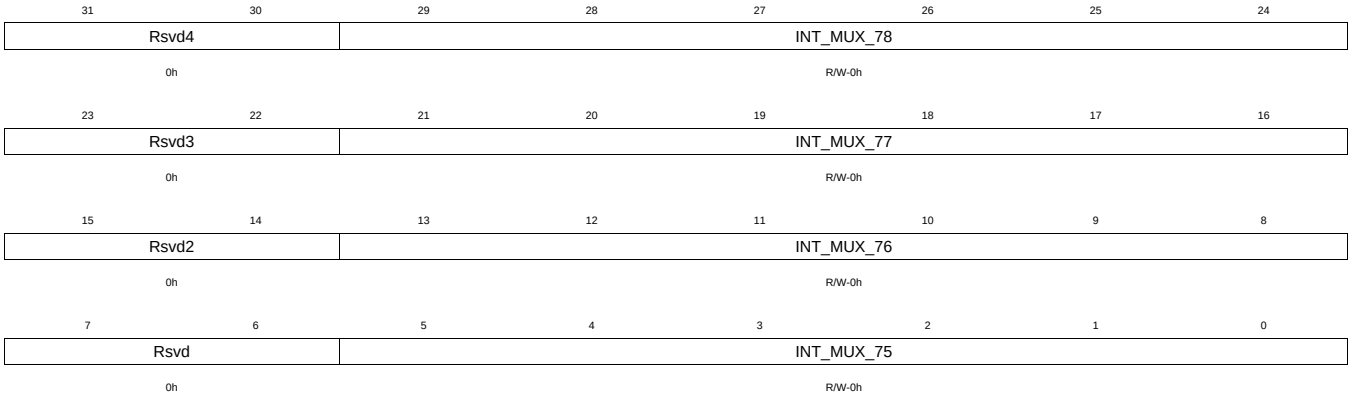

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.103 DSP_INTMUX_75_78 Register (offset = F3Ch) [reset = 0h]

DSP_INTMUX_75_78 is shown in Figure 3-104 and described in Table 3-106.

This register controls interrupt mux assignment for Interrupt 75 to 78

**Figure 3-104. DSP_INTMUX_75_78 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-106. DSP_INTMUX_75_78 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_78 | R/W | 0h | Interrupt mux for GEM s interrupt number 78 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_77 | R/W | 0h | Interrupt mux for GEM s interrupt number 77 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_76 | R/W | 0h | Interrupt mux for GEM s interrupt number 76 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_75 | R/W | 0h | Interrupt mux for GEM s interrupt number 75 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.104 DSP_INTMUX_79_82 Register (offset = F40h) [reset = 0h]

DSP_INTMUX_79_82 is shown in Figure 3-105 and described in Table 3-107.

This register controls interrupt mux assignment for Interrupt 79 to 82

**Figure 3-105. DSP_INTMUX_79_82 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_82 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_81 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_80 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_79 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-107. DSP_INTMUX_79_82 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_82 | R/W | 0h | Interrupt mux for GEM s interrupt number 82 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_81 | R/W | 0h | Interrupt mux for GEM s interrupt number 81 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_80 | R/W | 0h | Interrupt mux for GEM s interrupt number 80 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_79 | R/W | 0h | Interrupt mux for GEM s interrupt number 79 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.105 DSP_INTMUX_83_86 Register (offset = F44h) [reset = 0h]

DSP_INTMUX_83_86 is shown in Figure 3-106 and described in Table 3-108.

This register controls interrupt mux assignment for Interrupt 83 to 86

**Figure 3-106. DSP_INTMUX_83_86 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_86 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_85 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_84 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_83 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-108. DSP_INTMUX_83_86 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_86 | R/W | 0h | Interrupt mux for GEM s interrupt number 86 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_85 | R/W | 0h | Interrupt mux for GEM s interrupt number 85 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_84 | R/W | 0h | Interrupt mux for GEM s interrupt number 84 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_83 | R/W | 0h | Interrupt mux for GEM s interrupt number 83 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.106 DSP_INTMUX_87_90 Register (offset = F48h) [reset = 0h]

DSP_INTMUX_87_90 is shown in Figure 3-107 and described in Table 3-109.

This register controls interrupt mux assignment for Interrupt 87 to 90

**Figure 3-107. DSP_INTMUX_87_90 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | INT_MUX_90 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | INT_MUX_89 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | INT_MUX_88 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | INT_MUX_87 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-109. DSP_INTMUX_87_90 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_90 | R/W | 0h | Interrupt mux for GEM s interrupt number 90 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_89 | R/W | 0h | Interrupt mux for GEM s interrupt number 89 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_88 | R/W | 0h | Interrupt mux for GEM s interrupt number 88 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_87 | R/W | 0h | Interrupt mux for GEM s interrupt number 87 |

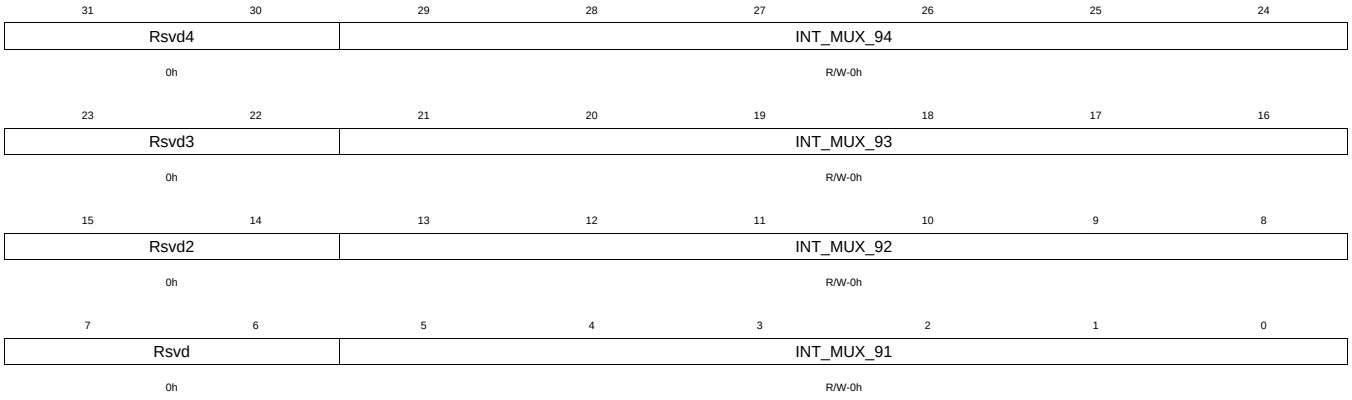

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.107 DSP_INTMUX_91_94 Register (offset = F4Ch) [reset = 0h]

DSP_INTMUX_91_94 is shown in Figure 3-108 and described in Table 3-110.

This register controls interrupt mux assignment for Interrupt 91 to 94

**Figure 3-108. DSP_INTMUX_91_94 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-110. DSP_INTMUX_91_94 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | INT_MUX_94 | R/W | 0h | Interrupt mux for GEM s interrupt number 94 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | INT_MUX_93 | R/W | 0h | Interrupt mux for GEM s interrupt number 93 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | INT_MUX_92 | R/W | 0h | Interrupt mux for GEM s interrupt number 92 |
| 7-6 | Rsvd | | 0h | Reserved |
| 5-0 | INT_MUX_91 | R/W | 0h | Interrupt mux for GEM s interrupt number 91 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.108 DSP_INTMUX_95 Register (offset = F50h) [reset = 0h]

DSP_INTMUX_95 is shown in Figure 3-109 and described in Table 3-111.

This register controls interrupt mux assignment for Interrupt 95

**Figure 3-109. DSP_INTMUX_95 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd | | | | | | | |
| 0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Rsvd | | | | | | | |
| 0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Rsvd | | | | | | | |
| 0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | | INT_MUX_95 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-111. DSP_INTMUX_95 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | INT_MUX_95 | R/W | 0h | Interrupt mux for GEM s interrupt number 95 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.109 Media_Controller_INTMUX_0_3 Register (offset = F54h) [reset = 0h]

Media_Controller_INTMUX_0_3 is shown in Figure 3-110 and described in Table 3-112.

This register controls Media Controller interrupt assignment for interrupts through 0 to 3

**Figure 3-110. Media_Controller_INTMUX_0_3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | INT_MUX_3 | | | | | | |
| 0h | R/W-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | INT_MUX_2 | | | | | | |
| 0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | INT_MUX_1 | | | | | | |
| 0h | R/W-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | INT_MUX_0 | | | | | | |
| 0h | R/W-0h | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-112. Media_Controller_INTMUX_0_3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_3 | R/W | 0h | Interrupt mux 3 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_2 | R/W | 0h | Interrupt mux 2 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_1 | R/W | 0h | Interrupt mux 1 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_0 | R/W | 0h | Interrupt mux 0 |

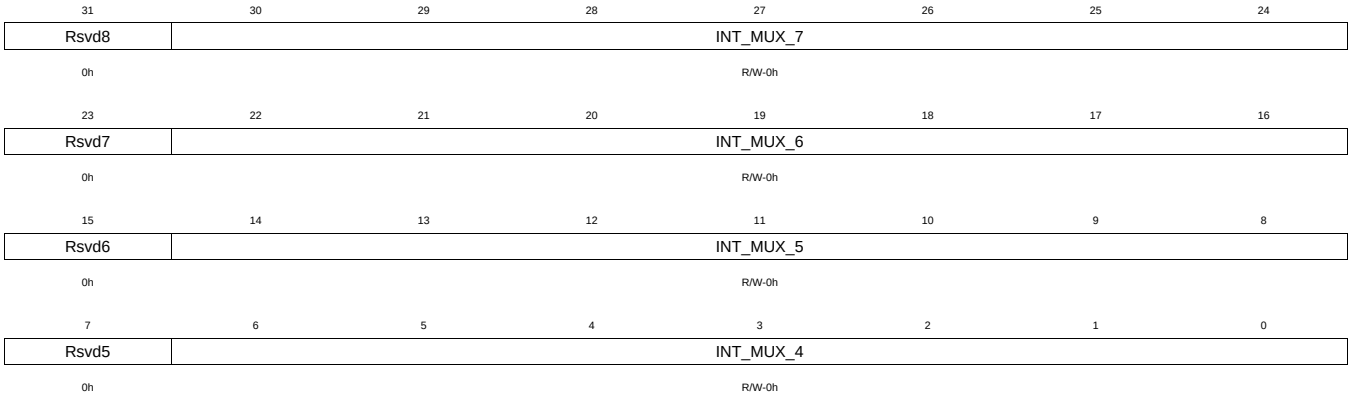

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.110 Media_Controller_INTMUX_4_7 Register (offset = F58h) [reset = 0h]

Media_Controller_INTMUX_4_7 is shown in Figure 3-111 and described in Table 3-113.

This register controls Media Controller interrupt assignment for interrupts through 4 to 7

**Figure 3-111. Media_Controller_INTMUX_4_7 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-113. Media_Controller_INTMUX_4_7 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd8 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_7 | R/W | 0h | Interrupt mux 7 |
| 23 | Rsvd7 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_6 | R/W | 0h | Interrupt mux 6 |
| 15 | Rsvd6 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_5 | R/W | 0h | Interrupt mux 5 |
| 7 | Rsvd5 | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_4 | R/W | 0h | Interrupt mux 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.111 Media_Controller_INTMUX_8_11 Register (offset = F5Ch) [reset = 0h]

Media_Controller_INTMUX_8_11 is shown in Figure 3-112 and described in Table 3-114.

This register controls Media Controller interrupt assignment for interrupts through 8 to 11

**Figure 3-112. Media_Controller_INTMUX_8_11 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | INT_MUX_11 | | | | | | |
| 0h | R/W-0h | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Rsvd3 | INT_MUX_10 | | | | | | |
| 0h | R/W-0h | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Rsvd2 | INT_MUX_9 | | | | | | |
| 0h | R/W-0h | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | INT_MUX_8 | | | | | | |
| 0h | R/W-0h | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-114. Media_Controller_INTMUX_8_11 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_11 | R/W | 0h | Interrupt mux 11 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_10 | R/W | 0h | Interrupt mux 10 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_9 | R/W | 0h | Interrupt mux 9 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_8 | R/W | 0h | Interrupt mux 8 |

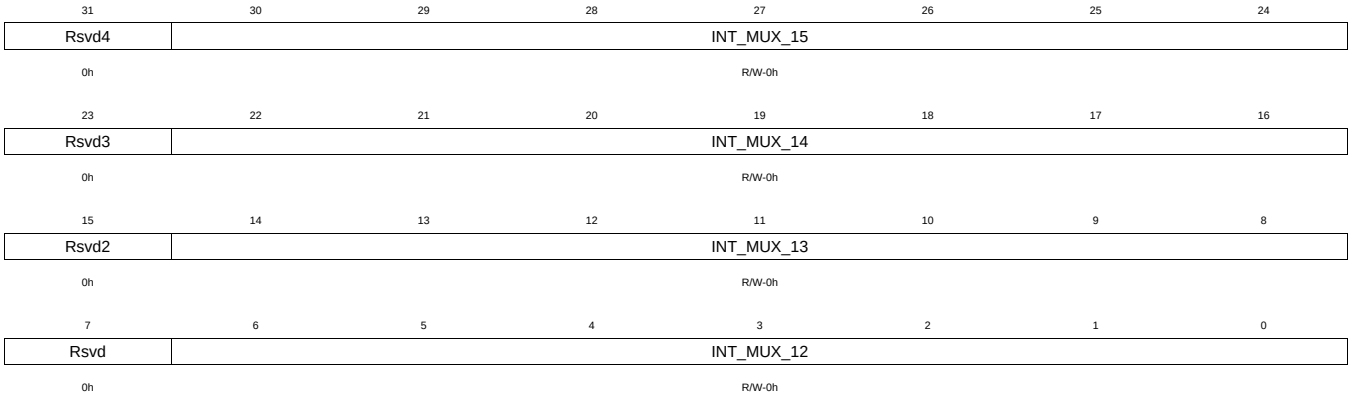

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.112 Media_Controller_INTMUX_12_15 Register (offset = F60h) [reset = 0h]

Media_Controller_INTMUX_12_15 is shown in Figure 3-113 and described in Table 3-115.

This register controls Media Controller interrupt assignment for interrupts through 12 to 15

**Figure 3-113. Media_Controller_INTMUX_12_15 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-115. Media_Controller_INTMUX_12_15 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_15 | R/W | 0h | Interrupt mux 15 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_14 | R/W | 0h | Interrupt mux 14 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_13 | R/W | 0h | Interrupt mux 13 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_12 | R/W | 0h | Interrupt mux 12 |

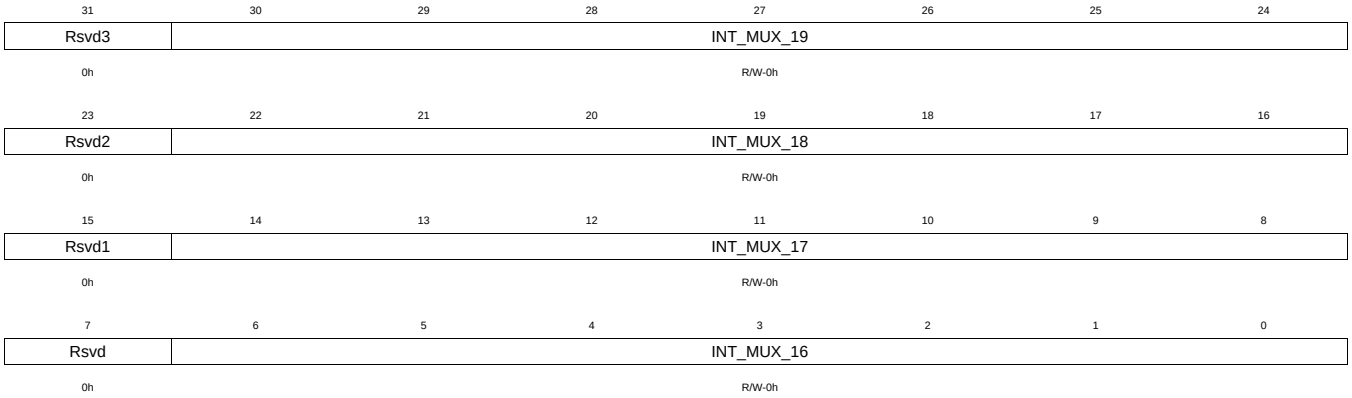

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.113 Media_Controller_INTMUX_16_19 Register (offset = F64h) [reset = 0h]

Media_Controller_INTMUX_16_19 is shown in Figure 3-114 and described in Table 3-116.

This register controls Media Controller interrupt assignment for interrupts through 16 to 19

**Figure 3-114. Media_Controller_INTMUX_16_19 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-116. Media_Controller_INTMUX_16_19 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd3 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_19 | R/W | 0h | Interrupt mux 19 |
| 23 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_18 | R/W | 0h | Interrupt mux 18 |
| 15 | Rsvd1 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_17 | R/W | 0h | Interrupt mux 17 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_16 | R/W | 0h | Interrupt mux 16 |

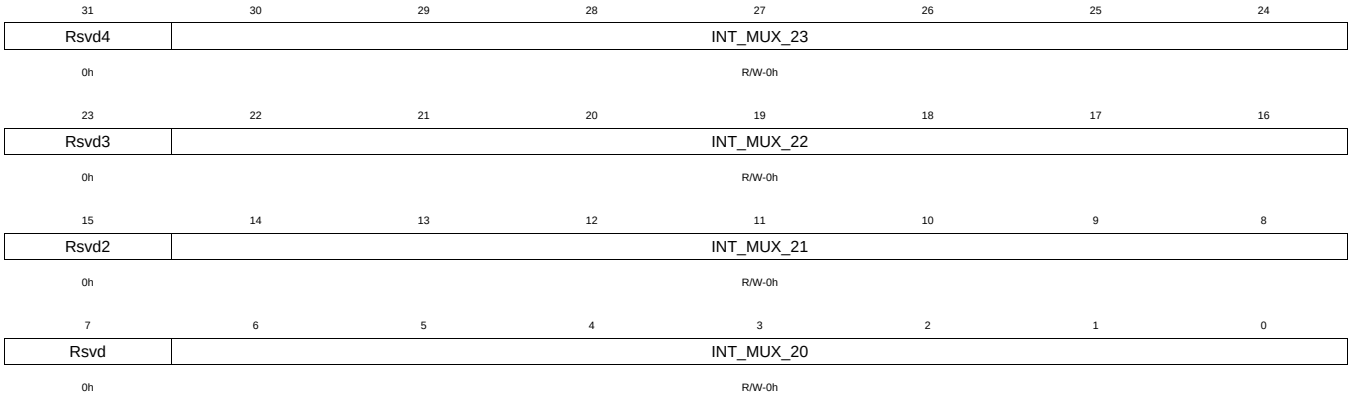

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.114 Media_Controller_INTMUX_20_23 Register (offset = F68h) [reset = 0h]

Media_Controller_INTMUX_20_23 is shown in Figure 3-115 and described in Table 3-117.

This register controls Media Controller interrupt assignment for interrupts through 20 to 23

**Figure 3-115. Media_Controller_INTMUX_20_23 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-117. Media_Controller_INTMUX_20_23 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_23 | R/W | 0h | Interrupt mux 23 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_22 | R/W | 0h | Interrupt mux 22 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_21 | R/W | 0h | Interrupt mux 21 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_20 | R/W | 0h | Interrupt mux 20 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.115 Media_Controller_INTMUX_24_27 Register (offset = F6Ch) [reset = 0h]

Media_Controller_INTMUX_24_27 is shown in Figure 3-116 and described in Table 3-118.

This register controls Media Controller interrupt assignment for interrupts through 24 to 27

**Figure 3-116. Media_Controller_INTMUX_24_27 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | INT_MUX_27 | | | | | | |
| 0h | R/W-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | INT_MUX_26 | | | | | | |
| 0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | INT_MUX_25 | | | | | | |
| 0h | R/W-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | INT_MUX_24 | | | | | | |
| 0h | R/W-0h | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-118. Media_Controller_INTMUX_24_27 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_27 | R/W | 0h | Interrupt mux 27 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_26 | R/W | 0h | Interrupt mux 26 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_25 | R/W | 0h | Interrupt mux 25 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_24 | R/W | 0h | Interrupt mux 24 |

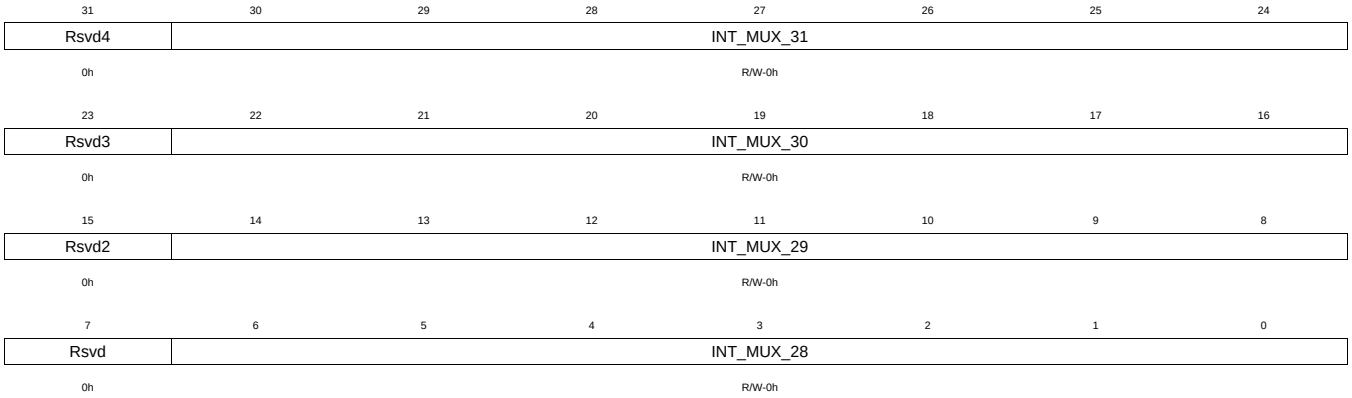

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.116 Media_Controller_INTMUX_28_31 Register (offset = F70h) [reset = 0h]

Media_Controller_INTMUX_28_31 is shown in Figure 3-117 and described in Table 3-119.

This register controls Media Controller interrupt assignment for interrupts through 28 to 31

**Figure 3-117. Media_Controller_INTMUX_28_31 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-119. Media_Controller_INTMUX_28_31 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_31 | R/W | 0h | Interrupt mux 31 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_30 | R/W | 0h | Interrupt mux 30 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_29 | R/W | 0h | Interrupt mux 29 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_28 | R/W | 0h | Interrupt mux 28 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.117 Media_Controller_INTMUX_32_35 Register (offset = F74h) [reset = 0h]

Media_Controller_INTMUX_32_35 is shown in Figure 3-118 and described in Table 3-120.

This register controls Media Controller interrupt assignment for interrupts through 32 to 35

**Figure 3-118. Media_Controller_INTMUX_32_35 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | INT_MUX_35 | | | | | | |
| 0h | R/W-0h | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Rsvd3 | INT_MUX_34 | | | | | | |
| 0h | R/W-0h | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Rsvd2 | INT_MUX_33 | | | | | | |
| 0h | R/W-0h | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | INT_MUX_32 | | | | | | |
| 0h | R/W-0h | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-120. Media_Controller_INTMUX_32_35 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_35 | R/W | 0h | Interrupt mux 35 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_34 | R/W | 0h | Interrupt mux 34 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_33 | R/W | 0h | Interrupt mux 33 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_32 | R/W | 0h | Interrupt mux 32 |

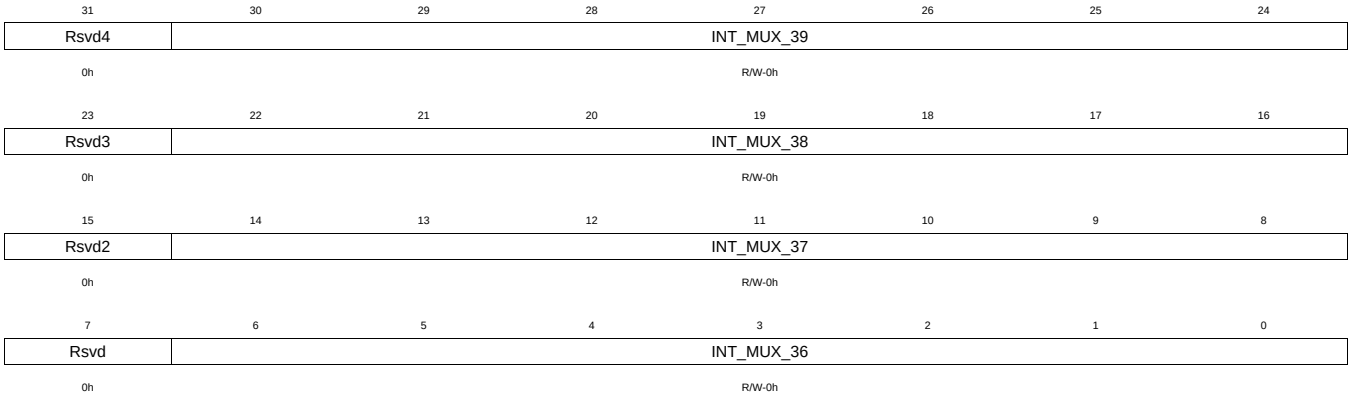

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.118 Media_Controller_INTMUX_36_39 Register (offset = F78h) [reset = 0h]

Media_Controller_INTMUX_36_39 is shown in Figure 3-119 and described in Table 3-121.

This register controls Media Controller interrupt assignment for interrupts through 36 to 39

**Figure 3-119. Media_Controller_INTMUX_36_39 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-121. Media_Controller_INTMUX_36_39 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_39 | R/W | 0h | Interrupt mux 39 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_38 | R/W | 0h | Interrupt mux 38 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_37 | R/W | 0h | Interrupt mux 37 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_36 | R/W | 0h | Interrupt mux 36 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.119 Media_Controller_INTMUX_40_43 Register (offset = F7Ch) [reset = 0h]

Media_Controller_INTMUX_40_43 is shown in Figure 3-120 and described in Table 3-122.

This register controls Media Controller interrupt assignment for interrupts through 40 to 43

**Figure 3-120. Media_Controller_INTMUX_40_43 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | INT_MUX_43 | | | | | | |
| 0h | R/W-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | INT_MUX_42 | | | | | | |
| 0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | INT_MUX_41 | | | | | | |
| 0h | R/W-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | INT_MUX_40 | | | | | | |
| 0h | R/W-0h | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-122. Media_Controller_INTMUX_40_43 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_43 | R/W | 0h | Interrupt mux 43 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_42 | R/W | 0h | Interrupt mux 42 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_41 | R/W | 0h | Interrupt mux 41 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_40 | R/W | 0h | Interrupt mux 40 |

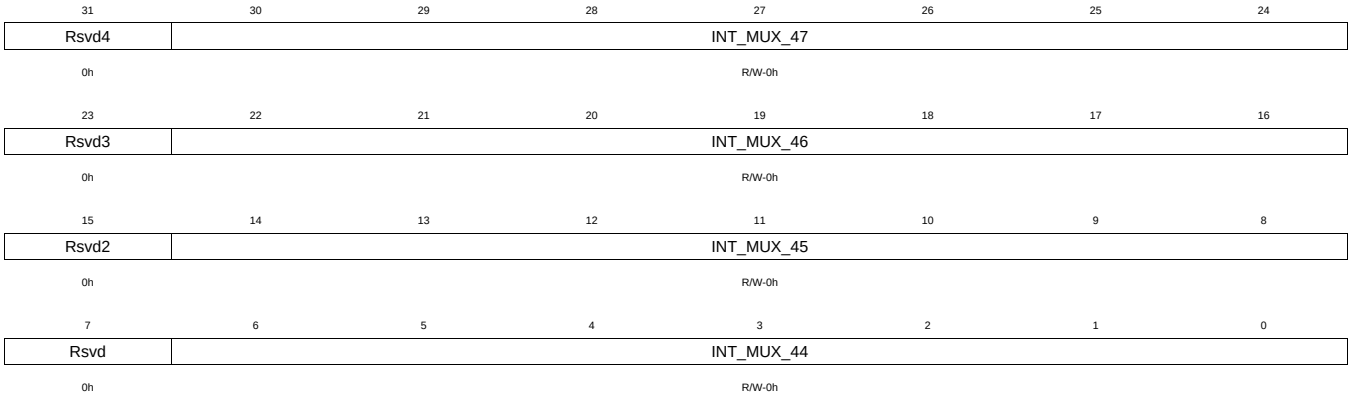

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.120 Media_Controller_INTMUX_44_47 Register (offset = F80h) [reset = 0h]

Media_Controller_INTMUX_44_47 is shown in Figure 3-121 and described in Table 3-123.

This register controls Media Controller interrupt assignment for interrupts through 44 to 47

**Figure 3-121. Media_Controller_INTMUX_44_47 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-123. Media_Controller_INTMUX_44_47 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_47 | R/W | 0h | Interrupt mux 47 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_46 | R/W | 0h | Interrupt mux 46 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_45 | R/W | 0h | Interrupt mux 45 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_44 | R/W | 0h | Interrupt mux 44 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.121 Media_Controller_INTMUX_48_51 Register (offset = F84h) [reset = 0h]

Media_Controller_INTMUX_48_51 is shown in Figure 3-122 and described in Table 3-124.

This register controls Media Controller interrupt assignment for interrupts through 48 to 51

**Figure 3-122. Media_Controller_INTMUX_48_51 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | INT_MUX_51 | | | | | | |
| 0h | R/W-0h | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Rsvd3 | INT_MUX_50 | | | | | | |
| 0h | R/W-0h | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Rsvd2 | INT_MUX_49 | | | | | | |
| 0h | R/W-0h | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | INT_MUX_48 | | | | | | |
| 0h | R/W-0h | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-124. Media_Controller_INTMUX_48_51 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_51 | R/W | 0h | Interrupt mux 51 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_50 | R/W | 0h | Interrupt mux 50 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_49 | R/W | 0h | Interrupt mux 49 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_48 | R/W | 0h | Interrupt mux 48 |

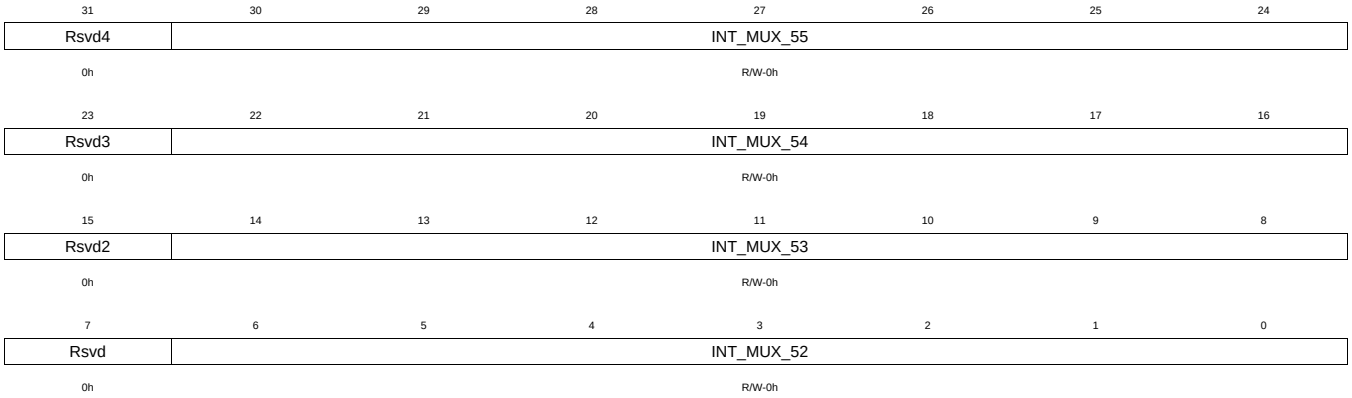

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.122 Media_Controller_INTMUX_52_55 Register (offset = F88h) [reset = 0h]

Media_Controller_INTMUX_52_55 is shown in Figure 3-123 and described in Table 3-125.

This register controls Media Controller interrupt assignment for interrupts through 52 to 55

**Figure 3-123. Media_Controller_INTMUX_52_55 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-125. Media_Controller_INTMUX_52_55 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Rsvd4 | | 0h | Reserved -Read returns 0 |
| 30-24 | INT_MUX_55 | R/W | 0h | Interrupt mux 55 |
| 23 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 22-16 | INT_MUX_54 | R/W | 0h | Interrupt mux 54 |
| 15 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 14-8 | INT_MUX_53 | R/W | 0h | Interrupt mux 53 |
| 7 | Rsvd | | 0h | Reserved-Read returns 0 |
| 6-0 | INT_MUX_52 | R/W | 0h | Interrupt mux 52 |

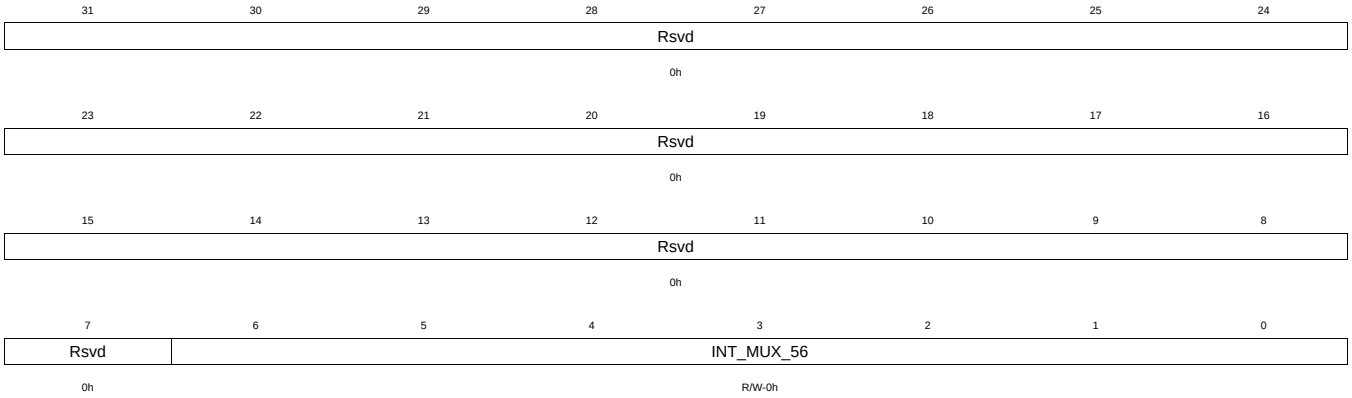

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.123 Media_Controller_INTMUX_56 Register (offset = F8Ch) [reset = 0h]

Media_Controller_INTMUX_56 is shown in Figure 3-124 and described in Table 3-126.

This register controls Media Controller interrupt assignment for interrupt number 56.

**Figure 3-124. Media_Controller_INTMUX_56 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-126. Media_Controller_INTMUX_56 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | Rsvd | | 0h | Reserved- Read returns 0 |
| 6-0 | INT_MUX_56 | R/W | 0h | Interrupt mux 56 |

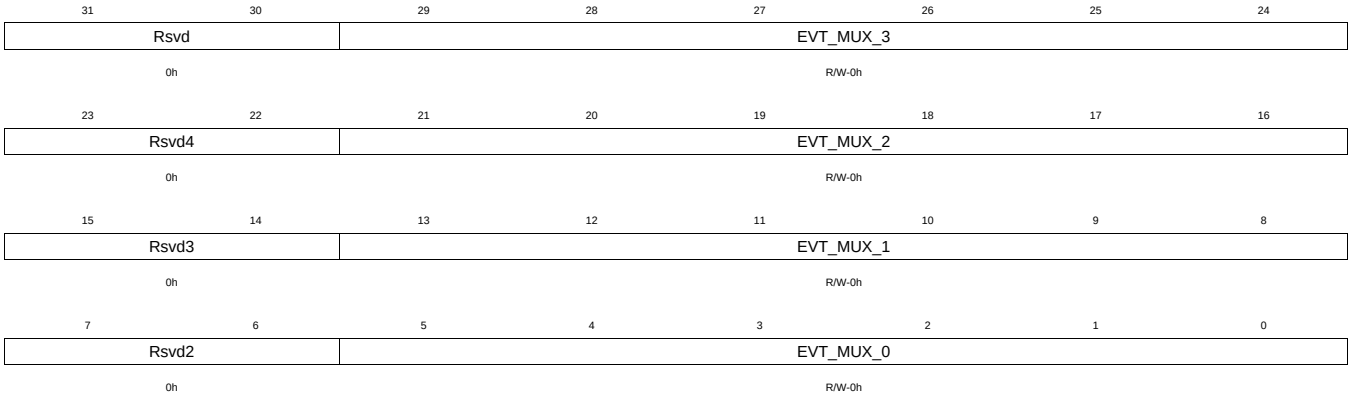

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.124 EDMA3CC_EVTMUX_0_3 Register (offset = F90h) [reset = 0h]

EDMA3CC_EVTMUX_0_3 is shown in Figure 3-125 and described in Table 3-127.

This register controls EDMA3CC Event mux assignment for event no 0 to 3

**Figure 3-125. EDMA3CC_EVTMUX_0_3 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-127. EDMA3CC_EVTMUX_0_3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_3 | R/W | 0h | Event mux for EDMA3CC event number 3 |
| 23-22 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_2 | R/W | 0h | Event mux for EDMA3CC event number 2 |
| 15-14 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_1 | R/W | 0h | Event mux for EDMA3CC event number 1 |
| 7-6 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_0 | R/W | 0h | Event mux for EDMA3CC event number 0 |

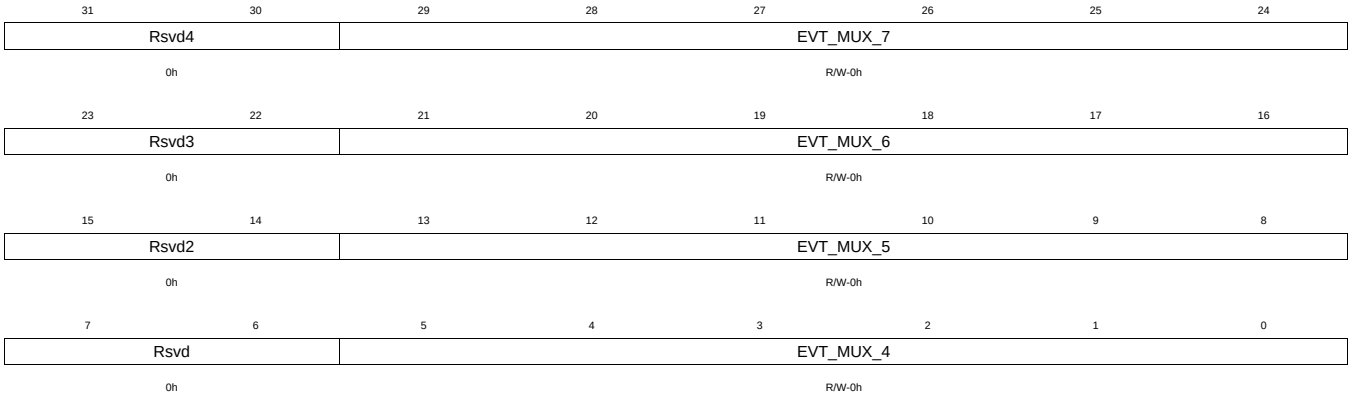

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.125 EDMA3CC_EVTMUX_4_7 Register (offset = F94h) [reset = 0h]

EDMA3CC_EVTMUX_4_7 is shown in Figure 3-126 and described in Table 3-128.

This register controls EDMA3CC Event mux assignment for event no 4 to 7

**Figure 3-126. EDMA3CC_EVTMUX_4_7 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-128. EDMA3CC_EVTMUX_4_7 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_7 | R/W | 0h | Event mux for EDMA3CC event number 7 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_6 | R/W | 0h | Event mux for EDMA3CC event number 6 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_5 | R/W | 0h | Event mux for EDMA3CC event number 5 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_4 | R/W | 0h | Event mux for EDMA3CC event number 4 |

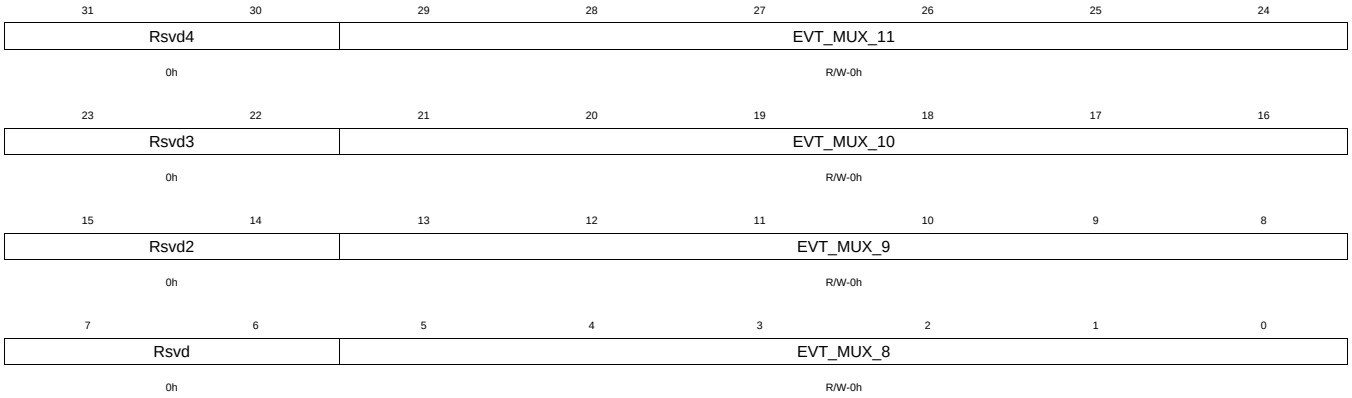

Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.126 EDMA3CC_EVTMUX_8_11 Register (offset = F98h) [reset = 0h]

EDMA3CC_EVTMUX_8_11 is shown in Figure 3-127 and described in Table 3-129.

This register controls EDMA3CC Event mux assignment for event no 8 to 11

**Figure 3-127. EDMA3CC_EVTMUX_8_11 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-129. EDMA3CC_EVTMUX_8_11 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_11 | R/W | 0h | Event mux for EDMA3CC event number 11 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_10 | R/W | 0h | Event mux for EDMA3CC event number 10 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_9 | R/W | 0h | Event mux for EDMA3CC event number 9 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_8 | R/W | 0h | Event mux for EDMA3CC event number 8 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.127 EDMA3CC_EVTMUX_12_15 Register (offset = F9Ch) [reset = 0h]

EDMA3CC_EVTMUX_12_15 is shown in Figure 3-128 and described in Table 3-130.

This register controls EDMA3CC Event mux assignment for event no 12 to 15

**Figure 3-128. EDMA3CC_EVTMUX_12_15 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_15 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_14 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_13 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_12 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-130. EDMA3CC_EVTMUX_12_15 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_15 | R/W | 0h | Event mux for EDMA3CC event number 15 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_14 | R/W | 0h | Event mux for EDMA3CC event number 14 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_13 | R/W | 0h | Event mux for EDMA3CC event number 13 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_12 | R/W | 0h | Event mux for EDMA3CC event number 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.128 EDMA3CC_EVTMUX_16_19 Register (offset = FA0h) [reset = 0h]

EDMA3CC_EVTMUX_16_19 is shown in Figure 3-129 and described in Table 3-131.

This register controls EDMA3CC Event mux assignment for event no 16 to 19

**Figure 3-129. EDMA3CC_EVTMUX_16_19 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_19 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_18 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_17 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_16 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-131. EDMA3CC_EVTMUX_16_19 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_19 | R/W | 0h | Event mux for EDMA3CC event number 19 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_18 | R/W | 0h | Event mux for EDMA3CC event number 18 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_17 | R/W | 0h | Event mux for EDMA3CC event number 17 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_16 | R/W | 0h | Event mux for EDMA3CC event number 16 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.129 EDMA3CC_EVTMUX_20_23 Register (offset = FA4h) [reset = 0h]

EDMA3CC_EVTMUX_20_23 is shown in Figure 3-130 and described in Table 3-132.

This register controls EDMA3CC Event mux assignment for event no 20 to 23

**Figure 3-130. EDMA3CC_EVTMUX_20_23 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_23 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_22 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_21 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_20 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-132. EDMA3CC_EVTMUX_20_23 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_23 | R/W | 0h | Event mux for EDMA3CC event number 23 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_22 | R/W | 0h | Event mux for EDMA3CC event number 22 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_21 | R/W | 0h | Event mux for EDMA3CC event number 21 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_20 | R/W | 0h | Event mux for EDMA3CC event number 20 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.130 EDMA3CC_EVTMUX_24_27 Register (offset = FA8h) [reset = 0h]

EDMA3CC_EVTMUX_24_27 is shown in Figure 3-131 and described in Table 3-133.

This register controls EDMA3CC Event mux assignment for event no 24 to 27

**Figure 3-131. EDMA3CC_EVTMUX_24_27 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_27 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_26 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_25 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_24 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-133. EDMA3CC_EVTMUX_24_27 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_27 | R/W | 0h | Event mux for EDMA3CC event number 27 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_26 | R/W | 0h | Event mux for EDMA3CC event number 26 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_25 | R/W | 0h | Event mux for EDMA3CC event number 25 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_24 | R/W | 0h | Event mux for EDMA3CC event number 24 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.131 EDMA3CC_EVTMUX_28_31 Register (offset = FACh) [reset = 0h]

EDMA3CC_EVTMUX_28_31 is shown in Figure 3-132 and described in Table 3-134.

This register controls EDMA3CC Event mux assignment for event no 28 to 31

**Figure 3-132. EDMA3CC_EVTMUX_28_31 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_31 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_30 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_29 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_28 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-134. EDMA3CC_EVTMUX_28_31 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_31 | R/W | 0h | Event mux for EDMA3CC event number 31 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_30 | R/W | 0h | Event mux for EDMA3CC event number 30 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_29 | R/W | 0h | Event mux for EDMA3CC event number 29 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_28 | R/W | 0h | Event mux for EDMA3CC event number 28 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.132 EDMA3CC_EVTMUX_32_35 Register (offset = FB0h) [reset = 0h]

EDMA3CC_EVTMUX_32_35 is shown in Figure 3-133 and described in Table 3-135.

This register controls EDMA3CC Event mux assignment for event no 32 to 35

**Figure 3-133. EDMA3CC_EVTMUX_32_35 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_35 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_34 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_33 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_32 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-135. EDMA3CC_EVTMUX_32_35 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_35 | R/W | 0h | Event mux for EDMA3CC event number 35 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_34 | R/W | 0h | Event mux for EDMA3CC event number 30 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_33 | R/W | 0h | Event mux for EDMA3CC event number 33 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_32 | R/W | 0h | Event mux for EDMA3CC event number 32 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.133 EDMA3CC_EVTMUX_36_39 Register (offset = FB4h) [reset = 0h]

EDMA3CC_EVTMUX_36_39 is shown in Figure 3-134 and described in Table 3-136.

This register controls EDMA3CC Event mux assignment for event no 36 to 39

**Figure 3-134. EDMA3CC_EVTMUX_36_39 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_39 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_38 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_37 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_36 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-136. EDMA3CC_EVTMUX_36_39 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_39 | R/W | 0h | Event mux for EDMA3CC event number 39 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_38 | R/W | 0h | Event mux for EDMA3CC event number 38 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_37 | R/W | 0h | Event mux for EDMA3CC event number 37 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_36 | R/W | 0h | Event mux for EDMA3CC event number 36 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.134 EDMA3CC_EVTMUX_40_43 Register (offset = FB8h) [reset = 0h]

EDMA3CC_EVTMUX_40_43 is shown in Figure 3-135 and described in Table 3-137.

This register controls EDMA3CC Event mux assignment for event no 40 to 43

**Figure 3-135. EDMA3CC_EVTMUX_40_43 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_43 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_42 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_41 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_40 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-137. EDMA3CC_EVTMUX_40_43 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_43 | R/W | 0h | Event mux for EDMA3CC event number 43 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_42 | R/W | 0h | Event mux for EDMA3CC event number 42 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_41 | R/W | 0h | Event mux for EDMA3CC event number 41 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_40 | R/W | 0h | Event mux for EDMA3CC event number 40 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.135 EDMA3CC_EVTMUX_44_47 Register (offset = FBCh) [reset = 0h]

EDMA3CC_EVTMUX_44_47 is shown in Figure 3-136 and described in Table 3-138.

This register controls EDMA3CC Event mux assignment for event no 44 to 47

**Figure 3-136. EDMA3CC_EVTMUX_44_47 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd8 | | EVT_MUX_47 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd7 | | EVT_MUX_46 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd6 | | EVT_MUX_45 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd5 | | EVT_MUX_44 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-138. EDMA3CC_EVTMUX_44_47 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd8 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_47 | R/W | 0h | Event mux for EDMA3CC event number 47 |
| 23-22 | Rsvd7 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_46 | R/W | 0h | Event mux for EDMA3CC event number 46 |
| 15-14 | Rsvd6 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_45 | R/W | 0h | Event mux for EDMA3CC event number 45 |
| 7-6 | Rsvd5 | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_44 | R/W | 0h | Event mux for EDMA3CC event number 44 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.136 EDMA3CC_EVTMUX_48_51 Register (offset = FC0h) [reset = 0h]

EDMA3CC_EVTMUX_48_51 is shown in Figure 3-137 and described in Table 3-139.

This register controls EDMA3CC Event mux assignment for event no 48 to 51

**Figure 3-137. EDMA3CC_EVTMUX_48_51 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_51 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_50 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_49 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_48 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-139. EDMA3CC_EVTMUX_48_51 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_51 | R/W | 0h | Event mux for EDMA3CC event number 51 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_50 | R/W | 0h | Event mux for EDMA3CC event number 50 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_49 | R/W | 0h | Event mux for EDMA3CC event number 49 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_48 | R/W | 0h | Event mux for EDMA3CC event number 48 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.137 EDMA3CC_EVTMUX_52_55 Register (offset = FC4h) [reset = 0h]

EDMA3CC_EVTMUX_52_55 is shown in Figure 3-138 and described in Table 3-140.

This register controls EDMA3CC Event mux assignment for event no 52 to 55

**Figure 3-138. EDMA3CC_EVTMUX_52_55 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_55 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_54 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_53 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_52 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-140. EDMA3CC_EVTMUX_52_55 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_55 | R/W | 0h | Event mux for EDMA3CC event number 55 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_54 | R/W | 0h | Event mux for EDMA3CC event number 54 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_53 | R/W | 0h | Event mux for EDMA3CC event number 53 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_52 | R/W | 0h | Event mux for EDMA3CC event number 52 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.138 EDMA3CC_EVTMUX_56_59 Register (offset = FC8h) [reset = 0h]

EDMA3CC_EVTMUX_56_59 is shown in Figure 3-139 and described in Table 3-141.

This register controls EDMA3CC Event mux assignment for event no 56 to 59

**Figure 3-139. EDMA3CC_EVTMUX_56_59 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_59 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_58 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_57 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_56 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-141. EDMA3CC_EVTMUX_56_59 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_59 | R/W | 0h | Event mux for EDMA3CC event number 59 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_58 | R/W | 0h | Event mux for EDMA3CC event number 58 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_57 | R/W | 0h | Event mux for EDMA3CC event number 57 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_56 | R/W | 0h | Event mux for EDMA3CC event number 56 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.139 EDMA3CC_EVTMUX_60_63 Register (offset = FCCh) [reset = 0h]

EDMA3CC_EVTMUX_60_63 is shown in Figure 3-140 and described in Table 3-142.

This register controls EDMA3CC Event mux assignment for event no 60 to 63

**Figure 3-140. EDMA3CC_EVTMUX_60_63 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd4 | | EVT_MUX_63 | | | | | |
| 0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Rsvd3 | | EVT_MUX_62 | | | | | |
| 0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Rsvd2 | | EVT_MUX_61 | | | | | |
| 0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | | EVT_MUX_60 | | | | | |
| 0h | | R/W-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 3-142. EDMA3CC_EVTMUX_60_63 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Rsvd4 | | 0h | Reserved-Read returns 0 |
| 29-24 | EVT_MUX_63 | R/W | 0h | Event mux for EDMA3CC event number 63 |
| 23-22 | Rsvd3 | | 0h | Reserved-Read returns 0 |
| 21-16 | EVT_MUX_62 | R/W | 0h | Event mux for EDMA3CC event number 62 |
| 15-14 | Rsvd2 | | 0h | Reserved-Read returns 0 |
| 13-8 | EVT_MUX_61 | R/W | 0h | Event mux for EDMA3CC event number 61 |
| 7-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5-0 | EVT_MUX_60 | R/W | 0h | Event mux for EDMA3CC event number 60 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.140 GPIO_MUX Register (offset = FD4h) [reset = 0h]

GPIO_MUX is shown in Figure 3-141 and described in Table 3-143.

This register controls the input source for GPIO1_0 to GPIO1_5 inputs. The input source can either come from the device pin (as a GPIO), or from an internal interrupt source.

**Figure 3-141. GPIO_MUX Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd | | | | | | | |
| 0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Rsvd | | | | | | | |
| 0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Rsvd | | | | | | | |
| 0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | | GPIO1_5_MUX | GPIO1_4_MUX | GPIO1_3_MUX | GPIO1_2_MUX | GPIO1_1_MUX | GPIO1_0_MUX |
| 0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-143. GPIO_MUX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Rsvd | | 0h | Reserved-Read returns 0 |
| 5 | GPIO1_5_MUX | R/W | 0h | GPIO1_5 input coming from :<br>0x0: GPIO1_5 pad<br>0x1: USB1 charge detect |
| 4 | GPIO1_4_MUX | R/W | 0h | GPIO1_4 input coming from :<br>0x0: GPIO1_4 pad<br>0x1: USB0 charge detect |
| 3 | GPIO1_3_MUX | R/W | 0h | GPIO1_3 input coming from :<br>0x0: GPIO1_3 pad<br>0x1: bandgap1 Tshut |
| 2 | GPIO1_2_MUX | R/W | 0h | GPIO1_2 input coming from :<br>0x0: GPIO1_2 pad<br>0x1: bandgap0 Tshut |
| 1 | GPIO1_1_MUX | R/W | 0h | GPIO1_1 input coming from :<br>0x0: GPIO1_1 pad<br>0x1: vdac_tvint (tv detect) |
| 0 | GPIO1_0_MUX | R/W | 0h | GPIO1_0 input coming from :<br>0x0: GPIO1_0 pad<br>0x1: vdac_tvint (tv detect) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 3.2.141 RESET_ISO Register (offset = 1000h) [reset = 0h]

RESET_ISO is shown in Figure 3-142 and described in Table 3-144.

This register controls the reset isolation operation for the Ethernet Subsystem.

**Figure 3-142. RESET_ISO Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Rsvd | | | | | | | |
| 0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Rsvd | | | | | | | |
| 0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Rsvd | | | | | | | |
| 0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | | | | RMII_ISO1 | RMII_ISO0 | GMII_ISO | ISO_CONTROL |
| 0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 3-144. RESET_ISO Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | Rsvd | | 0h | Reserved, driven low. |
| 3 | RMII_ISO1 | R/W | 0h | RMII Mux1 Pin Groups Isolation enable<br>0x0 : Pin group is not isolated.<br>0x1 : The following pin groups are isolated from reset: EMAC_RMREFCLK, EMAC[1]_RMx (Option 1), EMAC[0]_RMx, and MDIO/MDCLK. Refer to the Reset Isolation section for pin group definitions. |
| 2 | RMII_ISO0 | R/W | 0h | RMII Mux0 Pin Groups Isolation enable<br>0x0 : Pin group is not isolated.<br>0x1 : The following pin groups are isolated from reset: EMAC_RMREFCLK, EMAC[1]_RMx (Option 0), EMAC[0]_RMx, and MDIO/MDCLK. Refer to the Reset Isolation section for pin group definitions. |
| 1 | GMII_ISO | R/W | 0h | GMII Pin Groups Isolation enable<br>0x0 : Pin group is not isolated.<br>0x1 : The following pin groups are isolated from reset: EMAC[1]_Mx/Other, EMAC[0]_Mx/EMAC[1:0]_RGx, and MDIO/MDCLK. Refer to the Reset Isolation section for pin group definitions. |
| 0 | ISO_CONTROL | R/W | 0h | Ethernet Reset Isolation control<br>0x0 : Reset isolation is disabled<br>0x1 : The Ethernet subsystem is isolated from reset, along with the pin groups selected by the GPII_ISO, RMII_ISO0, and RMII_ISO1 settings. Exactly one bit of either the GMII_ISO, RMII_ISO0, or RMII_ISO1 should be set in this mode. Refer to the Reset Isolation section for pin group definitions. |


Copyright © 2011–2016, Texas Instruments Incorporated

# ROM Code Memory and Peripheral Booting

This chapter describes the booting functionality of the public ROM code, referred hereafter as ROM Code. The booting functionality covers the following features:

- *Memory Booting* – booting the device by starting code stored on permanent memories like flash-memory or memory cards. This process is usually performed after either device cold or warm reset.
- *Peripheral Booting* – booting the device by downloading the executable code over a communication interface like UART or Ethernet. This process is intended for flashing a device.

The device always starts up in secure mode. The ROM Code takes care of early initialization. The ROM code switches the device into public mode; hence, the Public ROM Code provides run-time services for cache maintenance.

This chapter is intended for all who design systems based on this device, develop bootloader images and flashing tools.

**Topic** **Page**

**4.1 Introduction** ..... **1018**
**4.2 Overview** ..... **1019**
**4.3 Memory Map** ..... **1021**
**4.4 Startup and Configuration** ..... **1026**
**4.5 Booting** ..... **1027**
**4.6 Fast External Booting** ..... **1030**
**4.7 Memory Booting** ..... **1031**
**4.8 Peripheral Booting** ..... **1060**
**4.9 Image Format** ..... **1065**
**4.10 Execution** ..... **1067**
**4.11 Services for HLOS Support** ..... **1068**
**4.12 Tracing** ..... **1069**


Copyright © 2011–2016, Texas Instruments Incorporated

## 4.1 Introduction

This chapter describes the booting functionality of the Public ROM Code.

### *4.1.1 Acronyms and Naming Conventions*

The following tables provides acronyms and abbreviations, as well as the naming conventions used in this document.

**Table 4-1. Acronyms and Abbreviations**

| Acronym/Abbreviation | Definition |
|---|---|
| ASIC | Application Specific Integrated Circuit |
| B | Byte (8 bits) |
| CS | Chip Select |
| CH | Configuration Header |
| DPLL | Digital PLL |
| EMIF | External Memory InterFace (SDRAM Controller) |
| FIQ | Fast Interrupt Request |
| FS USB | Full Speed USB |
| GPMC | General Purpose Memory Controller |
| HLOS | High Level Operating System |
| HS USB | High Speed USB |
| HW | Hardware |
| HWA | Hardware Accelerator |
| I2C | Inter-Integrated Circuit |
| IRQ | Interrupt |
| ISR | Interrupt Service Routine |
| JTAG | Joint Test Action Group. This term is used to refer to the Debugging and Testing interface. |
| Kbps | Kilobits per second |
| KB | Kilobyte, 1024 B |
| LPDDR2 | Low Power Double Data Rate rev. 2 |
| LPDDR2-NVM | Non volatile type of memory connected to EMIF interface |
| MB | Megabyte, 1024 KB |
| MMC | Multimedia Card. |
| MPU | Micro Processor Unit |
| OCM | On-Chip Memory |
| OneNAND™ | A type of Flash memory |
| PA | Protected Application (also see Table 4-2) |
| PLL | Phase Locked Loop |
| PK | Public Key |
| PPA | Primary Protected Application (also see Table 4-2) |
| POR | Power On Reset |
| PMU | Power Management Unit |
| PRCM | Power, Reset and Clock Management module |
| RAM | Random Access Memory. Refers to the OMAP internal static RAM |
| ROM | Read Only Memory |
| RSA | Rivest, Shamir, Adleman encryption algorithm |
| SAR | Save & Restore |
| SDRAM | Synchronous Dynamic Random Access Memory |
| SWI | Software Interrupt |
| TOC | Table Of Contents – a structure within an executable image |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-1. Acronyms and Abbreviations (continued)**

| Acronym/Abbreviation | Definition |
|---|---|
| UART | Universal Asynchronous Receiver/Transmitter |
| USB | Universal Serial Bus |
| XIP | Execute In Place |
| WDT | Watchdog timer |

**Table 4-2. Naming Conventions**

| Name | Definition |
|---|---|
| Bootstrap | Initial SW that is launched by the ROM Code during the Memory Booting phase |
| Downloaded SW | Initial SW that is downloaded into internal RAM by the ROM Code during peripheral booting phase. |
| E-Fuse | A one-time programmable memory location usually set at the factory |
| Flash Loader | Downloaded software launched by the ROM Code in Pre-Flashing and which programs an image into external memories. |
| Initial SW | Software which is executed by any of the ROM Code mechanisms (Memory Booting or Peripheral Booting). Initial software is a generic term for Bootstrap and Downloaded software. |
| Memory Booting | ROM Code mechanism that consists of executing an Initial SW from external memory. |
| Monitor Mode Manager | ROM Code software component in charge of handling all entry/exit to/from MPU Secure State. |
| Peripheral Booting | ROM Code mechanism that consists of polling selected interfaces, downloading and executing an Initial SW (in this case called Downloaded SW) in internal RAM. |
| Pre-Flashing | Specific case of peripheral booting, where the ROM Code mechanism is used to program external Flash memories. |

## 4.2 Overview

### *4.2.1 Architecture*

The architecture of the ROM code is shown in Figure 4-1. It is split into three main layers with a top-down approach: high-level, drivers, and hardware abstraction layer (HAL). One layer communicates with a lower level layer through a unified interface.

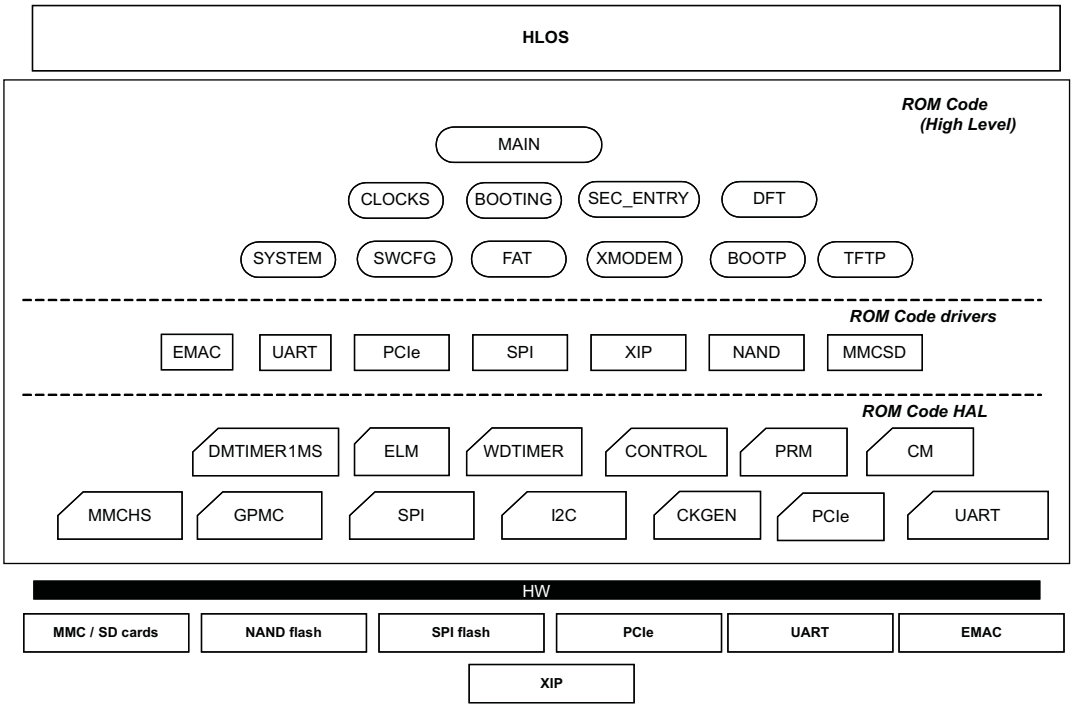
The high level layer is in charge of the main tasks of the ROM Code: watchdog and clocks configuration and main booting routine.

The drivers layer implements the logical and communication protocols for any booting device in accordance with the interface specification.

Finally the HAL implements the lowest level code for interacting with the hardware infrastructure IPs. End booting devices are attached to device IO pads.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 4-1. ROM Code Architecture**




### 4.2.2 Functionality

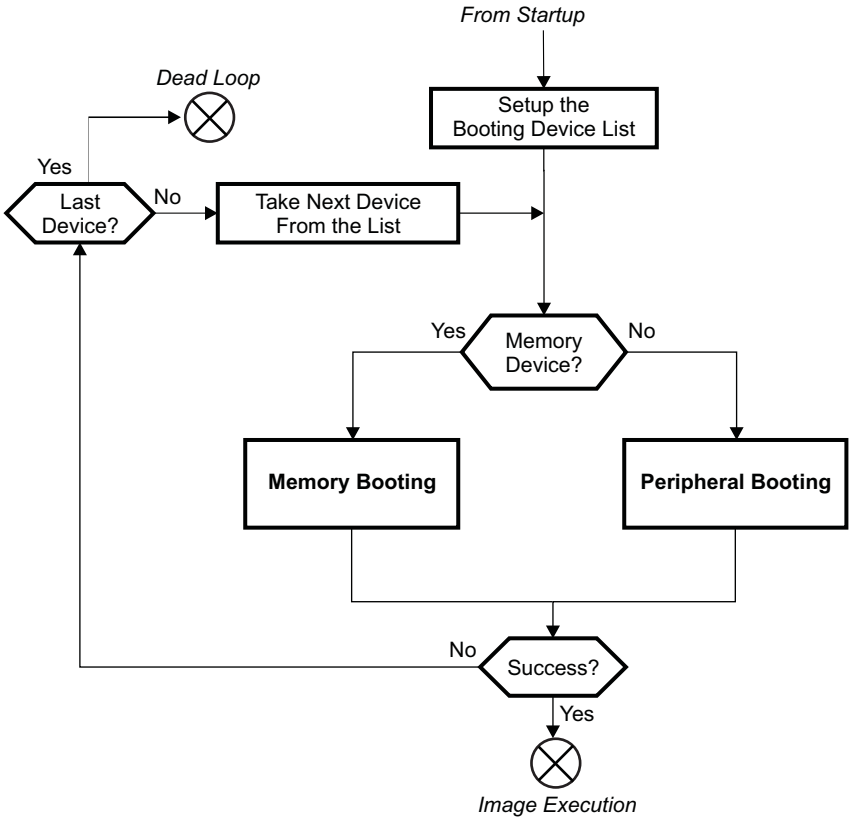
Figure 4-2 illustrates the high level flow for the Public ROM Code booting procedure. On the device, the Public ROM Code starts upon completion of the secure startup (performed by the Secure ROM Code). The ROM Code then performs platform configuration and initialization as part of the public start-up procedure.

The booting device list is created based on the MBOOT. A booting device can be a memory booting device (soldered flash memory or temporarily booting device like memory card) or a peripheral interface connected to a host.

The main loop of the booting procedure goes through the booting device list and tries to search for an image from the currently selected booting device. This loop is exited if a valid booting image is found and successfully executed or upon watchdog expiration.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 4-2. ROM Code Boot Procedure**




## 4.3 Memory Map

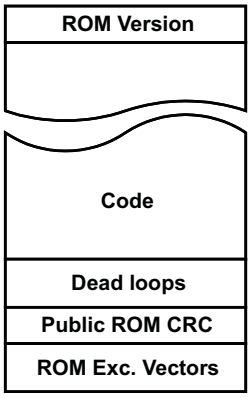
### *4.3.1 ROM Memory Map*

The on-chip ROM memory map is shown in Figure 4-3. The ROM Code mapping consists in the following:

- Exception vectors
- CRC
- Dead loops collection
- Code and const data sections
- API Table
- ROM Version


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 4-3. ROM Memory Map**




#### 4.3.1.1 ROM Exception Vectors

Table 4-3 lists the ROM exception vectors. The reset exception is redirected to the ROM Code startup. Other exceptions are redirected to their RAM handlers by loading appropriate addresses into the PC register.

**Table 4-3. ROM Exception Vectors**

| Address | Exception | Content |
|---|---|---|
| 20000h | Reset | Branch to the ROM Code startup |
| 20004h | Undefined | PC = 4030 D004h |
| 20008h | SWI | PC = 4030 D008h |
| 2000Ch | Pre-fetch abort | PC = 4030 D00Ch |
| 20010h | Data abort | PC = 4030 D010h |
| 20014h | Unused | PC = 4030 D014h |
| 20018h | IRQ | PC = 4030 D018h |
| 2001Ch | FIQ | PC = 4030 D01Ch |

#### 4.3.1.2 ROM Code CRC

The ROM Code CRC is calculated as 32 bit CRC code (CRC-32-IEEE 802.3) for the address range 20000h–2BFFFh. The four bytes CRC code is stored at location 20020h.

#### 4.3.1.3 Dead Loops

Built-in dead loops are used for different purposes as shown in Table 4-4. All dead loops are branch instructions coded in ARM mode. The fixed location of these dead loops facilitates debugging and testing. The first seven dead loops are default exception handlers linked with RAM exception vectors. The dead loops might be called directly from the user code. However there exists a special function which can be called from ROM code in order to execute a dead loop. The function is an assembly code in ARM mode which takes the dead loop address from the R0 register. The main purpose of the function is to issue a global SW reset before going to a dead loop. The function is located at address 200C0h. In addition the function clears global cold reset status upon issuing the global SW reset.

**Table 4-4. Dead Loops**

| Address | Purpose |
|---|---|
| 20080h | Undefined exception default handler |
| 20084h | SWI exception default handler |
| 20088h | Pre-fetch abort exception default handler |
| 2008Ch | Data abort exception default handler |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-4. Dead Loops (continued)**

| Address | Purpose |
|---|---|
| 20090h | Unused exception default handler |
| 20094h | IRQ exception default handler |
| 20098h | FIQ exception default handler |
| 2009Ch | Validation tests PASS |
| 200A0h | Validation tests FAIL |
| 200A4h | Reserved |
| 200A8h | Image not executed or returned. |
| 200ACh | Reserved |
| 200B0h | Reserved |
| 200B4h | Reserved |
| 200B8h | Reserved |
| 200BCh | Reserved |

#### 4.3.1.4 Code

This space is used to hold code and constant data.

#### 4.3.1.5 Public ROM Code Version

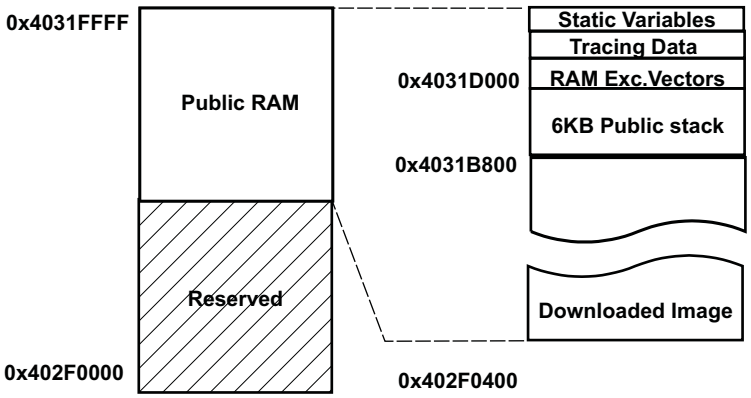
The ROM code version consists of two decimal numbers: major and minor. It can be used to identify the ROM Code release version burnt in a given IC (e.g., useful to identify an IC version). The ROM code version is a 32 bits hexadecimal value located at address 2BFFCh.


Copyright © 2011–2016, Texas Instruments Incorporated

### 4.3.2 RAM Memory Map

The ROM code makes use of the on chip RAM module connected to the L3 interconnect (further referred as L3 RAM in this document). Its usage is shown in Figure 4-4. The RAM memory map ranges from address 4020F000h to 4031FFFFh.

**Figure 4-4. Public RAM Memory Map**




#### 4.3.2.1 Downloaded Image

This area is used by the ROM code to store the downloaded boot image. It can be up to 173 KB.

#### 4.3.2.2 Public Stack

Space reserved for stack.

#### 4.3.2.3 RAM Exception Vectors

The RAM exception vectors enable a simple means for redirecting exceptions to custom handlers. Table 4-5 shows content of the RAM space reserved for RAM vectors. The first seven addresses are ARM instructions which load the value located in the subsequent seven addresses into the PC register. Theses instructions are executed when an exception occurs since they are called from the ROM exception vectors. Undefined, SWI, Unused and FIQ exceptions are redirected to a hardcoded dead loop. Pre-fetch abort, data abort, and IRQ exception are redirected to pre-defined ROM handlers. User code can redirect any exception to a custom handler either by writing its address to the appropriate location from 4031 D024h to 4031 D03Ch or by overriding the branch (load into PC) instruction between addresses from 4031 D004h to 4031 D01Ch.

**Table 4-5. RAM Exception Vectors**

| Address | Exception | Content |
|---|---|---|
| 4031 D000h | Reserved | Reserved |
| 4031 D004h | Undefined | PC = [4031 D024h] |
| 4031 D008h | SWI | PC = [4031 D028h] |
| 4031 D00Ch | Pre-fetch abort | PC = [4031 D02Ch] |
| 4031 D010h | Data abort | PC = [4031 D030h] |
| 4031 D014h | Unused | PC = [4031 D034h] |
| 4031 D018h | IRQ | PC = [4031 D038h] |
| 4031 D01Ch | FIQ | PC = [4031 D03Ch] |
| 4031 D020h | Reserved | 20090h |
| 4031 D024h | Undefined | 20080h |
| 4031 D028h | SWI | 20084h |
| 4031 D02Ch | Pre-fetch abort | Address of default pre-fetch abort handler (*) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-5. RAM Exception Vectors (continued)**

| Address | Exception | Content |
|---|---|---|
| 4031 D030h | Data abort | Address of default data abort handler (*) |
| 4031 D034h | Unused | 20090h |
| 4031 D038h | IRQ | Address of default IRQ handler |
| 4031 D03Ch | FIQ | 20098h |

(*) the default handlers for pre-fetch and data abort are performing reads from CP15 debug registers to retrieve the reason of the abort:

- In case of pre-fetch abort: the IFAR register is read from CP15 and stored into R0. The IFSR register is read and stored into the R1 register. Then the ROM Code jumps to the pre-fetch abort dead loop (20088h).
- In case of data abort: the DFAR register is read from CP15 and stored into R0. The DFSR register is read and stored into the R1 register. Then the ROM Code jumps to the data abort dead loop (2008Ch).

#### 4.3.2.4 Tracing Data

This section contains trace vectors reflecting the execution path of the public boot. Section 4.12 describes the usage of the different trace vectors and lists all the possible trace codes.

**Table 4-6. Tracing Data**

| Address | Size [bytes] | Description |
|---|---|---|
| 4031 D040h | 4 | Current tracing vector, word 1 |
| 4031 D044h | 4 | Current tracing vector, word 2 |
| 4031 D048h | 4 | Current tracing vector, word 3 |
| 4031 D04Ch | 4 | Current copy of the PRM_RSTST register (reset reasons) |
| 4031 D050h | 4 | Cold reset run tracing vector, word 1 |
| 4031 D054h | 4 | Cold reset run tracing vector, word 2 |
| 4031 D058h | 4 | Cold reset run tracing vector, word 3 |
| 4031 D05Ch | 4 | Reserved |
| 4031 D060h | 4 | Reserved |
| 4031 D064h | 4 | Reserved |

#### 4.3.2.5 Static Variables

This area contains the ROM code static variables used during boot time (and possibly during run-time, if calling the Public ROM API functions).


Copyright © 2011–2016, Texas Instruments Incorporated

## 4.4 Startup and Configuration

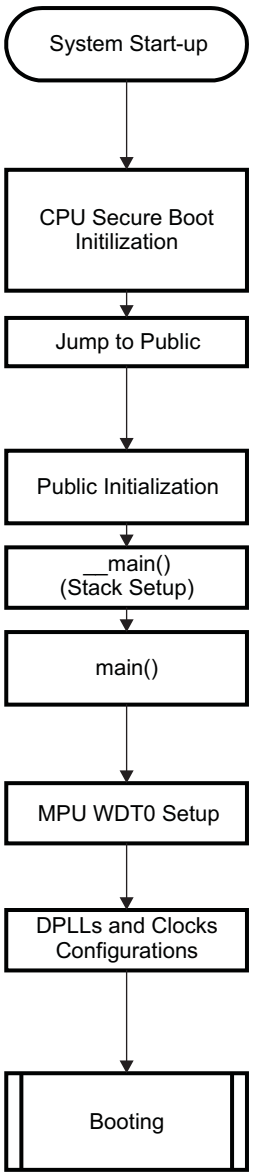
### 4.4.1 ROM Code Start-up

The main MPU subsystem always starts its execution in secure mode after reset due to the TrustZone architecture (the Secure ROM code implements the reset handler). The Public ROM code is physically located at the address 20000h that is immediately next to the Secure ROM code.

As shown at top of Figure 4-5, the CPU jumps to the Public ROM Code reset vector once it has completed the secure boot initialization.

Once in public mode the CPU performs the public-side initialization and stack setup (compiler auto generated C- initialization or “scatter loading”) as shown in left hand side of Figure 4-5. Then it configures the watchdog timer 2 (set to three minutes), and performs system clocks configuration. Finally it jumps to the booting routine.

**Figure 4-5. ROM Code Start-up Sequence**





Copyright © 2011–2016, Texas Instruments Incorporated

### 4.4.2 CPU State at Public Startup

The CPU L1 instruction cache and branch prediction mechanisms are not activated as part of the public boot process. The public vector base address is configured to the reset vector of Public ROM Code (20000h). MMU is left switched off during the public boot (hence L1 data cache off).

No specific configuration is performed for the slave CPU which keeps its default configuration after reset (L1 instruction and data caches off, branch prediction off, MMU off, no remap of public vectors base address).

### 4.4.3 Clocking Configuration

The supported system clock frequencies are OSC0 = 20 MHz.

The ROM Code configures the clocks and DPLLs that are necessary for ROM Code execution:

- L3 DPLL locked to provide 220MHz clocks for peripheral blocks
- DDR DPLL locked to provide 400MHz
- ARM DPLL is locked to provide 600MHz for the A8
- USB DPLL is locked to provide 960MHz and 192MHz for peripheral blocks

Table 4-7 summarizes the ROM Code default settings for clocks. This default configuration enables all the ROM Code functionalities with minimized needs on power during boot.

The DPLLs and PRCM clock dividers are configured with the ROM Code default values after cold or warm reset in order to give the same working conditions to the Public ROM Code sequence

**Table 4-7. ROM Code Default Clock Settings**

| Clock | Frequency (MHz) | Source |
|---|---|---|
| PLL_ARM | 600 | OSC0 |
| PLL_L3 | 220 | OSC0 |
| PLL_USB | 960 | OSC0 |
| PLL_DDR | 400 | OSC0 |
| SYSCLK4 | 220 | PLL_L3/1 |
| SYSCLK8 | 192 | (PLL_USB/5)/1 |
| SYSCLK10 | 48 | (PLL_USB/5)/4 |
| SYSCLK18B | 32KHz | RTCDIVIDER |

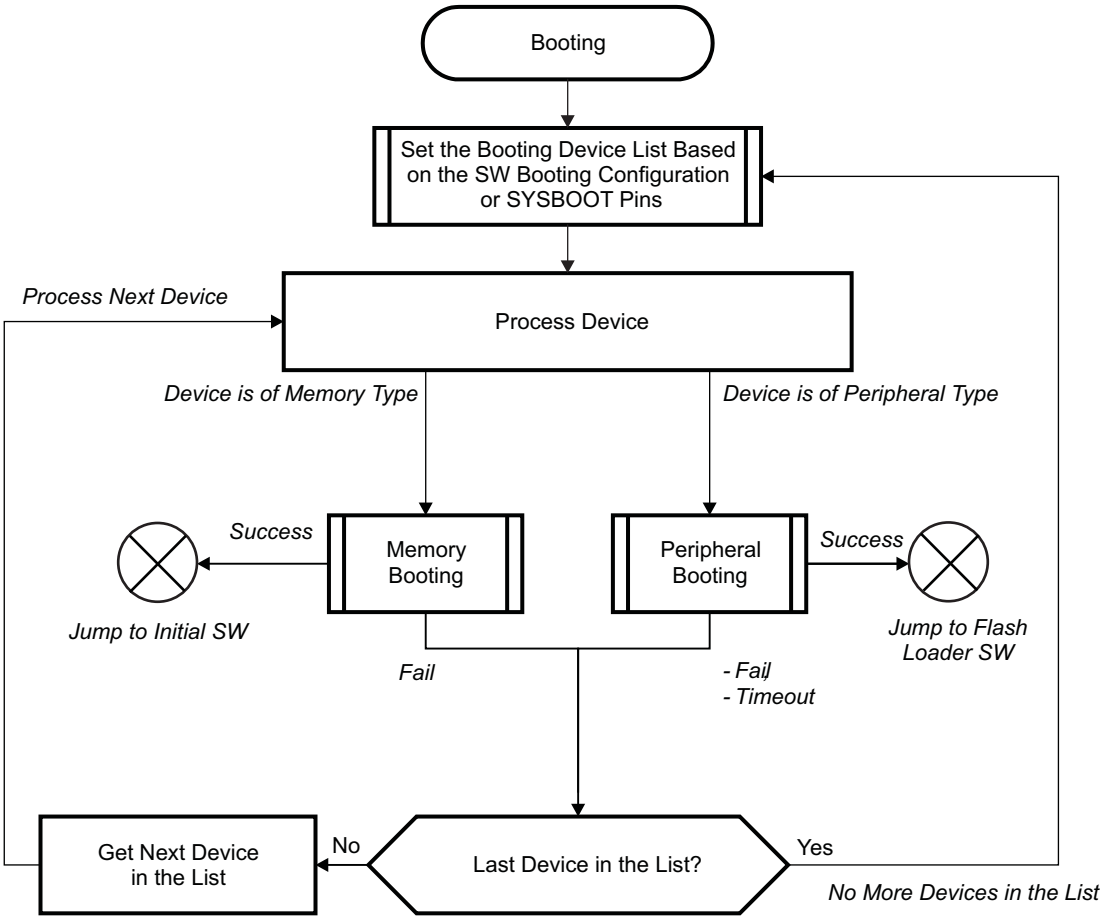
## 4.5 Booting

### 4.5.1 Overview

Figure 4-6 shows the booting procedure. First, a booting device list is created. The list consists of all devices which will be searched for a booting image. The list is filled in based on the SYSBOOT.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 4-6. ROM Code Booting Procedure**




Once the booting device list is set up, the booting routine examines the devices enumerated in the list sequentially and either executes the memory booting or peripheral booting procedure depending on the booting device type. The memory booting procedure is executed when the booting device type is one of NOR, NAND, MMC, or SPI-EEPROM. The peripheral booting is executed when the booting device type is Ethernet, PCIe or UART.

The memory booting procedure reads data from a memory type device. If a valid booting image is found and successfully read from the external memory device, the initial SW is simply started.

The peripheral booting procedure downloads data from a host (commonly a PC) to the device by means of Ethernet, PCIe or UART links. The ROM Code uses a host-slave logical protocol for synchronization. Upon successful PCIe enumeration (or UART or Ethernet connection) the host sends the image binary contents. The peripheral booting procedure is described in detail in Section 4.8.

If the memory or peripheral booting fails for all devices enumerated in the device list, then the ROM Code gets back to the first device in the list. The device list is then processed again in a loop. This loop shall be further interrupted by the MPU or secure watchdog.

### 4.5.2 Device List

The ROM Code creates the device list based on information gathered from the SYSBOOT configuration pins sensed in the control module. The pins are used to index the device table from which the list of devices is extracted.

#### 4.5.2.1 BTMode[4-0] Configuration Pins

The following table describes the BTMode[4-0] configuration pins.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-8. BTMode[4-0] Configuration Pins**

| Boot Modes | | | | BTMODE[4:0] |
|---|---|---|---|---|
| 1st | 2nd | 3rd | 4th | |
| Reserved | Reserved | Reserved | Reserved | 00000 |
| UART | XIP w/ WAIT (MUX0) | MMC | SPI | 00001 |
| UART | SPI | NAND | NANDI2C | 00010 |
| UART | SPI | XIP (MUX0) | MMC | 00011 |
| EMAC | SPI | NAND | NANDI2C | 00100 |
| Reserved | Reserved | Reserved | Reserved | 00101 |
| Reserved | Reserved | Reserved | Reserved | 00110 |
| EMAC | MMC | SPI | XIP (MUX 1) | 00111 |
| PCIE_32 | Reserved | Reserved | Reserved | 01000 |
| PCIE_64 | Reserved | Reserved | Reserved | 01001 |
| Reserved | Reserved | Reserved | Reserved | 01010 |
| Reserved | Reserved | Reserved | Reserved | 01011 |
| Reserved | Reserved | Reserved | Reserved | 01100 |
| Reserved | Reserved | Reserved | Reserved | 01101 |
| Reserved | Reserved | Reserved | Reserved | 01110 |
| Fast External Boot | UART | EMAC | PCIE_64 | 01111 |
| XIP (MUX1) | UART | EMAC | MMC | 10000 |
| XIP w/WAIT (MUX1) | UART | EMAC | MMC | 10001 |
| NAND | NANDI2C | SPI | UART | 10010 |
| NAND | NANDI2C | MMC | UART | 10011 |
| NAND | NAND12C | SPI | EMAC | 10100 |
| NANDI2C | MMC | EMAC | UART | 10101 |
| SPI | MMC | UART | EMAC | 10110 |
| MMC | SPI | UART | EMAC | 10111 |
| SPI | MMC | PCIE_32 | Reserved | 11000 |
| SPI | MMC | PCIe_64 | Reserved | 11001 |
| XIP (MUX0) | UART | SPI | MMC | 11010 |
| XIP w/ WAIT (MUX0) | UART | SPI | MMC | 11011 |
| Reserved | Reserved | Reserved | Reserved | 11100 |
| Reserved | Reserved | Reserved | Reserved | 11101 |
| Reserved | Reserved | Reserved | Reserved | 11110 |
| Fast external Boot | EMAC | UART | PCIE_32 | 11111 |

The ROM Code uses the row pointed by the MBOOT configuration value .The device list is filled in with the 1st to 4th devices. is the decoding table for MBOOT configuration pins. The following shortcuts are used in the table:

| MMC | MMC or SD card (MMC port 1) |
|---|---|
| NAND / NANDI2C | NAND flash memory / read flash geometry from I2C EEPROM |
| XIP / XIP w/ WAIT | NOR or other XIP device with or without wait monitoring |
| UART | UART interface (UART port 0) |
| EMAC | Ethernet GMII0m RGMII and RMII interface (EMAC port 0) |
| SPI | SPI EEPROM (SPI 0, CS0) |
| PCIE_32 / PCIE_64 | PCI Express interface with 32 bit / 64 bit addressing |


Copyright © 2011–2016, Texas Instruments Incorporated

## 4.6 Fast External Booting

### 4.6.1 Overview

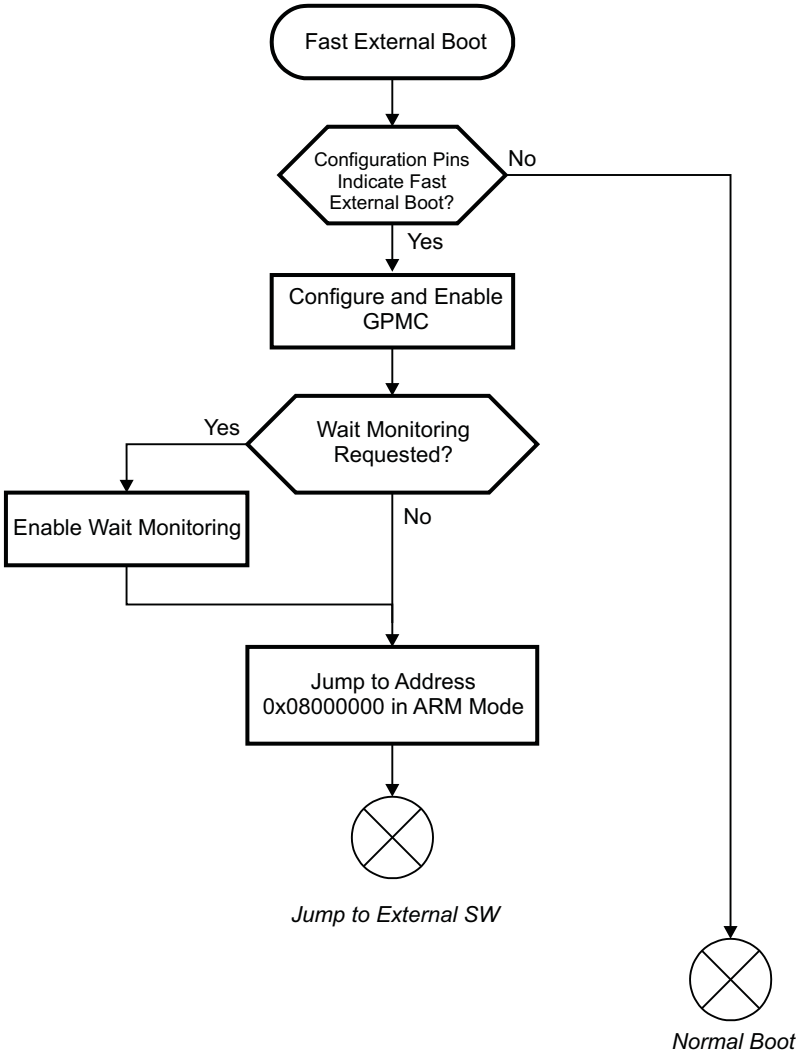
The fast external boot feature:

- Consists of a blind jump in ARM mode to a code located in an external XIP device connected to CS0
- The jump is performed with minimum on-chip ROM Code execution, without configuring any PLL
- Allows the customer to create its own booting code
- Is set up by means of the configuration pins, see Table 4-8

### 4.6.2 External Booting

Figure 4-7 shows the fast external boot procedure. The code does not make use of RAM and is designed for fast execution.

**Figure 4-7. Fast External Boot**





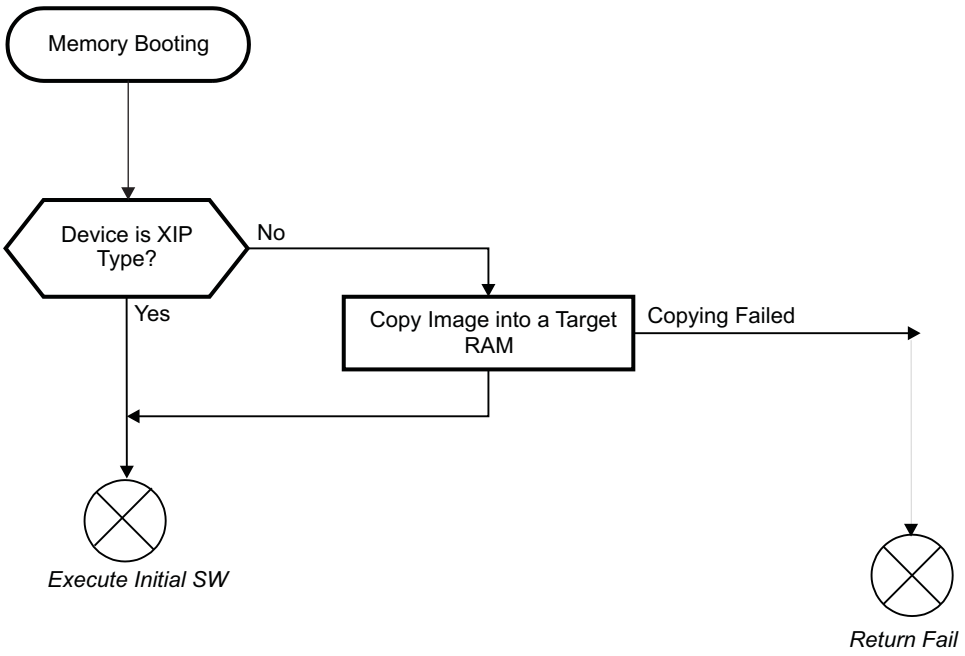

Copyright © 2011–2016, Texas Instruments Incorporated

## 4.7 Memory Booting

### 4.7.1 Overview

The memory booting procedure takes care of starting an external code located in memory device types. These devices are also called permanent booting devices since they are used in that manner.

**Figure 4-8. Memory Booting**




The permanent booting devices supported are:

- MMC/SD cards
- NOR flash
- NAND flash
- SPI EEPROMs

There are two groups of permanent booting devices distinguished by the need of code shadowing. The code shadowing means copying a code from a non-directly addressable device into a location (typically a RAM area) from where the code can be executed. Devices which are directly addressable are called eXecute In Place (XIP) devices.


Copyright © 2011–2016, Texas Instruments Incorporated

The memory booting flowchart is shown in Figure 4-8. The second step is about performing the shadowing of the image that is copying the image from external mass storage (non-XIP) into internal RAM. Failure in image copy results in memory booting returning to the main booting procedure which will select the next device for booting. The next sections detail procedures for device initialization & detection in addition to the description of the sector read routine for each supported device type. A sector is a logical unit of 512 bytes.

A valid image is considered to be present when the first 4 bytes word of the sector is not equal to 0000 0000h or FFFF FFFFh.

During the first read sector call, sectors are copied to a temporary RAM buffer. Once the image is found and a destination address is known, the content of the temporary buffer is moved to the target RAM location so it is needed to re-read the first image sector. The header is discarded; therefore, only executable code is located in RAM with the first executable instruction located at the destination address.

For more information about image formats and contents refer to Section 4.9.

MMC/SD cards and NAND devices can hold up to four copies of the booting image. Therefore the ROM Code searches for one valid image out of the four if present by walking over the first four blocks of the mass storage space. Other XIP devices (NOR) use only one copy of the booting image.

### *4.7.2 XIP Memory*

The ROM Code can boot directly from XIP devices. A typical XIP device is a NOR flash memory. Support for XIP devices is performed under the following assumptions:

- Uses GPMC as the communication interface
- Can connect up to 1 Gbit (128 Mbytes) memories
- Uses both ×8 and ×16 data bus width
- Follows asynchronous protocol
- Supports address / data multiplexed mode and non-multiplexed mode
- GPMC clock is 55 MHz
- Device is connected to CS0 mapped to address 800 0000h
- Wait pin signal WAIT0 is monitored depending on the MBOOT configuration pins (XIP/XIPWAIT).

Depending on the MBOOT option the GPMC is configured to use the WAIT signal connected on the WAIT0 pin or not. Wait pin polarity is set to stall accessing memory when the WAIT0 pin is low. The wait monitoring is intended to be used with memories which require long time for initialization after reset or need to pause while reading data. The boot procedure from XIP device can be described as such:

1. Configure GPMC for XIP device access
2. Set the image location to 800 0000h
3. Verify if bootable image is present at the image location.
4. If the image has been found, start it.
5. If the image has not been found, return from XIP booting to the main booting loop.

#### 4.7.2.1 XIP Initialization and Detection

***GPMC Initialization***

Figure 4-9 and Table 4-9 describes the GPMC timing settings set for XIP boot and other address-data accessible devices (e.g., OneNAND).


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 4-9. GPMC XIP Timings**

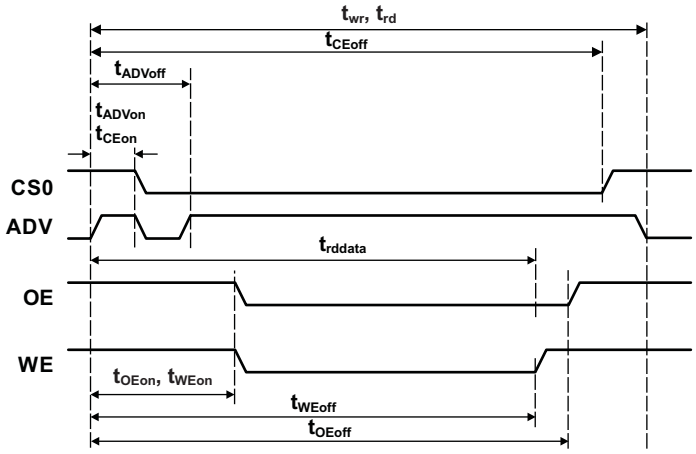



**Table 4-9. XIP Timings Parameters**

| Parameter | Description | Value [clock cycles] |
|---|---|---|
| twr | write cycle period | 17 |
| trd | read cycle period | 17 |
| tCEon | CE low time | 1 |
| tCEoff | CE high time | 16 |
| tADVon | ADV low time | 1 |
| tADVoff | ADV high time | 2 |
| tOEon | OE low time | 3 |
| tWEon | WE low time | 3 |
| trddata | data latch time | 15 |
| tOEoff | OE high time | 16 |
| tWEoff | WE high time | 15 |

The one clock cycle is 18.182 ns which corresponds to 55MHz frequency.

***Device Detection***

There is no specific identification routine executed prior to booting from an XIP device.

### 4.7.2.2 Pins Used for NOR Boot

The list of pins that are configured by the ROM in the case of NOR boot mode are listed in Table 4-10. Note that all the pins might not be driven at boot time. The decision as to which pins need to be driven is based on the type of NOR flash selected.

**Table 4-10. Pins Used for NOR Boot**

| Signal name | Pin used in XIP_MUX0 mode | Pin used in XIP_MUX1 mode |
|---|---|---|
| cs0 | gpmc_cs0 | gpmc_cs0 |
| advn_ale | gpmc_advn_ale | gpmc_advn_ale |
| oen_ren | gpmc_oen_ren | gpmc_oen_ren |
| be0n_cle | gpmc_be0n_cle | gpmc_be0n_cle |
| wen | gpmc_wen | gpmc_wen |
| wait | gpmc_wait | gpmc_wait |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-10. Pins Used for NOR Boot (continued)**

| Signal name | Pin used in XIP_MUX0 mode | Pin used in XIP_MUX1 mode |
|---|---|---|
| clk | gpmc_clk | gpmc_clk |
| ad0 - ad15 | gpmc_ad0 - gpmc_ad15 | gpmc_ad0 - gpmc_ad15 |
| a0 | gmii0_rxd[3] | vout1_b_cb_c[2] |
| a1 | gmii0_rxd[4] | mmc2_dat[3] |
| a2 | gmii0_rxd[5] | mmc2_dat[2] |
| a3 | gmii0_rxd[6] | mmc2_dat[1] |
| a4 | gmii0_rxd[7] | mmc2_dat[0] |
| a5 | gmii0_rxdv | vout1_g_y_yc[1] |
| a6 | gmii0_gtxclk | vout1_g_y_yc[0] |
| a7 | gmii0_txd[0] | vout1_r_cr[1] |
| a8 | gmii0_txd[1] | vout1_r_cr[0] |
| a9 | gmii0_txd[2] | vout1_b_cb_c[1] |
| a10 | gmii0_txd[3] | vout1_b_cb_c[0] |
| a11 | gmii0_txd[4] | vout1_fid |
| a12 | gmii0_txd[5] | vout0_fid |

**Note:** The ROM code does not multiplex high order address lines gpmc_a[27:13] to their address functions, although the external memory device generally needs to see logic "0" on its higher order address bits to correctly address memory. Many of the high order address pads default to internal pull-down active; however, some pins will default to internal pull-up active and therefore will need to be driven or pulled low by external hardware for the duration of XIP boot operation. See the device data manual for specific details.

In PG2.0, if MBOOT[10] = 1b, the following changes come into effect. These override the pin configuration mentioned above.

- GPMC_CLK is not configured
- GPMC_BE0N_CLE is not configured
- GPMC_ADVN_ALE is configured only for multiplexed NOR boot
- GPMC_A0 configured pinmux in case of 8-bit wide data NOR boot ONLY.
- GPMC_WAIT0 is configured for NOR (muxed and Non-muxed) with default settings in case we use as WAIT , else do a PULL DOWN
- For the following Pins a PULL DOWN is done for both Muxed NOR and Non-Muxed NOR devices
  - GPMC_A[20]
  - GPMC_A[22]
  - SD2_DAT[7]
  - SD2_DAT[6]
  - SD2_DAT[5]
  - GPMC_WAIT[0] (UNLESS configured as WAIT per BOOTMODE)
- For the following Pins a PULL DOWN is done for Non-Muxed NOR devices only
  - VOUT[1]_G_Y_YC[2]
  - VOUT[1]_R_CR[3]

#### 4.7.2.3 Sysboot Pins

Some of the SYSBOOT pins have special meanings when NOR boot is selected.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-11. Special SYSBOOT Pins for NOR Boot**

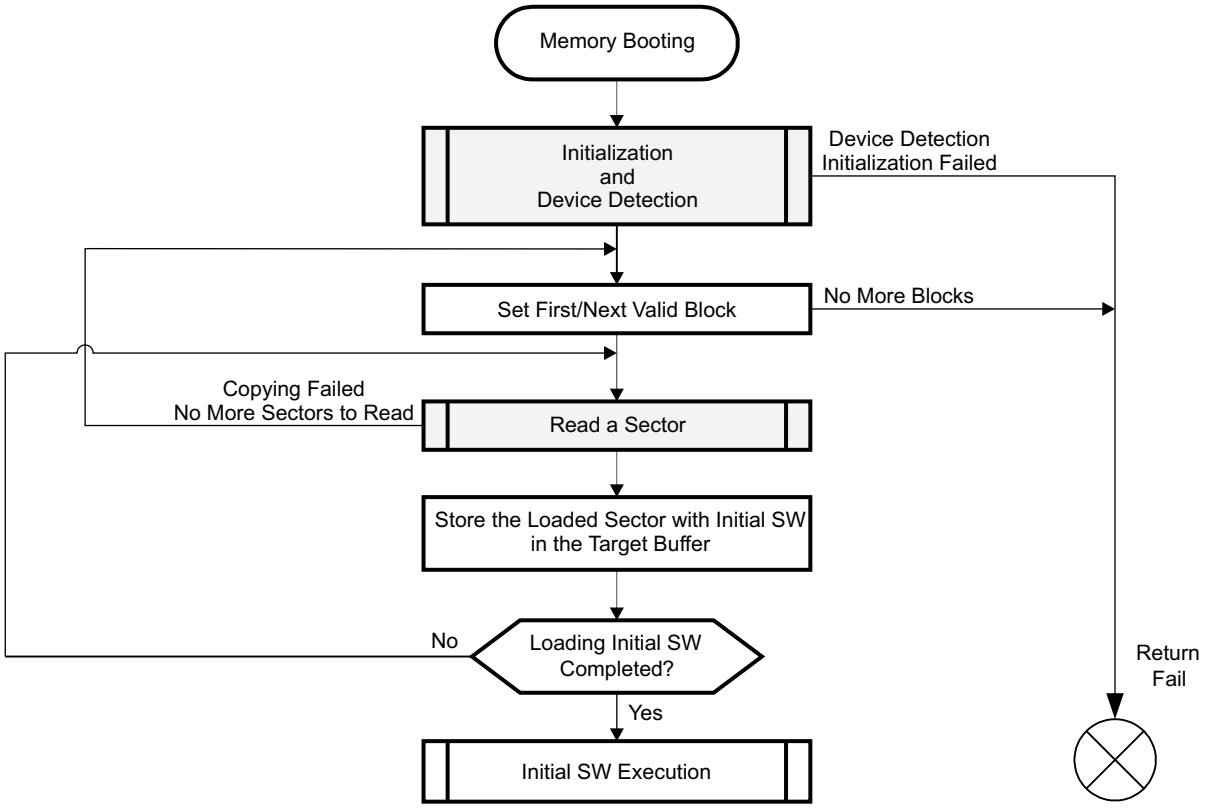
| SYSBOOT[n] | Description |
|---|---|
| [12] (CS0BW) | Must be 0 if 8-bit device is used<br>Must be 1 if 16-bit device is used |
| [15] (CS0WAIT) | Must be 0 if XIP WIAT if not used<br>Must be 1 if XIP WAIT is used |
| [14:13] (CS0MUX[1:0]) | 00b – Non-muxed device<br>01b – A/D muxed device<br>10b – AAD muxed device<br>11b – reserved. |

#### 4.7.2.4 Image Shadowing for non-XIP Memories

**Shadowing on the device**

Shadowing uses the following approach.

**Figure 4-10. Image Shadowing on the Device**





Copyright © 2011–2016, Texas Instruments Incorporated

### 4.7.3 NAND

The NAND flash memory is not XIP and requires shadowing before the code can be executed.

#### 4.7.3.1 NAND Features

The NAND features include:

- GPMC as the communication interface
- Device from 512Mbit (64 MByte)
- ×8 and ×16 bus width
- Support for large page size (2048 bytes + 64 spare bytes) or very large page size 4096 bytes + 128 / 218 spare bytes)
- CE don't care devices only
- Single Level Cell (SLC) and Multiple Level Cell (MLC) devices
- Device Identification based on ONFI or ROM table
- ECC correction : 8 bits/sector for most devices (16b/sector for devices with large spare area)
- GPMC timings adjusted for NAND access
- 55MHz GPMC clock
- Device connected to CS0
- Wait pin signal WAITPIN0 connected to NAND BUSY output
- Four physical blocks are searched for an image. The block size depends on device.

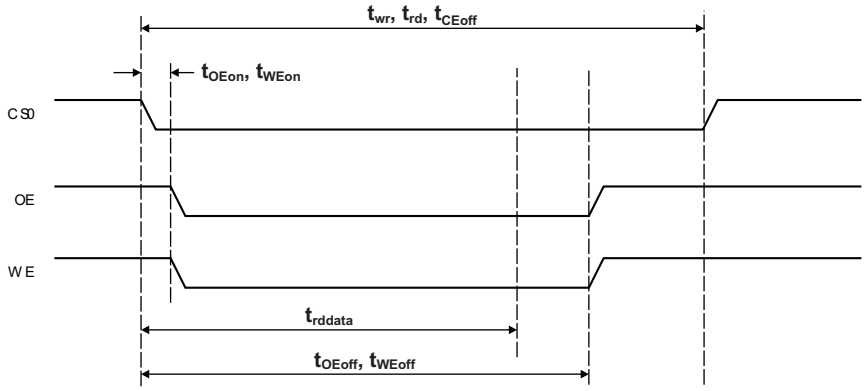
#### 4.7.3.2 Initialization and Detection

The initialization routine for NAND devices consists in three parts: GPMC initialization, device detection with parameters determination and finally bad block detection.

- **ONFI support.** The NAND identification starts with ONFI detection.
- **GPMC initialization.** The GPMC interface is configured such that it can be used for accessing NAND devices. The address bus is released since a NAND device does not use it. The data bus width is initially set to 8 bits; and changed to 16 bits if needed after device parameters determination. The following scheme is applied since NAND devices require different timings when compared to regular NOR devices:

Figure 4-11 and Table 4-12 describes the timings configured for NAND device access. The one clock cycle is 18.181 ns which correspond to 55 MHz frequency.

**Figure 4-11. GPMC NAND Timings**






Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-12. NAND Timings Parameters**

| Parameter | Description | Value [clock cycles] |
|---|---|---|
| twr | write cycle period | 30 |
| trd | read cycle period | 30 |
| tCEon | CE low (not marked on the figure) | 0 |
| tOEon | CE low to OE low time | 7 |
| tWEon | CE low to WE low time | 5 |
| trddata | CE low to data latch time | 21 |
| tOEoff | CE low to OE high time | 24 |
| tWEoff | CE low to WE high time | 22 |

- **Device detection and parameters.** The ROM code first performs an initial wait for device auto initialization (with 250ms timeout) with polling of the ready information. Then, it needs to identify the NAND type connected to the GPMC interface. The GPMC is initialized using 8 bits, asynchronous mode. The NAND device is reset (command FFh) and its status is polled until ready for operation (with 100ms timeout). The ONFI Read ID (command 90h / address 20h) is sent to the NAND device. If it replies with the ONFI signature (4 bytes) then a Read parameters page (command ECh) is sent.
- If the parameters page does not have the ONFI signature, then the ONFI identification fails and the device falls back to using the look up table. If the parameter page contains the ONFI signature, the information shown in Table 4-13 is then extracted: page size, spare area size, number of pages per block, and the addressing mode. The remaining data bytes from the parameters page stream are simply ignored.

**Table 4-13. ONFI Parameters Page Description**

| Offset | Description | Size (bytes) |
|---|---|---|
| 6 | Features supported | 2 |
| 80 | Number of data bytes per page | 4 |
| 84 | Number of spare bytes per page | 2 |
| 92 | Number of pages per block | 4 |
| 101 | Number of address cycles | 1 |

If the ONFI Read ID command fails (it will be the case with any device not supporting ONFI) then the device is reset again with polling for device to be ready (with 100ms timeout). Then, the standard Read ID (command 90h / address 00h) is sent. If the Device ID (2nd byte of the ID byte stream) is recognized as being a supported device, then the device parameters are extracted from an internal ROM Code table. The list of supported devices is shown in Table 4-14.

**Table 4-14. Supported NAND Devices**

| Capacity | Device ID | Bus Width | Page size |
|---|---|---|---|
| 512 Mb | F0 | x8 | 2048 |
| 512 Mb | C0 | x16 | 2048 |
| 512 Mb | A0 | x8 | 2048 |
| 512 Mb | B0 | x16 | 2048 |
| 512 Mb | F2 | x8 | 2048 |
| 512 Mb | C2 | x16 | 2048 |
| 512 Mb | A2 | x8 | 2048 |
| 512 Mb | B2 | x16 | 2048 |
| 1 Gb | F1 | x8 | 2048 |
| 1 Gb | C1 | x16 | 2048 |
| 1 Gb | A1 | x8 | 2048 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-14. Supported NAND Devices (continued)**

| Capacity | Device ID | Bus Width | Page size |
|---|---|---|---|
| 1 Gb | B1 | x16 | 2048 |
| 2 Gb | DA | x8 | 2048 |
| 2 Gb | CA | x16 | 2048 |
| 2 Gb | AA | x8 | 2048 |
| 2 Gb | BA | x16 | 2048 |
| 2 Gb | 83 | x8 | 2048 |
| 2 Gb | 93 | x16 | 2048 |
| 4 Gb | DC | x8 | 2048 |
| 4 Gb | CC | x16 | 2048 |
| 4 Gb | AC | x8 | 2048/4096 |
| 4 Gb | BC | x16 | 2048/4096 |
| 4 Gb | 84 | x8 | 2048 |
| 4 Gb | 94 | x16 | 2048 |
| 8 Gb | D3 | x8 | 2048/4096 |
| 8 Gb | C3 | x16 | 2048/4096 |
| 8 Gb | A3 | x8 | 2048/4096 |
| 8 Gb | B3 | x16 | 2048/4096 |
| 8 Gb | 85 | x8 | 2048 |
| 8 Gb | 95 | x16 | 2048 |
| 16 Gb | D5 | x8 | 2048/4096 |
| 16 Gb | C5 | x16 | 2048/4096 |
| 16 Gb | A5 | x8 | 2048/4096 |
| 16 Gb | B5 | x16 | 2048/4096 |
| 16 Gb | 86 | x8 | 2048 |
| 16 Gb | 96 | x16 | 2048 |
| 32 Gb | D7 | x8 | 2048/4096 |
| 32 Gb | C7 | x16 | 2048/4096 |
| 32 Gb | A7 | x8 | 2048/4096 |
| 32 Gb | B7 | x16 | 2048/4096 |
| 32 Gb | 87 | x8 | 2048 |
| 32 Gb | 97 | x16 | 2048 |
| 64 Gb | DE | x8 | 2048/4096 |
| 64 Gb | CE | x16 | 2048/4096 |
| 64 Gb | AE | x8 | 2048/4096 |
| 64 Gb | BE | x16 | 2048/4096 |

When the parameters are retrieved from the ROM table: page size and block size is updated based on 4th byte of NAND ID data. Due to inconsistency amongst different manufacturers, only devices which has been recognized to be at least 2Gb (included) have these parameters updated. Therefore, the ROM Code supports 4kB page devices but only if their size, according to the table, is at least 2Gb. Devices which are smaller than 2Gb have the block size parameter set to 128kB (when page size is 2KB). Table 4-15 shows the 4th ID Data byte encoding used in ROM Code.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-15. 4th NAND ID Data Byte**

| Item | Description | I/O # | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Page Size | 1kB | | | | | | | 0 | 0 |
| | 2kB | | | | | | | 0 | 1 |
| | 4kB | | | | | | | 1 | 0 |
| | 8kB | | | | | | | 1 | 1 |
| Cell type | 2 levels | | | | | 0 | 0 | | |
| | 4 levels | | | | | 0 | 1 | | |
| | 8 levels | | | | | 1 | 0 | | |
| | 16 levels | | | | | 1 | 1 | | |
| Block Size | 64kB | | | 0 | 0 | | | | |
| | 128kB | | | 0 | 1 | | | | |
| | 256kB | | | 1 | 0 | | | | |
| | 512kB | | | 1 | 1 | | | | |

- **Reading NAND geometry from I2C EEPROM.** ROM supports a special boot mode called NANDI2C to support NAND devices whose geometry cannot be detected by the ROM automatically using methods described in the previous section (Figure 4-12). If this boot mode is selected, the ROM code tries to read NAND geometry from an I2C EEPROM. If the read is successful, ROM code then proceeds to the next steps of NAND boot, beginning with reading bad blocks of information.

  The list of pins that are configured by the ROM in case of NANDI2C boot mode (this is in addition to the NAND boot pins) is described in Table 4-18).

**Table 4-16. Pins used for NANDI2C boot for I2C EEPROM access**

| Signal name | Pin used |
|---|---|
| I2C SCL | i2c0_scl |
| I2C SDA | i2c0_sda |

ROM accesses the I2C EEPROM at I2C slave address 50h and reads 7 bytes starting from address offset 80h. The format of this (NAND geometry information) is as follows:

**Table 4-17. NAND Geometry Information on I2C EEPROM**

| Byte address | Information | |
|---|---|---|
| | Upper nibble | Lower nibble |
| 80h | Magic Number – 10h | |
| 81h | Magic Number – B3h | |
| 82h | Magic Number – 57h | |
| 83h | Magic Number – A6h | |
| 84h | NAND column address (word/byte offset within a page) size in bytes, Example: 2 | NAND row address (page offset) size in bytes. Example: 3 |
| 85h | Page size (2N) exponent “N”. Example (for page size of 2048): 11 | Pages per block (2N) exponent “N” Example (for number of blocks 64): 6 |
| 86h | NAND bus width 0 → 8-bit, 1 → 16-bit | ECC Type 0→ No ECC, 1 → BCH8, 2 → BCH16 |


Copyright © 2011–2016, Texas Instruments Incorporated

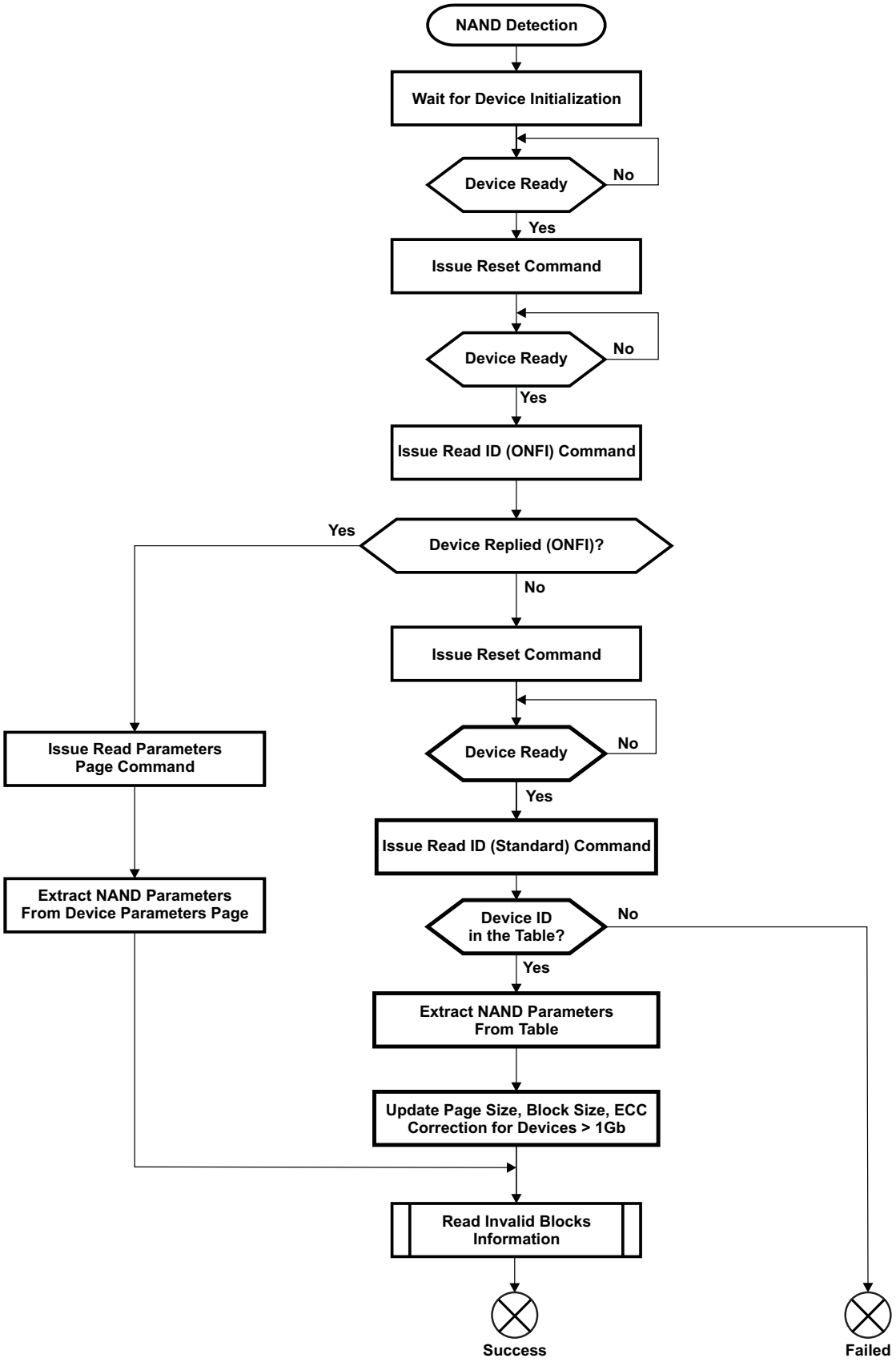
- **ECC correction.** The default ECC correction applied is BCH 8b/sector using the GPMC and ELM hardware.

  For device ID codes D3h, C3h, D5h, C5h, D7h, C7h, DEh, CEh when manufacturer code (first ID byte) is 98h the cell type information is checked in the 4th byte of ID data. If it is equal to 10b then the ECC correction applied is BCH 16b/sector.

The detection procedure is described in Figure 4-12. Once the device has been successfully detected, the ROM code changes GPMC to 16-bit bus width if necessary.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 4-12. NAND Device Detection**



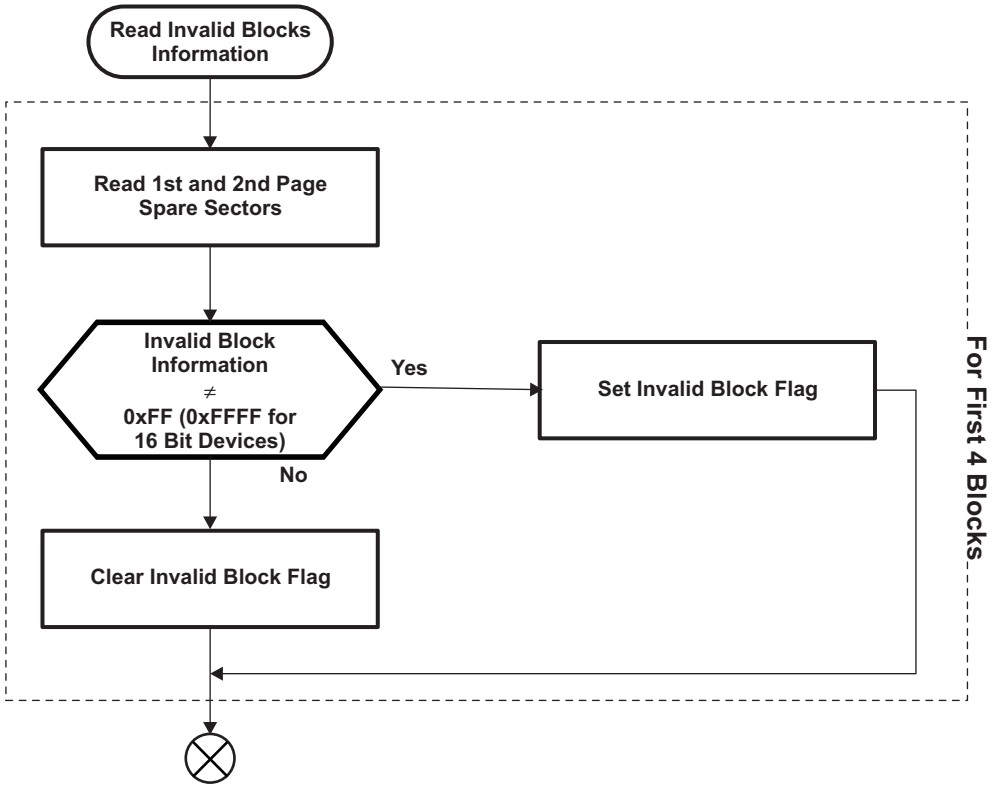


Copyright © 2011–2016, Texas Instruments Incorporated

- **Bad block verification.** Invalid blocks are blocks which contain invalid bits whose reliability cannot be guaranteed by the manufacturer. Those bits are identified in the factory or during the programming and reported in the initial invalid block information located in the spare area on the 1st and 2nd page of each block. Since the ROM Code is looking for an image in the first four blocks, it must detect block validity status of these blocks. Blocks which are detected as invalid are not accessed later on. Blocks validity status is coded in the spare areas of the first two pages of a block (first byte equal to FFh in 1st and 2nd pages for an 8 bits device / first word equal to FFFFh in 1st and 2nd page for a 16-bit device).

  Figure 4-13 depicts the invalid block detection routine. The routine consists in reading spare areas and checking validity data pattern.

**Figure 4-13. NAND Invalid Blocks Detection**





Copyright © 2011–2016, Texas Instruments Incorporated

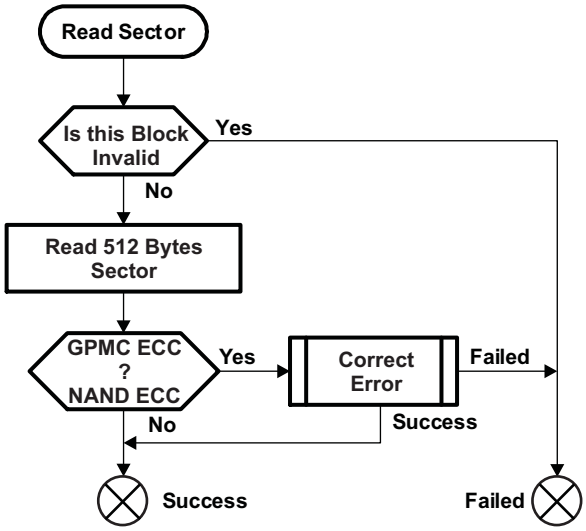
#### 4.7.3.3 NAND Read Sector Procedure

The ROM Code reads data from NAND devices in 512 bytes sectors. The read function fails in two cases:

- The accessed sector is within a block marked as invalid
- The accessed sector contains an error which cannot be corrected with ECC

Figure 4-14 shows the read sector routine for NAND devices. The ROM Code uses normal read (command 00h 30h) for reading NAND page data.

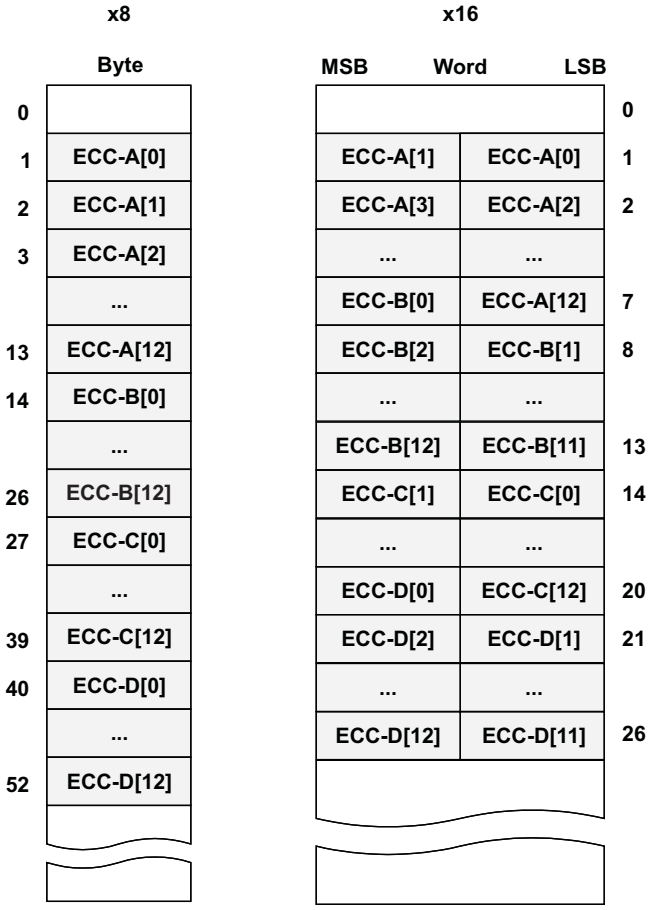
**Figure 4-14. NAND Read Sector Procedure**




Page data can contain errors due to memory alteration. The ROM Code uses an ECC correction algorithm to detect and possibly correct those errors. The ECC algorithm used is BCH with capability for correcting 8b or 16b errors per sector. The BCH data is automatically calculated by the GPMC on reading each 512 bytes sector. The computed ECC is compared against ECC stored in the spare area for the corresponding page. Depending on the page size, the amount of ECC data bytes stored in the corresponding spare area is different. Figure 4-15 and Figure 4-16 show the mapping of ECC data inside the spare area for respectively 2KB-page and 4KB- page devices. If both ECC data are equal then the Read Sector function returns the read 512 bytes sector without error. Otherwise, the ROM Code tries to correct error(s) in the corresponding sector (this procedure is assisted by the ELM hardware) and returns the data if successful. If errors are uncorrectable, the function returns with FAIL.


Copyright © 2011–2016, Texas Instruments Incorporated

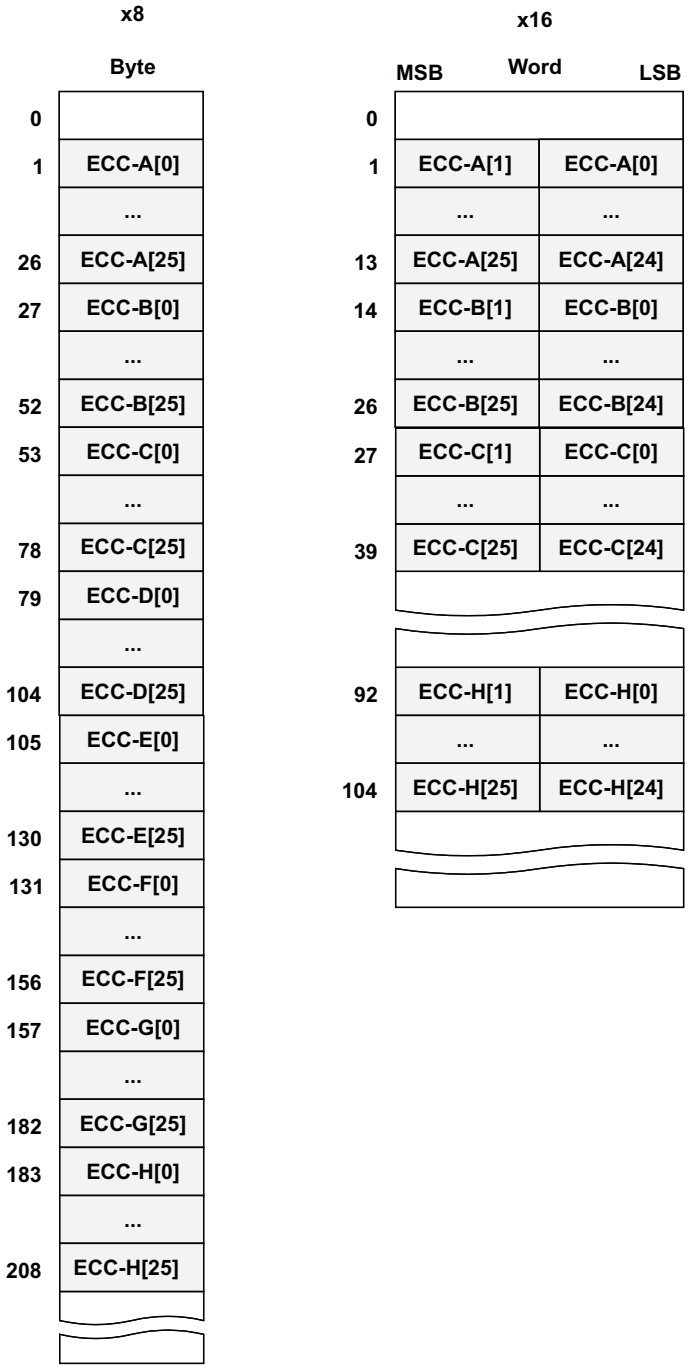
**Figure 4-15. ECC Data Mapping for 2KB Page and 8b BCH Encoding**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 4-16. ECC Data Mapping for 4KB Page and 16b BCH Encoding**




### 4.7.3.4 Pins Used for NAND Boot

The list of pins that are configured by the ROM in case of NAND boot mode are in Table 4-18. Note that all the pins might not be driven at boot time.

**Table 4-18. Pins Used for NAND Boot**

| Signal name | Pin used |
|---|---|
| cs0 | gpmc_cs0 |
| advn_ale | gpmc_advn_ale |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-18. Pins Used for NAND Boot (continued)**

| Signal name | Pin used |
|---|---|
| oen_ren | gpmc_oen_ren |
| be0n_cle | gpmc_be0n_cle |
| wen | gpmc_wen |
| wait | gpmc_wait |
| clk | gpmc_clk(1) |
| ad0 - ad15 | gpmc_ad0 - gpmc_ad15 |

(1) gpmc_clk is not configured in MBOOT[10] = 1b

#### 4.7.3.5 Sysboot Pins for NAND Boot

Some of the SYSBOOT pins have special meanings when NAND boot is selected.

**Table 4-19. Special SYSBOOT pins for NAND boot**

| SYSBOOT[n] | Description |
|---|---|
| [12] (CS0BW) | Must be 0 if 8-bit NAND is used |
| | Must be 1 if 16-bit NAND is used |
| [15] (CS0WAIT) | Must be 0 |
| [14:13] (CS0MUX[1:0]) | Must be 0 |

### *4.7.4 MMC/SD Cards*

#### 4.7.4.1 Overview

The ROM code supports booting from MMC/SD cards in the following conditions:

- MMC/SD Cards compliant to low and high capacities
- MMC/SD cards connected to MMC interface #1
- 3V VCC power supply, support for 3V or 1.8V I/O voltages
- Initial 1-bit MMC Mode, optional 4-bits or 8-bits mode
- Clock Frequency: identification mode: 400 KHz; data transfer mode up to 10 MHz
- Only one card connected to the bus
- Raw mode, image data read directly from sectors in the user area
- File system mode (FAT12/16/32 supported with or without Master Boot Record), image data is read from a booting file

#### 4.7.4.2 System Interconnection

An MMC/SD card can be connected to the MMC1 interface typically through a card cage.

Notes:

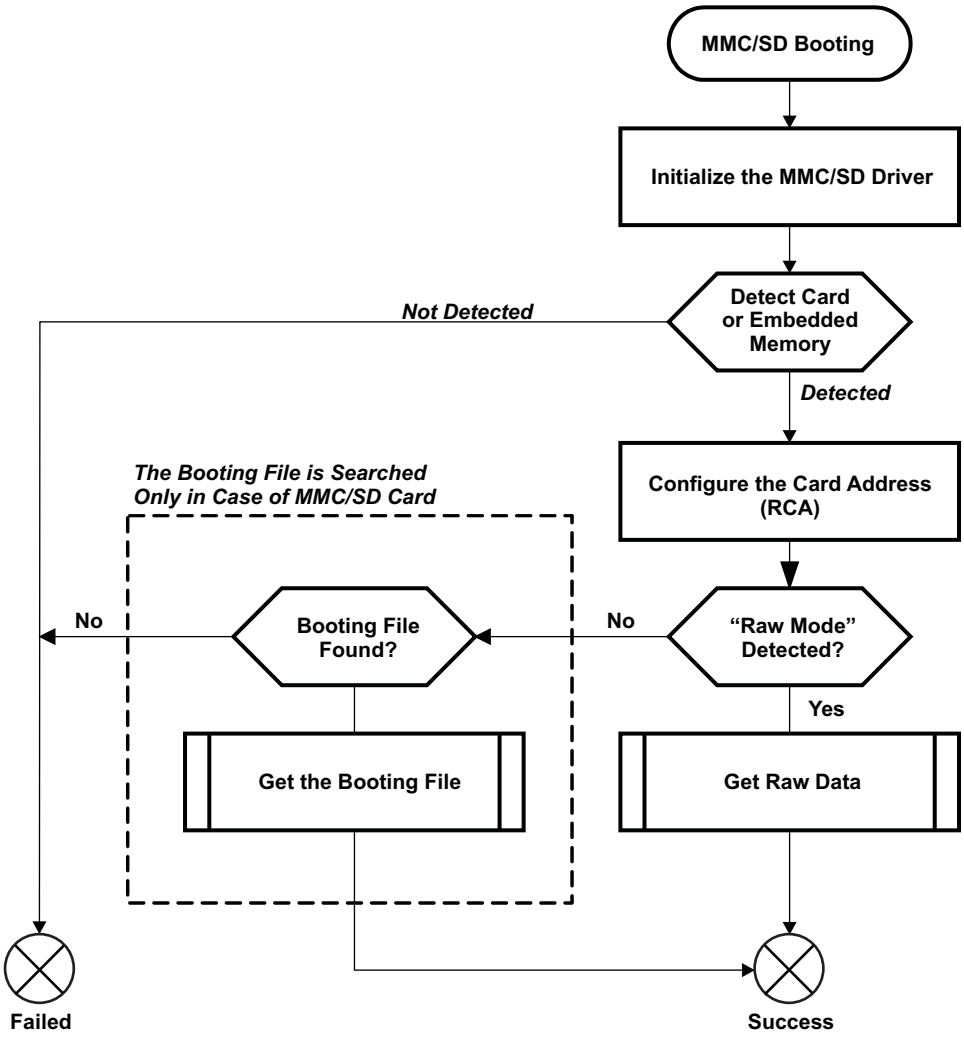
- It may be possible to design the system so that an eMMC/eSD memory type is connected to the MMC1 interface. However, the dual voltage IOs feature of MMC1 dedicates it preferably to cards.
- The ROM Code does not handle the card detection feature on card cage.

#### 4.7.4.3 Booting Procedure

The high level flowchart of the eMMC/eSD and MMC/SD booting procedure is depicted in Figure 4-17. The booting file is searched only in case of booting from a card. eMMC/eSD embedded memories only support raw mode.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 4-17. MMC/SD Booting**




### 4.7.4.4 Initialization and Detection

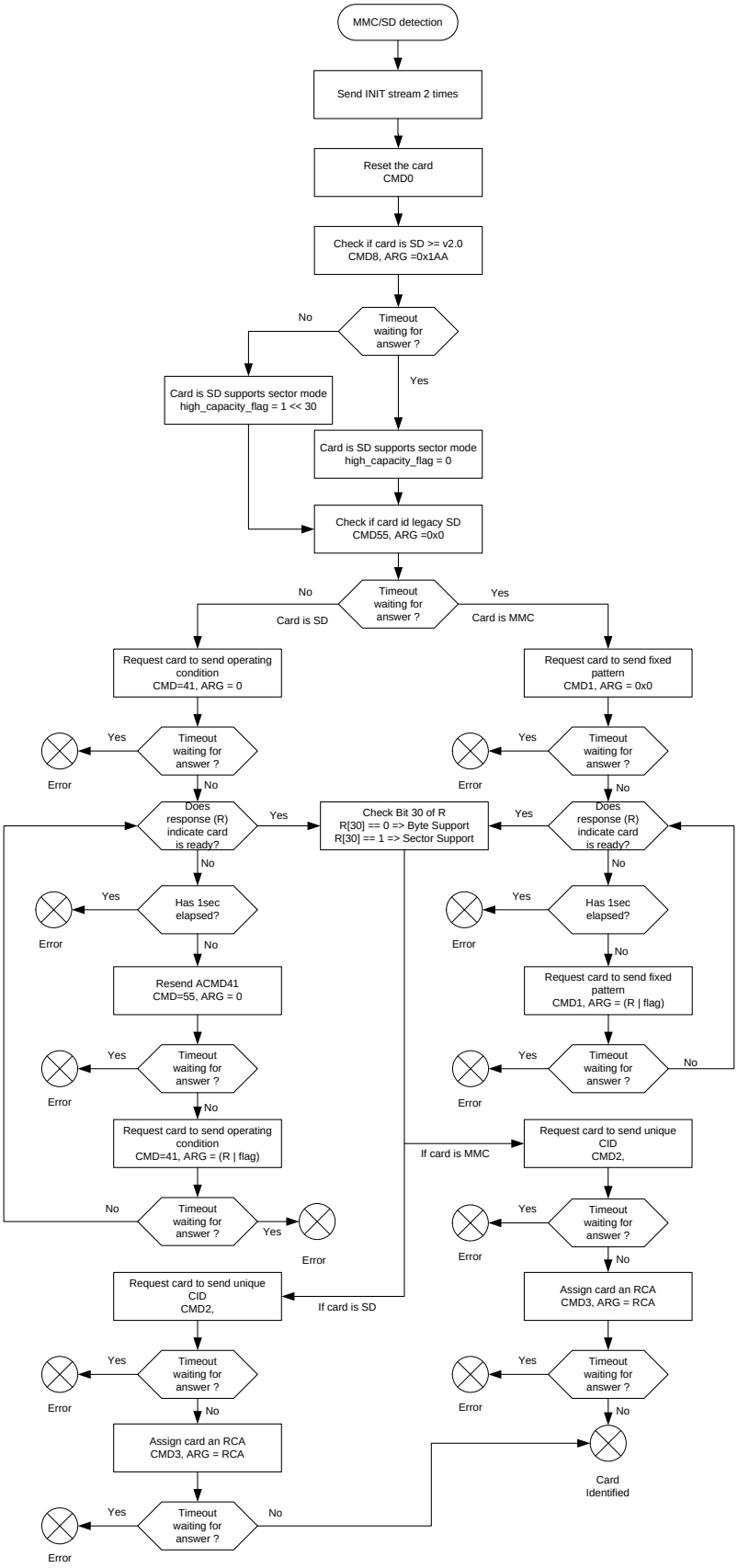
The ROM Code initializes the memory device or card connected on MMC interface using the standard High-Voltage range (3.0V). If neither memory device nor card is detected then the ROM code carries on to the next booting device. The standard identification process and relative card address (RCA) assignment are used. However, the ROM Code assumes that only one memory or card is present on the bus. This first sequence is done using the CMD signal that is common to SD and MMC devices.

MMC and SD standards detail this phase as initialization phase. Both standards differ in the first commands involved: CMD1 and ACMD41. The ROM code uses this difference in command set to discriminate between MMC and SD devices: CMD1 is supported only by the MMC standard whereas ACMD41 is only supported by SD standard. The ROM code first sends a CMD1 to the device and gets a response only if an MMC device is connected. If no response is received then ACMD41 (ACMD41 is made out of CMD55 and ACMD41) is sent and a response is expected from an SD device. If no response is received then it is assumed that no device is connected and the ROM Code exits the MMC/SD Booting procedure with FAIL. This is detection procedure is shown in Figure 4-18.

As previously mentioned, the contents of an MMC/D card may be formatted as raw binary or within a FAT filesystem. eMMC/SD devices only support raw mode. The ROM Code reads out raw sectors from image or the booting file within the file system and boots from it.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 4-18. MMC/SD Detection Procedure**






Copyright © 2011–2016, Texas Instruments Incorporated

As previously mentioned, the contents of an MMC/D card may be formatted as raw binary or within a FAT filesystem. eMMC/SD devices only support raw mode. The ROM Code reads out raw sectors from image or the booting file within the file system and boots from it.

#### 4.7.4.5 MMC/SD Read Sector Procedure in Raw Mode

In raw mode the booting image can be located at one of the four consecutive locations in the main area: offset 0/20000h (128KB)/40000h (256KB)/60000h (384KB). For this reason a booting image shall not exceed 128KB in size. However it is possible to flash a device with an image greater than 128KB starting at one of the aforementioned locations. Therefore the ROM code does not check the image size. The only drawback is that the image will cross the subsequent image boundary.

The raw mode is detected by reading sectors #0, #256, #512, #768. The content of these sectors is then verified for presence of a TOC structure as described in Figure 4-26. A configuration header (CH) must be located in the first sector followed by a header. The CH might be void (only containing a CHSETTINGS item for which the Valid field is zero).

#### 4.7.4.6 MMC/SD Read Sector Procedure in FAT Mode

MMC/SD cards may hold a FAT file system which ROM Code is able to read and process. The image used by the booting procedure is taken from a specific booting file named “MLO.” This file has to be located in the root directory on an active primary partition of type FAT12/16 or FAT32.

An MMC/SD card can be configured either as floppy-like or hard-drive-like.

- When acting as floppy-like, the content of the card is a single file system without any master boot record (MBR) holding a partition table
- When acting as hard-drive-like, an MBR is present in the first sector of the card. This MBR holds a table of partitions, one of which must be FAT12/16/32, primary and active.

The card should always hold an MBR except for MMC cards using floppy-like file system (refer to the CSD internal Register fields FILE_FORMAT_GRP and FILE_FORMAT). However, depending on the operating system, the MMC/SD card will be formatted either with partition(s) (using an MBR) or without. The ROM code supports both types; this is described in the following section.

The ROM code retrieves a map of the booting file from the FAT table. The booting file map is a collection of all FAT table entries related to the booting file (a FAT entry points to a cluster holding part of the file). The booting procedure uses this map to access any 512 byte sector within the booting file without involving ROM code FAT module.

The sector read procedure utilizes standard MMC/SD raw data read function. The sector address is generated based on the booting memory file map collected during the initialization. Hence the ROM Code can address sectors freely within the booting file space.

#### 4.7.4.7 FAT File System

This paragraph describes functionalities which are used by the ROM Code. It is not intended to fully describe the Master Boot Record and the FAT file system, but does include:

- How to recognize if a sector is the 1st sector of an MBR
- How to recognize if a sector is the 1st sector of a FAT12/16/32
- How to find the 1st cluster of the booting file
- How to buffer the booting file FAT entries

Some memory devices which support file systems can be formatted with or without MBR, therefore the first task of the ROM Code is to detect whether or not the device is holding an MBR in the first sector. If this is the case, an active FAT12/16/32 partition is searched in all four MBR partition entries, based on the Type field. If the MBR entries are not valid or if no useable partition is found then the ROM Code returns to the Booting procedure with FAIL. The Extended partitions are not checked, the booting file must reside in a primary partition.


Copyright © 2011–2016, Texas Instruments Incorporated

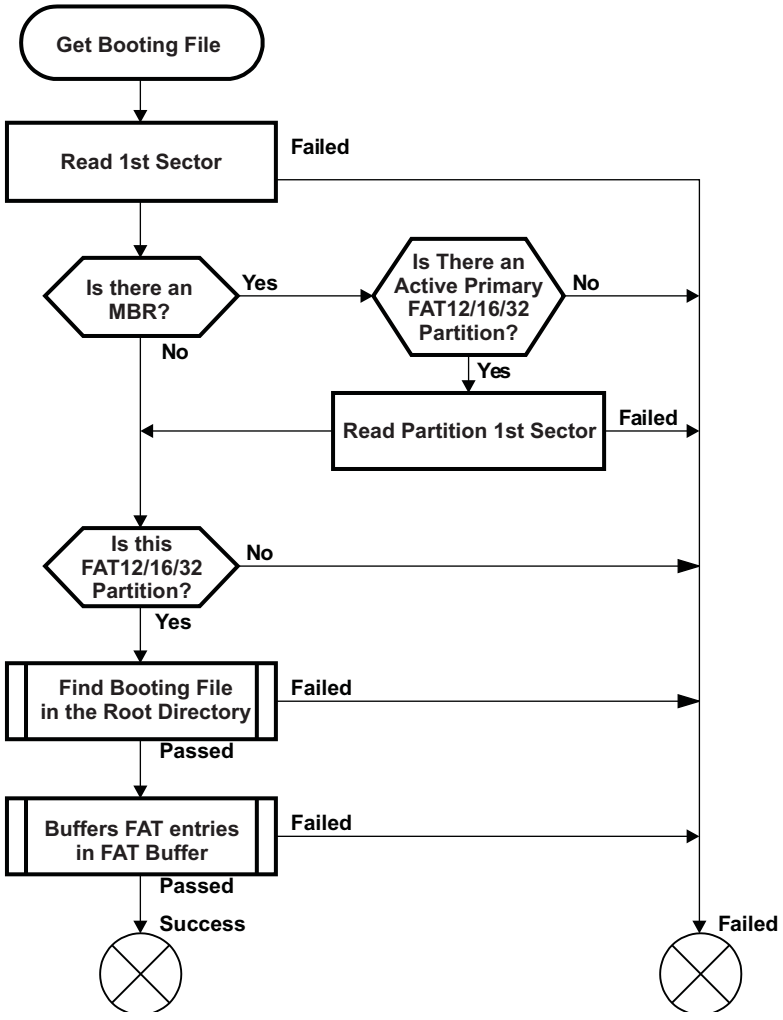
If a partition is found then its first sector is read and used further on. If no MBR is present (in case of a floppy-like system), the first sector of the device is read and used further on. The read sector is checked to be a valid FAT12/16 or FAT32 partition. If this fails, in case another partition type is used (i.e., Linux FS or any other) or if the partition is not valid, the ROM Code returns with FAIL.

Otherwise, the Root Directory entries are searched for a file named depending on the booting device. The Long File Names (LFN) format is not used and only File Names in 8.3 Format are searched for. If no valid file is found, the ROM Code returns with FAIL.

Once the file has been found, the ROM Code reads the File Allocation Table (FAT) and buffers the singly-linked chain of clusters in a FAT Buffer which will be used by the Booting Procedure to access the file directly sector by sector. For FAT12/16 and for FAT32 (valid if a specific flag has been set in the FAT32 Boot Sector), there exist multiples copies of the FAT (ROM Code supports only two copies). When buffering FAT entries, the two FATs are compared. If they are not the same, only entries from the last FAT are used. The FAT Buffer holds sector numbers and not cluster numbers. The ROM Code converts each cluster entry to one or several sector entries if applicable.

The whole process is described in Figure 4-19. Every part related to MBR or FAT12/16/32 is described in the next paragraph.

**Figure 4-19. MMC/SD Booting, Get Booting File**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 4.7.4.7.1 Master Boot Record: MBR

The Master boot record is the 1st sector of a memory device. It is made out of some executable code and four partition entries. The aim of such a structure is to be able to divide the hard disk in partitions mostly used to boot different systems (Microsoft Windows™, Linux, …). Its structure is described in Table 4-20 and Table 4-21. The valid partition types searched by the ROM Code are described in Table 4-22.

**Table 4-20. Master Boot Record Structure**

| Offset | Length [bytes] | Entry Description | Value |
|---|---|---|---|
| 0000h | 446 | Optional Code | |
| 01BEh | 16 | Partition Table Entry | (see ) |
| 01CEh | 16 | Partition Table Entry | (see ) |
| 01DEh | 16 | Partition Table Entry | (see ) |
| 01EEh | 16 | Partition Table Entry | (see ) |
| 01FEh | 2 | Signature | AA55h |

**Table 4-21. Partition Entry**

| Offset | Length [bytes] | Entry Description | Value |
|---|---|---|---|
| 0000h | 1 | Partition State | 00h: Inactive<br>80h: Active |
| 0001h | 1 | Partition Start Head | $H_s$ |
| 0002h | 2 | Partition Start Cylinder and Sector | $C_s$[7:0]-$C_s$[9:8]-$S_s$[5:0] |
| 0004h | 1 | Partition Type | Refer to for partial partition types |
| 0005h | 16 | Partition End Head | $H_e$ |
| 0006h | 2 | Partition End Cylinder and Sector | $C_e$[7:0] - $C_e$[9:8] - $S_e$[5:0] |
| 0008h | 4 | First sector position relative to the beginning of media | $LBA_s = C_s.H.S + H_s.S + S_s - 1$ |
| 000Ch | 4 | Number of sectors in partition | $LBA_e = C_e.H.S + H_e.S + S_e - 1$<br>$Nb_s = LBA_e - LBA_s + 1$ |

**Table 4-22. Partition Types**

| Partition Type | Description |
|---|---|
| 01h | FAT12 |
| 04h, 06h, 0Eh | FAT16 |
| 0Bh, 0Ch, 0Fh | FAT32 |

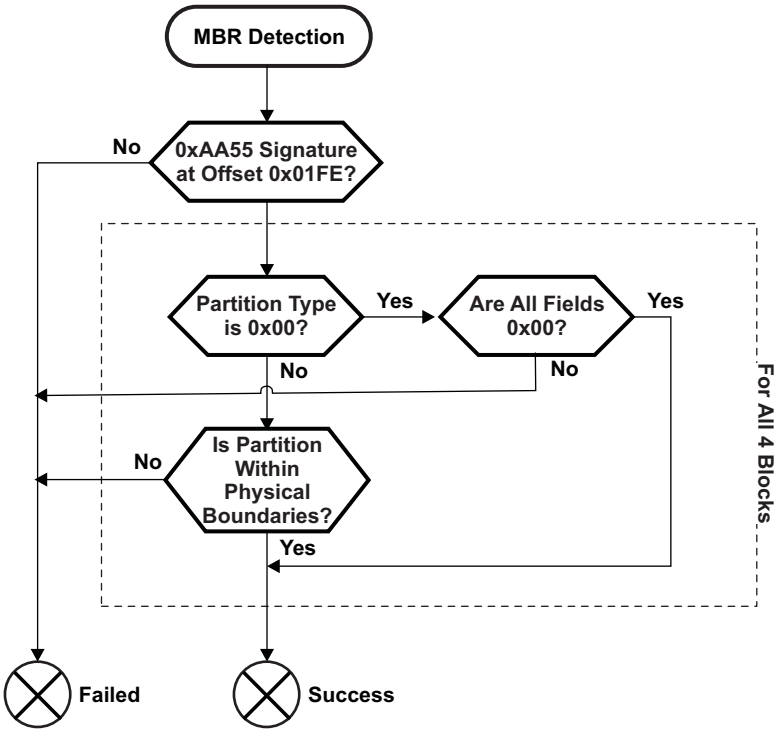

Copyright © 2011–2016, Texas Instruments Incorporated

The way the ROM Code detects whether a sector is the 1st sector of an MBR or not is described in Figure 4-20.

The ROM Code first checks if the signature is present. Each partition entry is checked:

- If its type is set to 00h then all fields in the entry must be 00h
- The partition is checked to be within physical boundaries, i.e. the partition is located inside and its size fits the total physical sectors.

**Figure 4-20. MBR Detection Procedure**

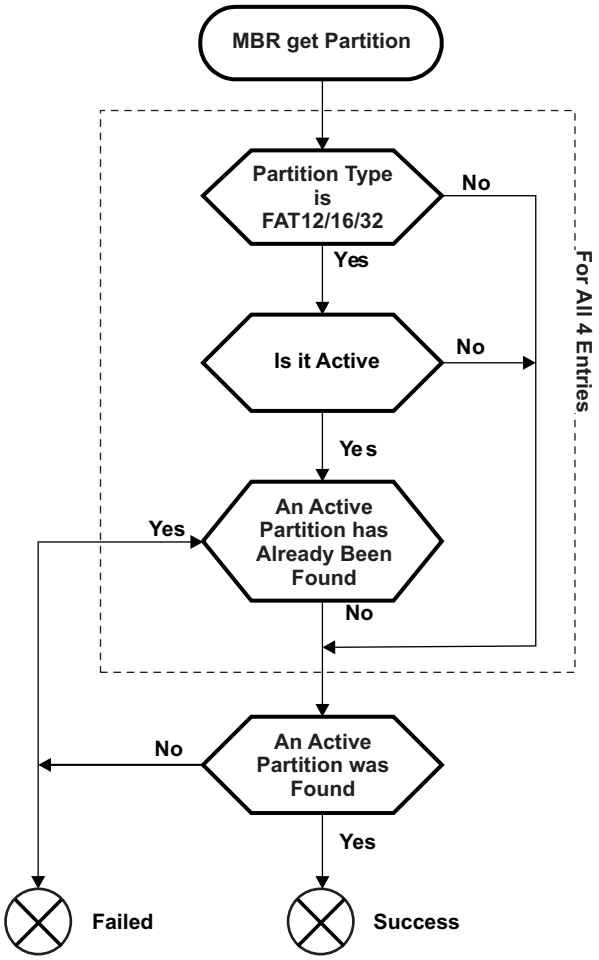





Copyright © 2011–2016, Texas Instruments Incorporated

Once identified, the ROM Code gets the partition using the procedure described in Figure 4-21. The partition type is checked to be FAT12/16 or FAT32. Its state must be 00h or 80h (if there is more than one active partition the test fails). The ROM Code returns with FAIL if no active primary FAT12/16/32 could be found.

**Figure 4-21. MBR, Get Partition**




#### 4.7.4.7.2 FAT12/16/32 Boot Sector

The FAT file system is made out of several parts:

- Boot Sector which holds the BIOS Parameter Block (BPB)
- File Allocation Table (FAT) which describes the use of each cluster of the partition
- Data Area which holds the Files, Directories and Root Directory (for FAT12/16, the Root Directory has a specific fixed location)

The boot sector is described in Table 4-23. In the following description, all the fields whose names start with BPB_ are part of the BPB. All the fields whose names start with BS_ are part of the Boot Sector and not really part of the BPB (not mandatory); they are not used at all by the ROM Code.

**Table 4-23. FAT Boot Sector**

| | Offset | Length [bytes] | Name | Description |
|---|---|---|---|---|
| | 0000h | 3 | BS_jmpBoot | Jump Instruction to Boot Code (not used) |
| | 0003h | 8 | BS_OEMName | Name of the System which created the partition |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-23. FAT Boot Sector (continued)**

| | Offset | Length [bytes] | Name | Description |
|---|---|---|---|---|
| | 000Bh | 2 | BPB_BytsPerSec | Counts of Bytes per sector (usually 512) |
| | 000Dh | 1 | BPB_SecPerClus | Number of sectors per allocation unit |
| | 000Eh | 2 | BPB_RsvdSecCnt | Number of reserved sectors for the Boot Sector<br>For FAT12/16 is 1, for FAT32, usually 32 |
| | 0010h | 1 | BPB_NumFATs | Number of copies of FAT, usually 2 |
| | 0011h | 2 | BPB_RootEntCnt | For FAT12/16, number of 32 bytes entries in the Root Directory (multiple of BPB_BytsPerSec/32)<br>For FAT32 this value is 0 |
| | 0013h | 2 | BPB_TotSec16 | Total Count of sectors on the volume. If the size is bigger than 10000h or for FAT32, this field is 0 and BPB_TotSec32 holds the value |
| | 0015h | 1 | BPB_Media | Media Type, usually F8h: fixed, non-removable |
| | 0016h | 2 | BPB_FATSz16 | For FAT12/16, size in sectors of one FAT<br>For FAT32, holds 0 |
| | 0018h | 2 | BPB_SecPerTrk | Number of sectors per track, 63 for MMC/SD |
| | 001Ah | 2 | BPB_NumHeads | Number of heads, 255 for MMC/SD |
| | 001Ch | 4 | BPB_HiddSec | Number of sectors preceding the partition |
| | 0020h | 4 | BPB_TotSec32 | Total Count of sectors on the volume. If the size is smaller than 10000h (for FAT12/16), this field is 0 and BPB_TotSec16 is valid |
| **FAT12/16** | 0024h | 1 | BS_DrvNum | Drive Number |
| | 0025h | 1 | BS_Reserved1 | 00h |
| | 0026h | 1 | BS_BootSig | Extended Boot Signature 29h. Indicates that the following 3 fields are present |
| | 0027h | 4 | BS_VolID | Volume Serial Number |
| | 002Bh | 11 | BS_VolLab | Volume Label |
| | 0036h | 8 | BS_FilSysType | File system Type: “FAT12”, “FAT16”, “FAT32”.<br>Note: This field is not mandatory (i.e., BS_) therefore it cannot be used to indentify the partition type. |
| **FAT32** | 0024h | 4 | BPB_FATSz32 | Size in sectors of one FAT. Field BPB_FATSz16 must be 0 |
| | 0028h | 2 | BPB_ExtFlags | FAT Flags:<br>[7]: 0 = FAT is mirrored; 1 = Only one FAT is used<br>[3:0]: Number of used FAT if no mirroring used |
| | 002Ah | 2 | BPB_FSVer | File system Version Number |
| | 002Ch | 4 | BPB_RootClus | First Cluster number of the Root Directory |
| | 0030h | 2 | BPB_FSInfo | Sector number of FSINFO Structure in the reserved-area, usually 1 |
| | 0032h | 2 | BPB_BkBootSec | If non-zero, indicates the sector number in the reserved-area of a copy of the Boot Sector |
| | 0034h | 12 | BPB_Reserved | Reserved, set to 00h |
| | 0040h | 1 | BS_DrvNum | Drive Number |
| | 0041h | 1 | BS_Reserved1 | 00h |
| | 0042h | 1 | BS_BootSig | Extended Boot Signature 29h. Indicates that the following 3 fields are present |
| | 0043h | 4 | BS_VolID | Volume Serial Number |
| | 0047h | 11 | BS_VolLab | Volume Label |
| | 0052h | 8 | BS_FilSysType | File system Type: “FAT12”, “FAT16”, “FAT32”.<br>Note: This field is not mandatory (i.e., BS_) therefore it cannot be used to indentify the partition type. |
| | 01FEh | 2 | BPB_Signature | AA55h |


Copyright © 2011–2016, Texas Instruments Incorporated

To check whether or not a sector holds a valid FAT12/16/32 partition, only fields starting with BPB can be checked as they are mandatory. The fields starting from offset 0024h to 01FDh cannot be used for the check as they will differ if using FAT12/16 or FAT32. The procedure is described in Figure 22. First the ROM Code checks if the BPB_Signature is equal to AA55h. Then it checks some fields which must have some specific values (BPB_BytsPerSec, BPB_SecPerClus, BPB_RsvdSecCnt, BPB_NumFATs, BPB_RootEntCnt). If the geometry of the device is known then it is compared against BPB_SecPerTrk and BPB_NumHeads fields. If an MBR was found before, the partition size is also checked.

$$BPB_TotSec16 = MBR_Partition_Size$$

*or*

$$BPB_TotSec32 = MBR_Partition_Size$$

The field BPB_ToSec16 is used if the total number of sectors is below 65518 (in this case BPB_TotSec32=0), otherwise, BPB_TotSec32 is used (BPB_TotSec16=0). The partition sector offset is also checked: BPB_HiddSec = MBR_Partition_Offset (if this value is not 0 as some operating systems do not update this field correctly). The last step is to decide which type of FAT file system it is. The ROM Code computes the number of clusters in the Data Area part of the partition:

$$Nb_{clusters} < 4085 \Rightarrow \text{FAT12}$$

$$4085 \le Nb_{clusters} < 65525 \Rightarrow \text{FAT16}$$

$$65525 \le Nb_{clusters} \Rightarrow \text{FAT32}$$

Where Nbclusters is given by the size of the Data Area:

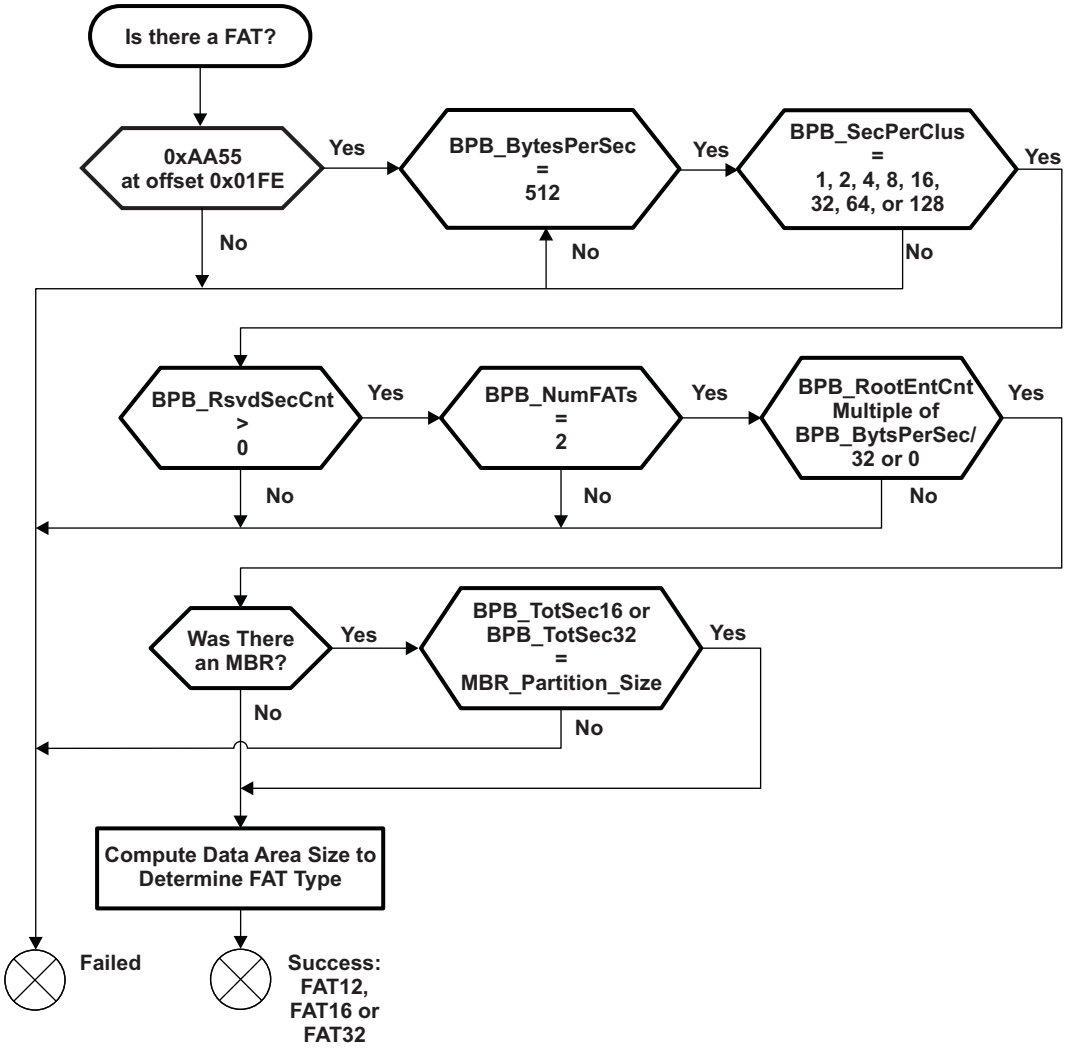
$$RootDirSectors = \frac{BPB_RootEntCnt * 32}{BPB_BytesPerSec}$$

$$DataSec = BPB_TotSec - (BPB_\text{Re}svdSecCnt + (BPB_NumFATs.BPB_FATSz) + RootDirSectors)$$

$$Nb_{clusters} = \frac{DataSec}{BPB_SecPerClus}$$


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 4-22. FAT Detection Procedure**




### 4.7.4.8 FAT12/16/32 Root Directory

The next task for the ROM Code is to find the booting file named “MLO” inside the Root Directory of the FAT12/16/32 file system. The file is not searched in any other location. For a FAT12/16 file system, the Root Directory has a fixed location which is cluster 0. For a FAT32 file system, its cluster location is given by BPB_RootClus. The general formulae to find the sector number (relative to device sector 0, not partition sector 0) of a cluster is given by:

$$Cluster_{sector} = BPB_HiddSec + BPB_RsvdSecCnt + BPB_NumFATs \cdot BPB_FATSz + Cluster \cdot BPB_SecPerClus$$

Note: BPB_FatSz is BPB_FatSz16 for FAT12/16 or BPB_FatSz32 for FAT32

Note: The BPB_HiddSec field can contain 0 even though the FAT file system is located somewhere other than on sector 0 (floppy-like). The ROM Code actually uses the partition offset taken from the MBR instead of this field which can be wrong. If no MBR was found (floppy-like) the value 0 is used.

Each entry in the Root Directory is 32 bytes long and hold information about the file, i.e., filename, date of creation, rights, cluster location etc. This is described in Table 4-24.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-24. FAT Directory Entry**

| Offset | Length | Name | Description | |
|---|---|---|---|---|
| | [bytes] | | | |
| 0000h | 11 | DIR_Name | Short Name (8+3) | |
| 000Bh | 1 | DIR_Attr | File Attributes: | |
| | | | ATTR_READ_ONLY | 01h |
| | | | ATTR_HIDDEN | 02h |
| | | | ATTR_SYSTEM | 04h |
| | | | ATTR_VOLUME_ID | 08h |
| | | | ATTR_DIRECTORY | 10h |
| | | | ATTR_ARCHIVE | 20h |
| | | | ATTR_LONG_NAME | ATTR_READ_ONLY \| ATTR_HIDDEN \| ATTR_SYSTEM \| ATTR_VOLUME_ID |
| 000Ch | 1 | DIR_NTRes | Reserved, set to 00h | |
| 000Dh | 1 | DIR_CrtTimeTenth | Millisecond stamp at file creation | |
| 000Eh | 2 | DIR_CrtTime | Time file was created | |
| 0010h | 2 | DIR_CrtDate | Date file was created | |
| 0012h | 2 | DIR_LstAccDate | Last Access date | |
| 0014h | 2 | DIR_FstClusHi | High word of this entry's first cluster number | |
| 0016h | 2 | DIR_WrtTime | Time of last write | |
| 0018h | 2 | DIR_WrtDate | Date of last write | |
| 001Ah | 2 | DIR_FstClusLo | Low word of this entry's first cluster number | |
| 001Ch | 4 | DIR_FileSize | File size in bytes | |

The ROM Code checks each entry in the Root Directory until either the booting file is found or the entry is empty (first byte is 00h) or when the end of the Root Directory has been reached. Entries with ATTR_LONG_NAME attribute (LFN) and with first byte at E5h (erased file) are ignored. When found, the first cluster offset of the file is read from the DIR_FstClusHi/DIR_FstClusLo fields.

There is a slight difference between FAT12/16 and FAT32 when handling the Root Directory. On FAT12/16, this directory has a fixed location (see above) and length fixed by BPB_RootEntCnt which is the total number of 32 bytes entries. Handling this directory is therefore straight forward. On FAT32, the Root Directory is like a standard file, the File Allocation Table (FAT) has to be used in order to retrieve each sector of the Directory. The way the FAT is handled is described in the next paragraph.

#### 4.7.4.9 FAT12/16/32 File Allocation Table

The ROM Code needs to read the FAT in order to retrieve sectors either for the booting file or for the Root Directory (in case the file system is FAT32).

There can be multiple copies of the FAT inside the file system (ROM Code supports only 2) located after the Boot Sector:

$$FATn_{sector} = BPB_HiddSec + BPB_RsvdSecCnt + BPB_FatSz \cdot n$$

Its size is given by BPB_FATSz16 or BPB_FATSz32. The ROM Code checks each copy of the FAT if identical. In case the values are different, the ROM Code uses the value from the last FAT copy. With FAT32 file system, the copy system can be disabled according to a flag located in BPB_ExtFlags[7]. If this flag is set then FAT BPB_ExtFlags[3:0] is used. In this case no verification is made by the ROM Code with other copies of FAT.


Copyright © 2011–2016, Texas Instruments Incorporated

The FAT is a simple array of values each refering to a cluster located in the Data Area. One entry of the array is 12, 16 or 32 bits depending on the file system in use. The value inside an entry defines whether the cluster is being used or not and if another cluster has to be taken into account. This creates a singly-linked chain of clusters defining the file. The meaning of an entry is described in Table 4-25.

Note: : For compatibility reasons, clusters 0 and 1 are not used for files and those entries must contain FF8h and FFFh (for FAT12); FFF8h and FFFFh (for FAT16); FFFFFF8h and FFFFFFFh (for FAT32).

**Table 4-25. FAT Entry Description**

| FAT12 | FAT16 | FAT32 | Description |
|---|---|---|---|
| 000h | 0000h | ?0000000h | Free Cluster |
| 001h | 0001h | ?0000001h | Reserved Cluster |
| 002h-FEFh | 0002h-FFEFh | 00000002h-?FFFFFEFh | Used Cluster; value points to next cluster |
| FF0h-FF6h | FFF0h-FFF6h | ?FFFFFF0h-?FFFFFF6h | Reserved values |
| FF7h | FFF7h | ?FFFFFF7h | Bad Cluster |
| FF8h-FFFh | FFF8h-FFFFh | ?FFFFFF8h-?FFFFFFFh | Last Cluster in File |

Note: FAT32 uses only bits [27:0], the upper 4 bits are usually 0 and should be left untouched. When accessing the Root Directory for FAT32, the ROM Code just starts from the Root Directory Cluster entry and follows the linked chain to retrieve the clusters.

When the booting file has been found, the ROM Code buffers each FAT entry corresponding to the file in a sector way. This means each cluster is translated to one or several sectors depending on how many sectors are in a cluster (BPB_SecPerClus). This buffer is used later on by the booting procedure to access the file.

#### 4.7.4.10 Pins Used for MMC Boot

The list of pins that are configured by the ROM in case of MMC boot mode are as follows. Please note that all the pins might not be driven at boot time.

**Table 4-26. Pins Used for MMC Boot**

| Signal name | Pin used |
|---|---|
| clk | mmc1_clk |
| cmd | mmc1_cmd_mux0 |
| dat0 | mmc1_dat0 |
| dat1 | mmc1_dat1 |
| dat2 | mmc1_dat2 |
| dat3 | mmc1_dat3 |

### *4.7.5 SPI*

SPI EEPROMs or SPI flashes have an EEPROM or NOR flash backend and they connect to the device using the serial SPI protocol. These typically devices operate in three stages – the command stage, the address stage and the data transfer stage. The command is usually an 8 bit value followed by the address (depending on the size of the device) followed by the data to be read or written. Because of the need for fewer pins, these devices are comparatively inexpensive, easy for board layout, and are the devices of choice when cost, complexity and form factor are critical considerations.

#### 4.7.5.1 SPI Features

The following features are included in SPI:

- Supports 12 MHz clock (50% duty cycle)
- Supports only SPI Mode 3 (clock polarity = 1, clock phase = 1)
- Supports only 24 bit addressable EEPROMs


Copyright © 2011–2016, Texas Instruments Incorporated

- Supports only 4-pin SPI mode (CS, CLK, Serial Input, Serial Output)
- The boot devices must be connected to channel 0 and must support the read command (03h)
- The boot image is copied into internal memory and then executed

#### 4.7.5.2 Initialization and Detection

The ROM Code initializes the SPI controller, pin muxing and clocks for communicating with the SPI device. The controller is initialized in Mode 3 and the clock is setup to operate at 12 MHz. There is no specific device identification routine that is executed by the ROM code to identify whether a boot device is preset or not. If no SPI device is present, the sector read will return only 0xFFFFFFFF and the SPI boot will be treated as failed.

#### 4.7.5.3 SPI Read Sector Procedure

The ROM Code reads SPI data from the boot device in 512 byte sectors. For each call to the SPI Read Sector routine, the SPI Read Command (0x03) is sent along with the 24 bit start address of the data to be read.

From the next iteration onwards, a dummy value is transmitted on the master out line and the data is received on the master in line. This needs to be done because SPI protocol always operated in full duplex mode. The dummy data transmitted by the ROM is the Read Command appended to the start address. The data from the boot device is received MSB first.

As the A8 is a little-endian processor, and SPI operates in a big-endian format, this means that while writing to the flash, care needs to be taken to write the image in a big-endian format. This way we can avoid doing the endian conversion at boot time, thus improving boot performance

#### 4.7.5.4 Pins Used for SPI Boot

The list of device pins that are configured by the ROM in the case of SPI boot mode are shown in the table below. Please note that all the pins might not be driven at boot time.

The default state of the other SPI chip select pins could be low at POR, so care must be taken if any other devices are connected to CS[1], CS[2] or CS[3]. External logic must be used to ensure that the chip select to these devices are not enabled by default. .

**Table 4-27. Pins Used for SPI Boot**

| Signal name | Pin used |
|---|---|
| cs | spi0_cs0 |
| miso | spi0_d0 |
| mosi | spi0_d1 |
| clk | spi0_sclk |

### *4.7.6 Blocks and Sectors Search Summary*

Table 4-28 summarizes numbers of blocks and sectors which are searched during the memory booting from devices requiring image shadowing. NAND is organized with blocks, which are erasable units.

**Table 4-28. Blocks and Sectors Searched on non-XIP Memories**

| Memory | Maximum number of blocks checked | Number of sectors searched |
|---|---|---|
| NAND | first 4 | Number of sectors in one block (1) |
| SPI, eMMC/eSD and MMC/SD cards (raw mode) | first 4 | 1 |

(1) Depends on NAND geometry.

Regarding MMC/SD card booting in FAT mode, the file system area is searched for one file.


Copyright © 2011–2016, Texas Instruments Incorporated

## 4.8 Peripheral Booting

### 4.8.1 Overview

The ROM Code can boot from three different peripheral interfaces:

| | |
|---|---|
| EMAC | 1000/100/10 Mbps Ethernet, using standard TCP/IP network boot protocols BOOTP and TFTP |
| PCIe | One lane @250 MB/s |
| UART | 115.2Kbps, 8 bits, even parity, 1 stop bit, no flow control |

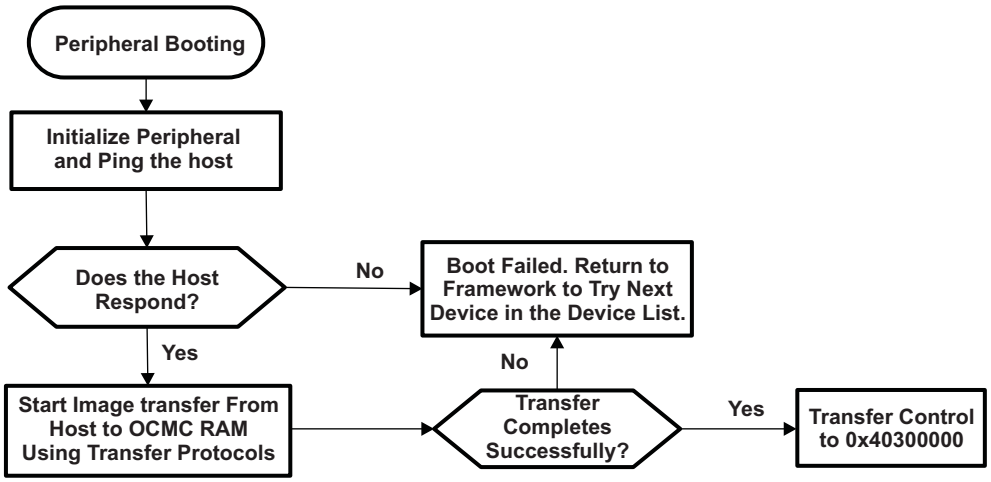
The purpose of booting from a peripheral interface is to download a boot image from an external host (typically a PC). This booting method is mostly used for programming flash memories connected to the device (e.g., in the case of initial flashing, firmware update or servicing).

### 4.8.2 Boot Image Location and Size

The boot image is downloaded directly into internal RAM at the location 40300000h. The maximum size of downloaded image is 110KB.

### 4.8.3 Peripheral Boot Procedure Overview

**Figure 4-23. Peripheral Booting Procedure**




### 4.8.4 EMAC Boot Procedure

#### 4.8.4.1 Device Initialization

- EMAC boot uses the CPGMAC0 subsystem of the device. It can be connected to external Ethernet PHY using the GMII0, RMII or RGMII interfaces and MDIO pins
- Device uses EFUSE register MAC_ID0 for Ethernet MAC address of the device.
- Device detects if the PHY is alive on the MDIO interface and
  - Reads the STATUS register to check if Ethernet link is active
  - Reads the CONTROL register to detect the auto-negotiated mode of operation
    - Is the mode full-duplex or half duplex
    - Speed of operation, 1000/100/10 Mbps


Copyright © 2011–2016, Texas Instruments Incorporated

#### 4.8.4.2 BOOTP (RFC 951)

The device then proceeds to obtain the IP and Boot information using BOOTP protocol. The device prepares and broadcasts the BOOTP message that has the following information:

- Device MAC address in “chaddr” field – to uniquely identify the device to the server.
- “Vender-class-identifier” option number 60 (RFC 1497, RFC 1533). Servers could use this information to identify the device type. The value present is "DM816x ROM v1.0"
- ”Client-identifier” option number 61 (RFC 1497, RFC 1533). This has the ASIC-ID structure which contains additional info for the device.

The device then expects a BOOTP response that provides the following information for the booting to proceed:

- Device IP address from “yiaddr” field
- Subnetmask from extended option 1 (RFC 1497, RFC 1533)
- Gateway IP from extended option number 3 (RFC 1497, RFC 1533) or from “giaddr” field of BOOTP response.
- Boot image filename from “file” field
- TFTP server IP address from the “siaddr” field

  Timeouts and retries
- Exponentially increasing timeouts starting from 4s
- Five retries

#### 4.8.4.3 FTP (RFC 1350)

After a successful BOOTP completion, the device initiates the TFTP download of the boot image into SRAM. The device has the capability to reach TFTP server within the local subnet or outside, though the gateway.

Timeouts and retries

- Timeout of 1s to receive a response for the READ request
- 5 retries for the READ request
- Retries are managed by server once data transfer starts (server re-sends a data packet if the ACK was not received within a timeout value)
- Device has a 60s timeout to complete the data transfer, to handle the scenario if the server dies in the middle of a data transfer

#### 4.8.4.4 Pins Used for EMAC Boot

The list of pins that are configured by the ROM in case of EMAC boot mode are in . Note that all the pins might not be driven at boot time.

**Table 4-29. Pins Used for EMAC Boot**

| Signal name | Pin used |
|---|---|
| col | gmii0_col |
| crs | gmii0_crs |
| gtx_clk | gmii0_gtxclk |
| rx_clk | gmii0_rxclk |
| rxd0-rxd7 | gmii0_rxd0-rxd7 |
| rxdv | gmii0_rxdv |
| rxer | gmii0_rxer |
| tx_clk | gmii0_txclk |
| txd_0-txd7 | gmii0_txd0-txd7 |
| txen | gmii0_txen |
| mclk | gmii0_mclk |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-29. Pins Used for EMAC Boot (continued)**

| Signal name | Pin used |
|---|---|
| mdio | gmii0_mdio |

**Table 4-30. Pins Used for Boot in RGMII Mode**

| Signal name | Pin used |
|---|---|
| rxclk | gmii0_txclk |
| rxctl | gmii0_col |
| txclk | gmii0_rxclk |
| txctl | gmii0_rxer |
| rxd0 | gmii0_rxd1 |
| rxd1 | gmii0_rxd2 |
| rxd2 | gmii0_crs |
| rxd3 | gmii0_rxd4 |
| txd0 | gmii0_rxd0 |
| txd1 | gmii0_rxd7 |
| txd2 | gmii0_rxd6 |
| txd3 | gmii0_rxd5 |
| mclk | gmii0_mclk |
| mdio | gmii0_mdio |

**Table 4-31. Pins Used for Boot in RMII Mode**

| Signal name | Pin used |
|---|---|
| refclk | rmii_refclk |
| crs | gmii0_rxclk |
| rxer | gmii0_rxer |
| txen | gmii0_rxd2 |
| rxd0 | gmii0_col |
| rxd1 | gmii0_crs |
| txd0 | gmii0_rxd0 |
| txd1 | gmii0_rxd1 |
| mclk | gmii0_mclk |
| mdio | gmii0_mdio |

**Table 4-32. Sysboot Pins**

| SYSBOOT[9:8] | PHY Mode |
|---|---|
| 00b | MII |
| 01b | RMII |
| 10b | RGMII |
| 11b | reserved |

## *4.8.5 PCIe Boot Procedure*

### 4.8.5.1 Device Initialization

- ROM code configures the PCIe vendor ID as 104Ch


Copyright © 2011–2016, Texas Instruments Incorporated

- ROM code configures the PCIe device ID as B801h.
- The BAR window sizes for inbound translation are configurable as per the tables below
- Only legacy interrupt A is enabled. MSI is disabled
- The device is configured in the D0 power state
- The link and device capability is configured for maximum flexibility
- The PCIe settings used by the ROM can be overridden after boot by the Root Complex

**Figure 4-24. PCIe Peripheral Booting Procedure**

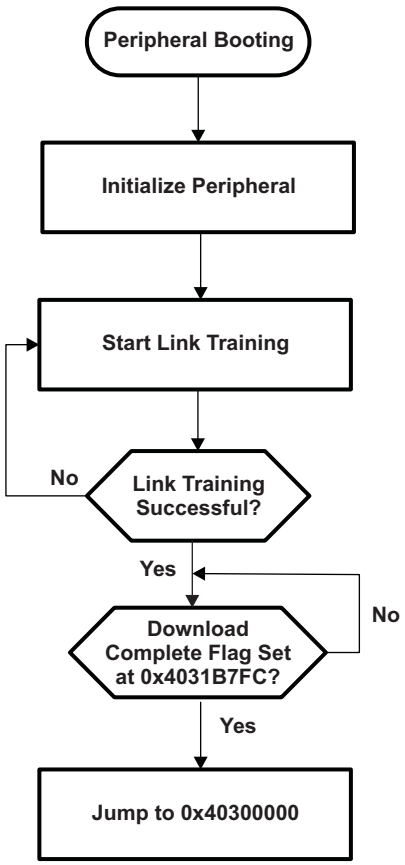



**Table 4-33. PCIe 32 BAR Window Size Configuration**

| PCIe32 | | | | | | | |
|---|---|---|---|---|---|---|---|
| BAR0 | BAR1 | BAR2 | | BAR3 | | BAR4 | |
| | | CS0BW | Size | CS0MUX[1:0] | Size | CS0WAIT | Size |
| 4KB | 8MB | 0 | 0 | 0 | 0 | 0 | 0 |
| | | 1 | 16MB | 01 | 32MB | 1 | 256MB |
| | | | | 10 | 64MB | | |
| | | | | 11 | 128MB | | |

**Table 4-34. PCIe 64 BAR Window Size Configuration**

| PCIe64 | | | | |
|---|---|---|---|---|
| BAR0/1 | BAR 2/3 | | BAR 4/5 | |
| | CS0WAIT :CS0BW | Size | CS0MUX[1:0] | Size |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-34. PCIe 64 BAR Window Size Configuration (continued)**

| PCIe64 | | | | |
|---|---|---|---|---|
| 4KB | 0 | 0 | 00 | 0 |
| | 01 | 256MB | 01 | 1GB |
| | 10 | 523MB | 10 | 2GB |
| | 11 | 1GB | 11 | 4GB |

**Table 4-35. PCIe BAR Window Base Address and Offset Configuration**

| BAR | Base Address | Offset | Comments |
|---|---|---|---|
| 0 | 0000 0000h | 0000 0000h | |
| 1 (64bit BAR0) | 4000 0000h | 4030 0000h | OCMC RAM |
| 2 | 6000 0000h | 5000 0000h | GPMC |
| 3 (64 Bit BAR2) | 8000 0000h | 8000 0000h | DDR 0 |
| 4 | A000 0000h | C000 0000h | DDR 1 |

Note that the base address and offset configurations can be changed from the host after the enumeration is complete.

### *4.8.6 UART Boot Procedure*

#### 4.8.6.1 Device Initialization

- UART boot uses UART0
- UART0 is configured to run at 115200 baud, 8-bits, even parity, 1 stop bit and no flow control.

#### 4.8.6.2 Boot Image Download

- UART boot uses x-modem client protocol to receive the boot image.
- Utilities like hyperterm, teraterm, minicom can be used on the PC side to download the boot image to the board
- With x-modem packet size of 1K throughout is roughly about 4KBytes/Sec.
- The ROM code will ping the host 10 times in 3s to start x-modem transfer. If host does not respond, UART boot will timeout.
- Once the transfer has started, if the host does not send any packet for 3s, UART boot will time out.
- If the delay between two consecutive bytes of the same packet is more than 2ms, the host is requested to re-transmit the entire packet again.
- Error checking using the CRC-16 support in x-modem. If an error is detected, the host is requested to re-transmit the packet again.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 4.8.6.3 Pins Used for UART Boot Mode

The list of pins that are configured by the ROM incase of UART boot mode are in Table 4-36. Note that all the pins might not be driven at boot time.

**Table 4-36. Pins Used for UART Boot**

| Signal name | Pin used |
|---|---|
| rx | uart0_rxd |
| tx | uart0_txd |

#### ASIC ID Structure

The ASIC ID size is 58 Bytes for UART and 81 Bytes for others.

## 4.9 Image Format

### *4.9.1 Overview*

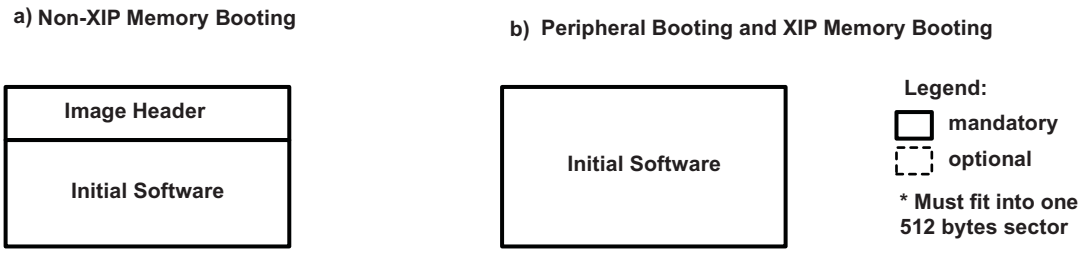
All preceding sections describe how the ROM Code searches and detects a boot image from a memory or a peripheral device type. This section describes the format of the boot image itself.

A boot image is basically made out of two major parts:

- The software to execute
- A header containing the destination address and size of the image for non-XIP memory devices

The mandatory section of a boot image contains the software which will be loaded into the memory and executed. An overview of the image formats is shown in Figure 4-25.

**Figure 4-25. Image Formats**




(a) Non - XIP Memory Booting
It is used for memories which require shadowing (for example, an MMC). The image must begin with a header which contains information on image size and destination address

(b) Peripheral Booting and XIP Memory Booting
When the memory device is of XIP type (e.g., NOR) the header is not needed and the image can contain code for direct execution. The same image format is used for peripheral booting (where the code is transferred to internal RAM).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 4.9.1.1 Image Formats

When the booting memory device is non-XIP (for example, an MMC), the image must contain a small header with the size of the software to load, and the destination address which tells where to store it.

The header is not needed when booting from an XIP memory device (for example, NOR) or in case of peripheral booting. In this case, the peripheral or memory booting image starts directly with executable code.

**Table 4-37. Image Formats**

| Field | non-XIP device (offset) | XIP device (offset) | Size (bytes) | Description |
|---|---|---|---|---|
| Size | 0000h | - | 4 | Size of the image |
| Destination | 0004h | - | 4 | Address where to store the image / code entry point |
| Image | 0008h | 0000h | x | Executable code |

**Note:** the “Destination” address field stands for both:

- Target address for the image copy from the non-XIP storage to the target XIP location (for example, internal RAM or SDRAM)
- Entry point for image code

In other words, the user must take care to locate the code entry point to the target address for image copy.

### *4.9.2 Table of Contents*

The TOC is 512 bytes long and consists of a maximum of 16 TOC items (32 bytes long each), located one after the other. If the number of TOC items is less than 16, the TOC item after the last used TOC item must be filled by FFh. Each TOC item contains information required by the ROM Code such as item location, size and name, as illustrated in Figure 4-26 and Table 4-38

**Figure 4-26. TOC Structure**

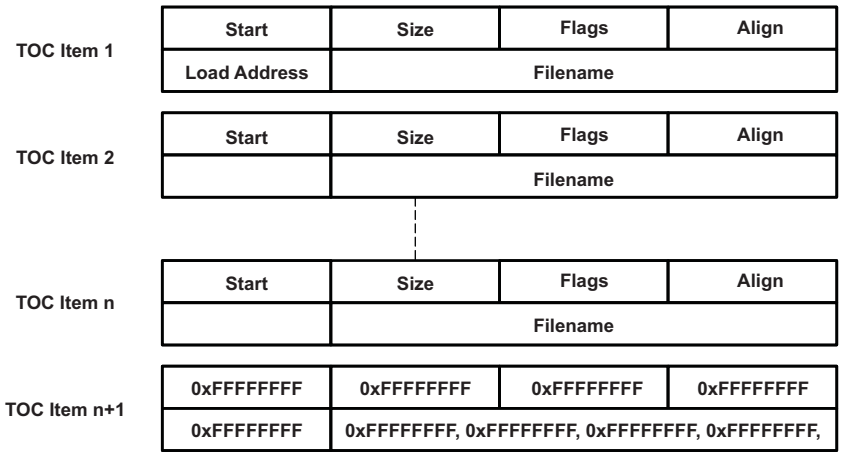



**Table 4-38. TOC Item Fields**

| Offset | Field | Size [bytes] | Description |
|---|---|---|---|
| 0h | Start | 4 | Offset from the start address of TOC to the actual address of sub image. |
| 4h | Size | 4 | Size of sub image. |
| 8h | Flags | 4 | Reserved |
| Ch | Align | 4 | Reserved |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-38. TOC Item Fields (continued)**

| Offset | Field | Size [bytes] | Description |
|---|---|---|---|
| 10h | Load Address | 4 | For an Initial SW TOC item during Memory Booting this field is the target address for item copy.<br>It is ignored in other cases. |
| 14h | Filename | 12 | 12 character long name of sub image, including the zero ('\0') terminator. |

The ROM Code recognizes sections pointed to by the TOC based on the filename as described in Table 4-39.

- The 'X-LOADER', '2ND', 'MLO', or 'ULO' section contains the Initial Software
- Optionally, the TOC may contain CH sections. CHSETTINGS is a mandatory section of CH and is used to recognize CH presence.

**Table 4-39. Filenames in TOC**

| Filename | Description |
|---|---|
| X-LOADER | Bootstrap |
| 2ND | Secondary loader (peripheral booting) |
| MLO | MMC/SD Loader (Memory booting) |
| ULO | Alternative filename used for Secondary Loader via USB (Peripheral booting) |
| CHSETTINGS | Configuration Header general setting item. this item is mandatory for CH |
| CHRAM | Configuration Header EMIF item |
| CHFLASH | Configuration Header GPMC item |
| CHMMCSD | Configuration Header MMD/SD item |

## 4.10 Execution

### *4.10.1 Overview*

One of the early steps of the public ROM code execution is to search for a boot image from the requested medium (configured by the MBOOT pins) and copy it to RAM if needed. If the boot interface is non-XIP then the image is simply copied to the provided destination address (internal or external RAM) and then executed. If the boot interface is of XIP type then the image copy is not needed and execution is directly given to the XIP memory.

### *4.10.2 Execution*

The image is executed at the time the ROM Code performs the branch to the first executable instruction inside the Initial SW. For a non-XIP device, the execution address is the first word after the header. The branch is performed in ARM supervisor mode. The R0 register points to the booting parameters structure (see Table 4-40) which contains details about the booting execution.

**Table 4-40. Booting Parameters Structure**

| Offset | Field | Size [bytes] | Description |
|---|---|---|---|
| 0h | Reserved | 4 | Reserved |
| 4h | Memory booting device descriptor address | 4 | Pointer to the memory device descriptor that has been used during the memory booting process |
| 5h | Current Booting Device | 1 | Code of device used for booting<br>0h – void, no device<br>1h – XIP MUX1 memory<br>2h – XIPWAIT MUX1 memory (wait monitoring on) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-40. Booting Parameters Structure (continued)**

| Offset | Field | Size [bytes] | Description |
|---|---|---|---|
| | | | 3h –XIP MUX2 memory<br>4h – XIPWAIT MUX 2 (wait monitoring on)<br>5h – NAND<br>6h – NAND with I2C<br>8h – MMC/SD port 0<br>9h – MMC/SD port 1<br>15h – SPI<br>41h – UART0<br>44h - USB<br>46h – CPGMAC0 |
| 6h | Reset Reason | 1 | Current reset reason bit mask (bit=1-event present)<br>[0] – power-on reset<br>[1] – global software reset<br>[3] – security violation reset<br>[4] – MPU watchdog reset<br>[5] – secure watchdog reset<br>[6] – external warm reset<br>Other bits – reserved |
| 7h | Reserved | 1 | Reserved |

## 4.11 Services for HLOS Support

The TrustZone® architecture of the Cortex™-A8 core restricts accesses to secure mode only coprocessor registers. The Cortex-A8 core of the device is forbidden from entering the secure mode and hence cannot directly access any secure resources. However, a high-level OS needs to access secure registers for L2 cache maintenance and other operations.

For these purposes, the ROM Code provides different primitives that can be called. These services are implemented in monitor mode (service must be called by writing function ID into R12 register and using the SMC instruction) and do not use any resources like RAM/stack or hardware outside the MPU. The caller must take care of saving R0, R1, and R12 before using these APIs, as these get overwritten by the ROM code.

The services provided are:

- Write Auxiliary Control Register (WRITE_ACR).
- Write Nonsecure Access Control Register (WRITE_NACR).
- Write L2 Cache Auxiliary Control Register (WRITE_CACR).

**Table 4-41. Write Auxiliary Control Register (WRITE_ACR)**

| Function ID | Description | |
|---|---|---|
| R12 = 0x100 | This function writes the ARM CP15 Auxiliary Control Register with the input value in R0. | |
| **Parameters** | | |
| **Type** | **Field** | **Description** |
| Input | 32 bit unsigned integer – R0<br>32 bit unsigned integer – R1 | Value to set in the register<br>Don't care |
| Output | - | - |
| Return | - | - |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-42. Write Nonsecure Access Control Register (WRITE_NACR)**

| Function ID | Description | |
|---|---|---|
| R12 = 0x101 | This function writes the ARM CP15 Nonsecure Access Control Register with the input value in R0. | |
| **Parameters** | | |
| **Type** | **Field** | **Description** |
| Input | 32 bit unsigned integer – R0<br>32 bit unsigned integer – R1 | Value to set in the register<br>Don't care |
| Output | - | - |
| Return | - | - |

**Table 4-43. Write L2 Cache Auxiliary Control Register (WRITE_CACR)**

| Function ID | Description | |
|---|---|---|
| R12 = 0x102 | This function writes the ARM CP15 L2 Cache Auxiliary Control Register with the input value in R0. | |
| **Parameters** | | |
| **Type** | **Field** | **Description** |
| Input | 32 bit unsigned integer – R0<br>32 bit unsigned integer – R1 | Value to set in the register<br>Don't care |
| Output | - | - |
| Return | - | - |

For more information about ARM registers, see the *Cortex™-A8 Technical Reference Manual*.

## 4.12 Tracing

Tracing in the Public ROM Code consist in three 32 bit vectors for which each bit corresponds to a particular “way point” in the ROM Code execution sequence (refer to Section 4.3.2.4). Tracing vectors are initialized at the very beginning of the startup phase and updated all along the boot process.

There are two sets of tracing vectors. The first set is the current trace information (after cold or warm reset). The second set holds a copy of trace vectors collected at the first ROM Code run after cold reset. As a consequence after a warm reset, it is possible to have visibility on the boot scenario that occurred during cold reset.

**Table 4-44. Tracing Vectors**

| Trace vector | Bit # | Group | Meaning |
|---|---|---|---|
| 1 | 0 | General | Passed the public reset vector |
| 1 | 1 | General | Entered main function |
| 1 | 2 | General | Running after the cold reset |
| 1 | 3 | Boot | Main booting routine entered |
| 1 | 4 | Memory Boot | Memory booting started |
| 1 | 5 | Peripheral Boot | Peripheral booting started |
| 1 | 6 | Boot | Booting loop reached last device |
| 1 | 7 | Boot | Header found |
| 1 | 8 | Boot | Reserved |
| 1 | 9 | Boot | Reserved |
| 1 | 10 | Peripheral Boot | Reserved |
| 1 | 11 | Peripheral Boot | Reserved |
| 1 | 12 | Peripheral Boot | Device initialized |
| 1 | 13 | Peripheral Boot | Asic Id sent |
| 1 | 14 | Peripheral Boot | Image received |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-44. Tracing Vectors (continued)**

| Trace vector | Bit # | Group | Meaning |
|---|---|---|---|
| 1 | 15 | Peripheral Boot | Peripheral booting failed |
| 1 | 16 | Peripheral Boot | Booting Message not received (timeout) |
| 1 | 17 | Peripheral Boot | Image size not received (timeout) |
| 1 | 18 | Peripheral Boot | Image not received (timeout) |
| 1 | 19 | Reserved | Reserved |
| 1 | 20 | Configuration Header | CHSETTINGS found |
| 1 | 21 | Configuration Header | CHSETTINGS executed |
| 1 | 22 | Configuration Header | CHRAM executed |
| 1 | 23 | Configuration Header | CHFLASH executed |
| 1 | 24 | Configuration Header | CHMMCSD clocks executed |
| 1 | 25 | Configuration Header | CHMMCSD bus width executed |
| 1 | 26 | Reserved | Reserved |
| 1 | 27 | Reserved | Reserved |
| 1 | 28 | Reserved | Reserved |
| 1 | 29 | Reserved | Reserved |
| 1 | 30 | Reserved | Reserved |
| 1 | 31 | Reserved | Reserved |
| 2 | 0 | Companion chip | Reserved |
| 2 | 1 | Companion chip | Reserved |
| 2 | 2 | Companion chip | Reserved |
| 2 | 3 | Companion chip | Reserved |
| 2 | 4 | USB | USB connect |
| 2 | 5 | USB | USB configured state |
| 2 | 6 | USB | USB VBUS valid |
| 2 | 7 | USB | USB session valid |
| 2 | 8 | Reserved | Reserved |
| 2 | 9 | Reserved | Reserved |
| 2 | 10 | Reserved | Reserved |
| 2 | 11 | Reserved | Reserved |
| 2 | 12 | Memory Boot | Memory booting trial 0 |
| 2 | 13 | Memory Boot | Memory booting trial 1 |
| 2 | 14 | Memory Boot | Memory booting trial 2 |
| 2 | 15 | Memory Boot | Memory booting trial 3 |
| 2 | 16 | Memory Boot | Execute GP image |
| 2 | 17 | Reserved | Reserved |
| 2 | 18 | Reserved | Rreserved |
| 2 | 19 | Reserved | Reserved |
| 2 | 20 | Reserved | Reserved |
| 2 | 21 | Reserved | Reserved |
| 2 | 22 | Reserved | Reserved |
| 2 | 23 | Reserved | Reserved |
| 2 | 24 | Reserved | Reserved |
| 2 | 25 | Reserved | Reserved |
| 2 | 26 | Reserved | Rreserved |
| 2 | 27 | Reserved | Reserved |
| 2 | 28 | Reserved | Reserved |
| 2 | 29 | Reserved | Reserved |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 4-44. Tracing Vectors (continued)**

| Trace vector | Bit # | Group | Meaning |
|---|---|---|---|
| 2 | 30 | Reserved | Reserved |
| 2 | 31 | Reserved | Reserved |
| 3 | 0 | Memory Boot | Memory booting device NULL |
| 3 | 1 | Memory Boot | Memory booting device XIP |
| 3 | 2 | Memory Boot | Memory booting device XIPWAIT |
| 3 | 3 | Memory Boot | Memory booting device NAND |
| 3 | 4 | Memory Boot | Memory booting device OneNAND |
| 3 | 5 | Memory Boot | Memory booting device MMCSD0 |
| 3 | 6 | Reserved | Reserved |
| 3 | 7 | Memory Boot | Memory booting device MMCSD1 |
| 3 | 8 | Reserved | Reserved |
| 3 | 9 | Reserved | Reserved |
| 3 | 10 | Memory Boot | Memory booting device EMIF LPDDR2-NVM |
| 3 | 11 | Reserved | Memory booting device SPI |
| 3 | 12 | Reserved | Reserved |
| 3 | 13 | Reserved | Reserved |
| 3 | 14 | Reserved | Reserved |
| 3 | 15 | Reserved | Reserved |
| 3 | 16 | Reserved | Peripheral booting device UART0 |
| 3 | 17 | Reserved | Peripheral booting device UART1 |
| 3 | 18 | Peripheral Boot | Peripheral booting device UART2 |
| 3 | 19 | Reserved | Reserved |
| 3 | 20 | Peripheral Boot | Peripheral booting device USB |
| 3 | 21 | Peripheral Boot | Peripheral booting device USB ULPI |
| 3 | 22 | Peripheral Boot | Peripheral booting device GPGMAC0 |
| 3 | 23 | Reserved | Peripheral booting device PCIe |
| 3 | 24 | Reserved | Peripheral booting device NULL |
| 3 | 25 | Reserved | Reserved |
| 3 | 26 | Reserved | Reserved |
| 3 | 27 | Reserved | Reserved |
| 3 | 28 | Reserved | Reserved |
| 3 | 29 | Reserved | Reserved |
| 3 | 30 | Reserved | Reserved |
| 3 | 31 | Reserved | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

# DCAN Controller Area Network

This document describes the DCAN controller area network (DCAN).

**Topic** **Page**


5.1 Overview ..... 1073
5.2 CAN Operation ..... 1075
5.3 Dual Clock Source ..... 1081
5.4 Interrupt Functionality ..... 1082
5.5 Local Power-Down Mode ..... 1084
5.6 Parity Check Mechanism ..... 1086
5.7 Debug/Suspend Mode ..... 1087
5.8 Configuration of Message Objects ..... 1087
5.9 Message Handling ..... 1090
5.10 CAN Bit Timing ..... 1095
5.11 Message Interface Register Sets ..... 1103
5.12 Message RAM ..... 1105
5.13 GIO Support ..... 1110
5.14 Registers ..... 1110



Copyright © 2011–2016, Texas Instruments Incorporated

## 5.1 Overview

This chapter contains a general description of the DCAN Controller Area Network module, a high-integrity serial communications protocol for distributed real-time applications.

The DCAN module supports bit rates up to 1 Mbit/s and is compliant to the CAN 2.0B protocol specification. This device has two DCAN modules, referred to as DCAN0 and DCAN1 in this document.

### 5.1.1 Features

The DCAN module implements the following features:

- Support for CAN protocol version 2.0 part A, B
- Bit rates up to 1 MBit/s
- Dual clock source
- 64 message objects (see Section 5.1.2.3)
- Individual identifier mask for each message object
- Programmable FIFO mode for message objects
- Programmable loop-back modes for self-test operation
- Suspend mode for debug support
- Software module reset
- Automatic bus on after Bus-Off state by a programmable 32-bit timer
- Message RAM parity check mechanism
- Direct access to message RAM during test mode
- CAN Rx/Tx pins configurable as general purpose input/output (I/O) pins
- Support for two interrupt lines: Level 0 and Level 1
- Local power down and wakeup support
- Automatic RAM initialization
- Support for DMA access

### 5.1.2 Functional Description

The DCAN module consists of the components CAN Core, Message RAM, Message RAM Interface, Message Handler, Registers and Message Object access and Module Interface (see Figure 5-1). The DCAN module performs CAN protocol communication according to ISO 11898-1. The bit rate can be programmed to values up to 1 MBit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

For communication on a CAN network, individual message objects can be configured. The message objects and identifier masks are stored in the message RAM.

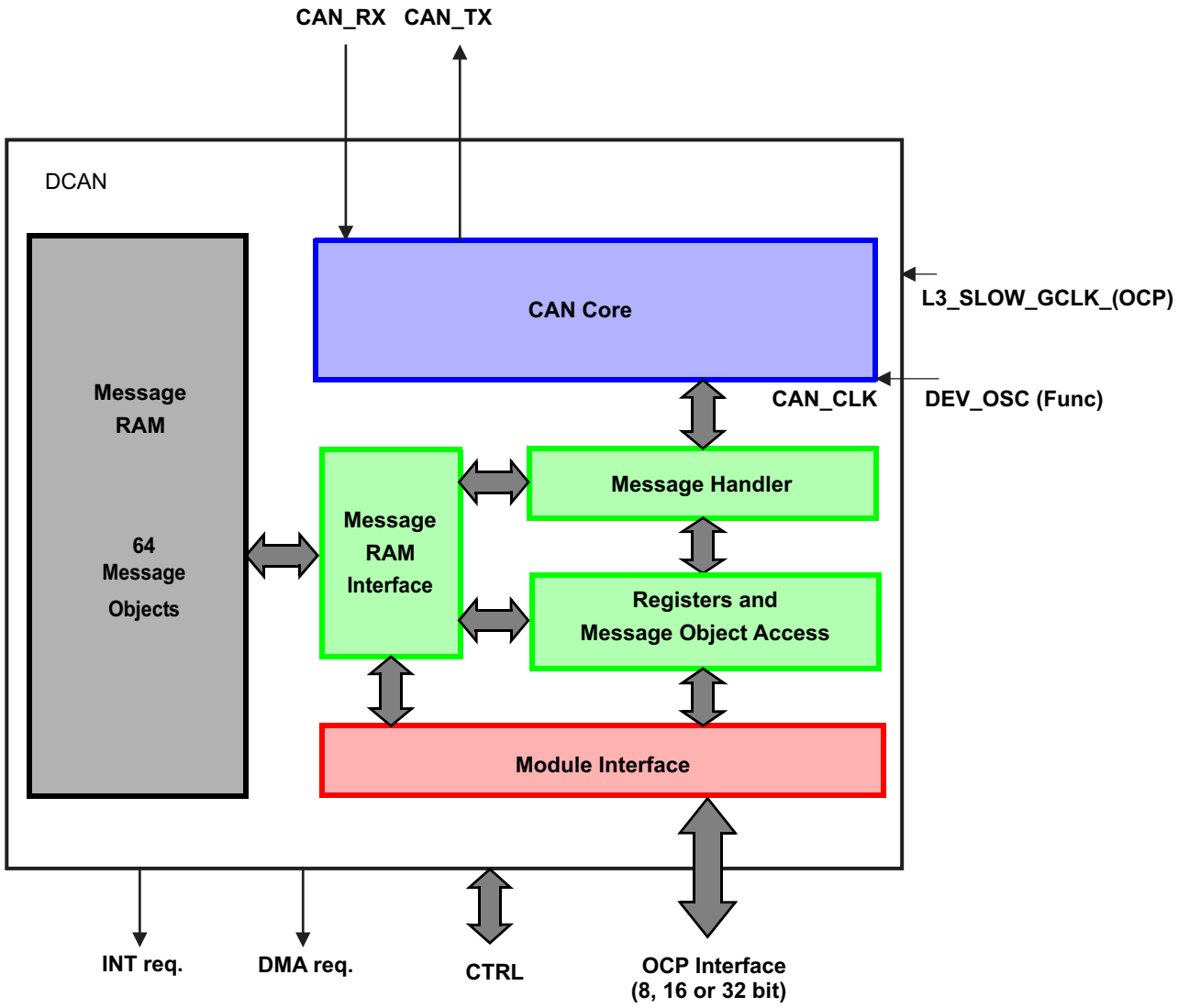
All functions concerning the handling of messages are implemented in the message handler. Those functions are acceptance filtering, the transfer of messages between the CAN core and the message RAM, and the handling of transmission requests, as well as the generation of interrupts or DMA requests.

The register set of the DCAN module can be accessed directly by the CPU via the module interface. These registers are used to control and configure the CAN core and the message handler, and to access the message RAM.

Figure 5-1 shows the DCAN block diagram and its features are described below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 5-1. DCAN Block Diagram**




#### 5.1.2.1 CAN Core

The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. It handles all ISO 11898-1 protocol functions.

#### 5.1.2.2 Message Handler

The message handler is a state machine that controls the data transfer between the single-ported message RAM and the CAN core's Rx/Tx shift register. It also handles acceptance filtering and the interrupt/DMA request generation as programmed in the control registers.

#### 5.1.2.3 Message RAM

DCAN0 and DCAN1 enables the storage of 64 CAN messages.

#### 5.1.2.4 Message RAM Interface

Three interface register sets control the CPU read and write accesses to the message RAM. There are two interface registers sets for read and write access, IF1 and IF2, and one interface register set for read access only, IF3. Additional information can be found in Section 5.9.12.


Copyright © 2011–2016, Texas Instruments Incorporated

The interface registers have the same word-length as the message RAM.

#### 5.1.2.5 Registers and Message Object Access

Data consistency is ensured by indirect accesses to the message objects. During normal operation, all CPU and DMA accesses to the message RAM are done through interface registers. In a dedicated test mode, the message RAM is memory mapped and can be directly accessed by either CPU or DMA.

#### 5.1.2.6 Module Interface

The DCAN module registers are accessed by the CPU or user software through a 32-bit peripheral bus interface.

#### 5.1.2.7 Dual Clock Source

Two clock domains are provided to the DCAN module: the peripheral synchronous clock domain (L3_SLOW_GCLK) and the peripheral asynchronous clock source domain (DEV_OSC) for CAN_CLK.

## 5.2 CAN Operation

After hardware reset, the Init bit in the CAN control register (DCAN CTL) is set and all CAN protocol functions are disabled. The CAN module must be initialized before operating it. Figure 5-2 illustrates the basic initialization flow for the CAN module.

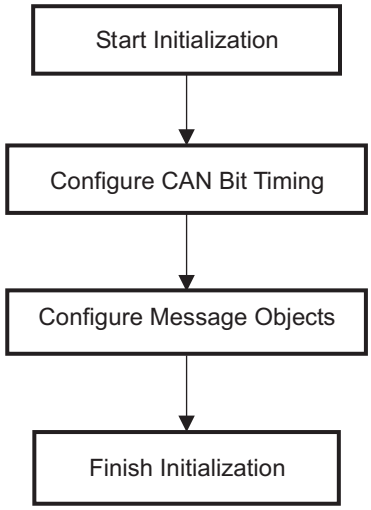
### *5.2.1 CAN Module Initialization*

A general CAN module initialization would mean the following two critical steps:

- Configuration of the CAN bit timing
- Configuration of message objects

To initialize the CAN controller, the CPU has to set up the CAN bit timing and those message objects that have to be used for CAN communication. Message objects that are not needed, can be deactivated.

**Figure 5-2. CAN Module General Initialization Flow**




#### 5.2.1.1 Configuration of CAN Bit Timing

The CAN module must be in initialization mode to configure the CAN bit timing.

For CAN bit timing software configuration flow, see Figure 5-3.

**Step 1:** Enter *initialization mode* by setting the Init (Initialization) bit in the CAN control register.

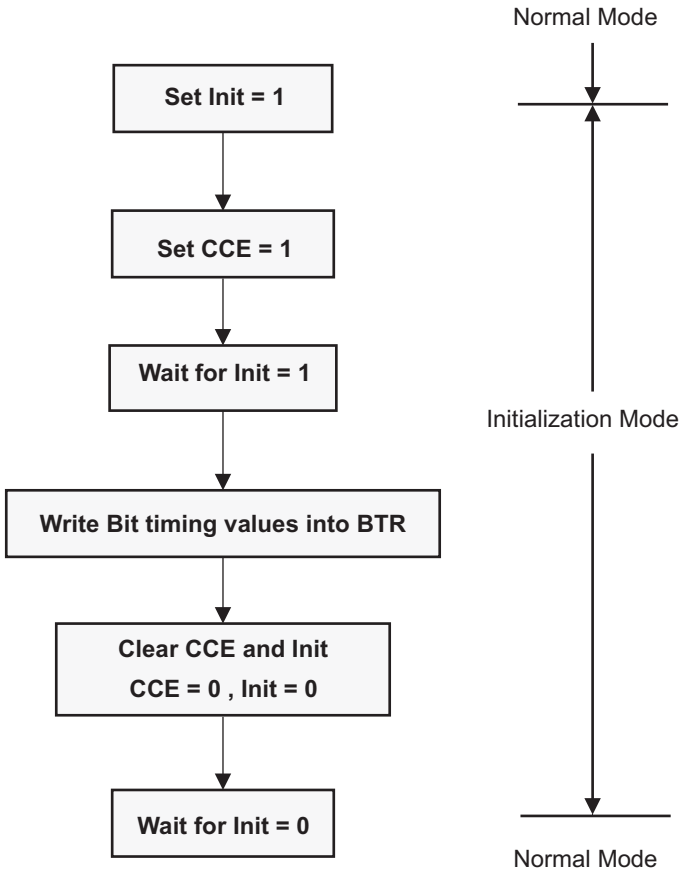

Copyright © 2011–2016, Texas Instruments Incorporated

While the Init bit is set, the message transfer from and to the CAN bus is stopped, and the status of the CAN_TX output is recessive (high).

The CAN error counters are not updated. Setting the Init bit does not change any other configuration register.

Also, note that the CAN module is in initialization mode on hardware reset and during Bus-Off.

**Figure 5-3. CAN Bit-Timing Configuration**




**Step 2:** Set the Configure Change Enable ( CCE) bit in the CAN control register.

The access to the Bit Timing register (BTR) for the configuration of the bit timing is enabled when both Init and CCE bits in the CAN Control register are set.

**Step 3:** Wait for the Init bit to get set. This would make sure that the module has entered *Initialization mode*.

**Step 4:** Write the bit timing values into the bit timing register. See Section 5.10.2 for the BTR value calculation for a given bit timing.

**Step 5:** Clear the CCE and Init bit.

**Step 6:** Wait for the Init bit to clear. This would ensure that the module has come out of *initialization mode*.

Following these steps, the module comes to operation by synchronizing itself to the CAN bus, provided the BTR is configured as per the CAN bus baud rate, although the message objects have to be configured before carrying out any communication.

**NOTE:** The module will not come out of the *initialization mode* if any incorrect BTR values are written in step 4.


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** The required message objects should be configured as transmit or receive objects before the start of data transfer as explained in Section 5.2.1.

#### 5.2.1.2 Configuration of Message Objects

The message objects can be configured only through the interface registers; the CPU does not have direct access to the message object (message RAM) . Familiarize yourself with the interface register set (IFx) usage (see Section 5.11) and the message object structure (see Section 5.12) before configuring the message objects.

For more information regarding the procedure to configure the message objects, see Section 5.8. All the message objects should be configured to particular identifiers or set to not valid before the message transfer is started. It is possible to change the configuration of message objects during normal operation (that is between data transfers).

**NOTE:** The message objects initialization is independent of the bit-timing configuration.

#### 5.2.1.3 DCAN RAM Hardware Initialization

The memory hardware initialization for the DCAN module is enabled in the device control register, DCAN_RAMINIT, which initializes the RAM with zeros and sets parity bits accordingly. Wait for the RAMINIT_DONE bit to be set to ensure successful RAM initialization.

For more details on RAM hardware initialization, see the system module reference guide.

### *5.2.2 CAN Message Transfer (Normal Operation)*

Once the DCAN is initialized and Init bit is reset to zero, the CAN core synchronizes itself to the CAN bus and is ready for message transfer as per the configured message objects.

The CPU may enable the interrupt lines (setting IE0 and IE1 to ‘1’) at the same time when it clears Init and CCE. The status interrupts EIE and SIE may be enabled simultaneously.

The CAN communication can be carried out in any of the following two modes: interrupt and polling.

The interrupt register points to those message objects with IntPnd = ‘1’. It is updated even if the interrupt lines to the CPU are disabled (IE0/IE1 are zero).

The CPU may poll all MessageObject’s NewDat and TxRqst bits in parallel from the NewData X registers and the Transmission Request X Registers (DCAN TXRQ X). Polling can be made easier if all transmit objects are grouped at the low numbers and all receive objects are grouped at the high numbers.

Received messages are stored into their appropriate message objects if they pass acceptance filtering.

The whole message (including all arbitration bits, DLC and up to eight data bytes) is stored into the message object. As a consequence (e.g., when the identifier mask is used), the arbitration bits which are masked to “don’t care” may change in the message object when a received message is stored.

The CPU may read or write each message at any time via the interface registers, as the message handler guarantees data consistency in case of concurrent accesses.

If a permanent message object (arbitration and control bits set up during configuration and leaving unchanged for multiple CAN transfers) exists for the message, it is possible to only update the data bytes.

If several transmit messages should be assigned to one message object, the whole message object has to be configured before the transmission of this message is requested.

The transmission of multiple message objects may be requested at the same time. They are subsequently transmitted, according to their internal priority.

Messages may be updated or set to not valid at any time, even if a requested transmission is still pending. However, the data bytes will be discarded if a message is updated before a pending transmission has started.


Copyright © 2011–2016, Texas Instruments Incorporated

Depending on the configuration of the message object, a transmission may be automatically requested by the reception of a remote frame with a matching identifier.

#### 5.2.2.1 Automatic Retransmission

According to the CAN Specification (ISO11898), the DCAN provides a mechanism to automatically retransmit frames which have lost arbitration or have been disturbed by errors during transmission. The frame transmission service will not be confirmed to the user before the transmission is successfully completed.

By default, this automatic retransmission is enabled. It can be disabled by setting bit disable automatic retransmission ( DAR) in the CAN Control register. Further details to this mode are provided in Section 5.9.3.

#### 5.2.2.2 Auto-Bus-On

By default, after the DCAN has entered Bus-Off state, the CPU can start a Bus-Off-Recovery sequence by resetting the Init bit. If this is not done, the module will stay in Bus-Off state.

The DCAN provides an automatic Auto-Bus-On feature which is enabled by bit ABO in the CAN control register. If set, the DCAN will automatically start the Bus-Off-Recovery sequence. The sequence can be delayed by a user-defined number of L3_SLOW_GCLK cycles which can be defined in the Auto-Bus-On Time register (DCAN ABOTR).

**NOTE:** If the DCAN goes to Bus-Off state due to a massive occurrence of CAN bus errors, it stops all bus activities and automatically sets the Init bit. Once the Init bit has been reset by the CPU or due to the Auto-Bus-On feature, the device will wait for 129 occurrences of bus Idle (equal to 129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus-Off recovery sequence, the error counters will be reset.

### *5.2.3 Test Modes*

The DCAN module provides several test modes which are mainly intended for production tests or self test.

For all test modes, the Test bit in the CAN control register needs to be set to 1. This enables write access to the test register (DCAN TEST).

**NOTE:** When using any of the Loop Back modes, it must be ensured by software that all message transfers are finished before setting the Init bit to 1.

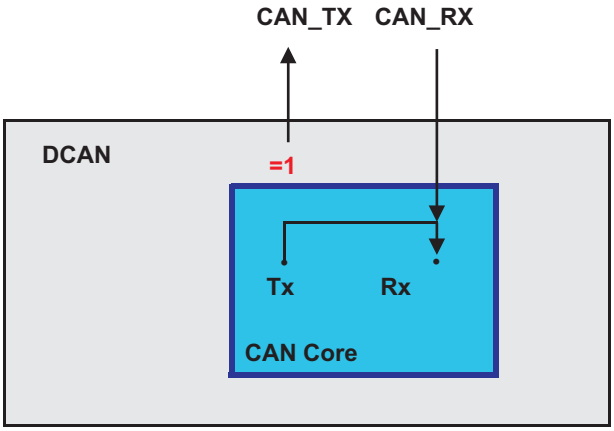
#### 5.2.3.1 Silent Mode

The silent mode may be used to analyze the traffic on the CAN bus without affecting it by sending dominant bits (for example, acknowledge bit, overload flag, active error flag). The DCAN is still able to receive valid data frames and valid remote frames, but it will not send any dominant bits. However, these are internally routed to the CAN core.

Figure 5-4 shows the connection of signals CAN_TX and CAN_RX to the CAN core in silent mode. Silent mode can be activated by setting the Silent bit in the Test register to one. In ISO 11898-1, the silent mode is called the bus monitoring mode.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 5-4. CAN Core in Silent Mode**




### 5.2.3.2 Loopback Mode

The loopback mode is mainly intended for hardware self-test functions. In this mode, the CAN core uses internal feedback from Tx output to Rx input. Transmitted messages are treated as received messages, and can be stored into message objects if they pass acceptance filtering. The actual value of the CAN_RX input pin is disregarded by the CAN core. Transmitted messages can still be monitored at the CAN_TX pin.

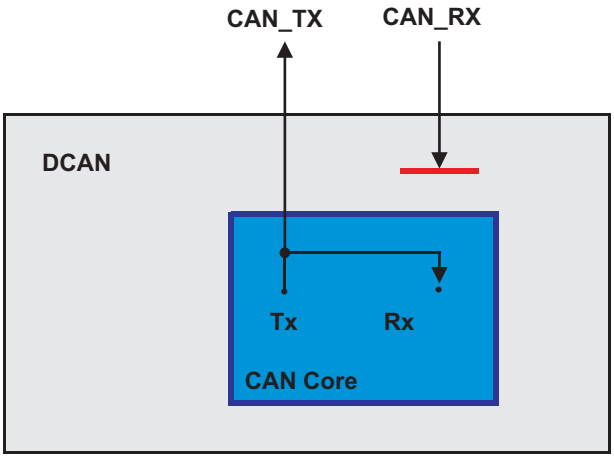
In order to be independent from external stimulation, the CAN core ignores acknowledge sampled in the acknowledge slot of a data/remote frame.

Figure 5-5 shows the connection of signals CAN_TX and CAN_RX to the CAN core in loopback mode.

Loopback mode can be activated by setting bit LBack in the test register to one.

**NOTE:** In loopback mode, the signal path from CAN core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN core are disregarded. For including these into the testing, see Section 5.2.3.3.

**Figure 5-5. CAN Core in Loopback Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

### 5.2.3.3 External Loopback Mode

The external loopback mode is similar to the loopback mode; however, it includes the signal path from CAN core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN core. When external loopback mode is selected, the input of the CAN core is connected to the input buffer of the Tx pin.

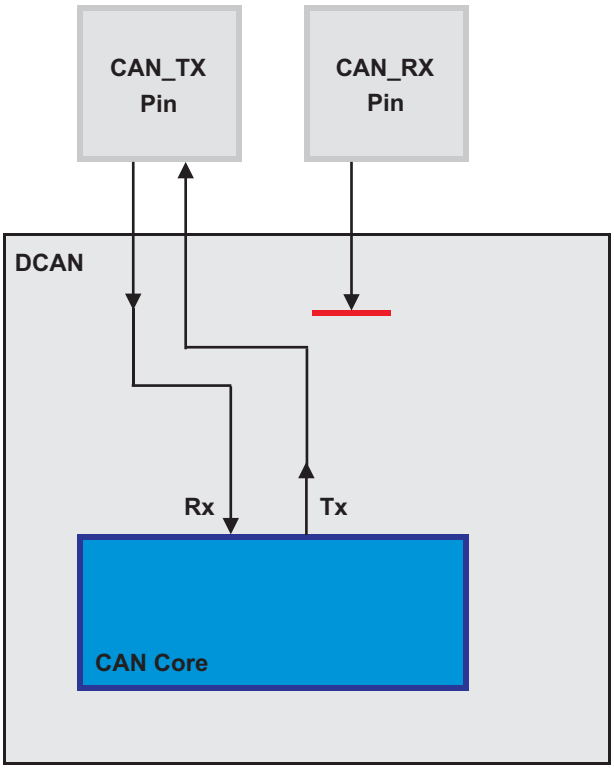
With this configuration, the Tx pin IO circuit can be tested.

External loopback mode can be activated by setting bit **ExL** in test register to one.

Figure 5-6 shows the connection of signals CAN_TX and CAN_RX to the CAN core in external loopback mode.

**NOTE:** When loopback mode is active (LBack bit set), the ExL bit will be ignored.

**Figure 5-6. CAN Core in External Loopback Mode**



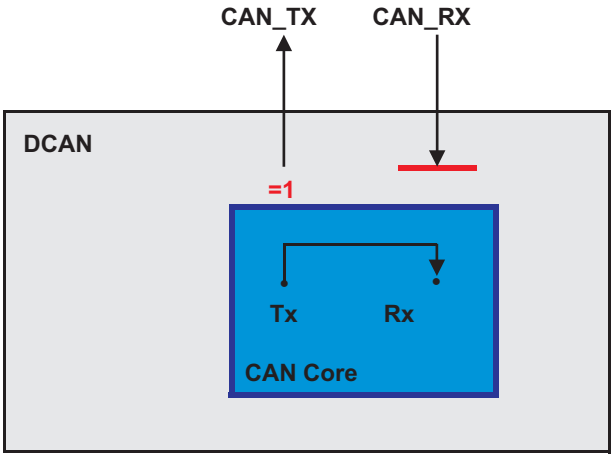

### 5.2.3.4 Loopback Mode Combined With Silent Mode

It is also possible to combine loopback mode and silent mode by setting bits LBack and Silent at the same time. This mode can be used for a “Hot Selftest,” i.e., the DCAN hardware can be tested without affecting the CAN network. In this mode, the CAN_RX pin is disconnected from the CAN core and no dominant bits will be sent on the CAN_TX pin.

Figure 5-7 shows the connection of the signals CAN_TX and CAN_RX to the CAN core in case of the combination of loopback mode with silent mode.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 5-7. CAN Core in Loop Back Combined With Silent Mode**




#### 5.2.3.5 Software Control of CAN_TX pin

Four output functions are available for the CAN transmit pin CAN_TX. In addition to its default function (serial data output), the CAN_TX pin can drive constant dominant or recessive values, or it can drive the CAN sample point signal to monitor the CAN core's bit timing.

Combined with the readable value of the CAN_RX pin, this function can be used to check the physical layer of the CAN bus.

The output mode of pin CAN_TX is selected by programming the test register bits Tx[1:0] as described in section Section 5.14.1.6.

**NOTE:** The software control for the CAN_TX pin interferes with CAN protocol functions. For CAN message transfer or any of the test modes (loopback mode, external loopback mode or silent mode), the CAN_TX pin should operate in its default functionality.

## 5.3 Dual Clock Source

Two clock domains are provided to the DCAN module: the peripheral synchronous clock domain (L3_SLOW_GCLK) as the general module clock source, and the peripheral asynchronous clock source domain (DEV_OSC) provided to the CAN core (as clock source CAN_CLK) for generating the CAN bit timing.

Both clock domains can be derived from the same clock source (so that L3_SLOW_GCLK = DEV_OSC).

For more information on how to configure the relevant clock source registers in the system module, see the system module reference guide and the device-specific data sheet.

Between the two clock domains, a synchronization mechanism is implemented in the DCAN module in order to ensure correct data transfer.

**NOTE:** If the dual clock functionality is used, then L3_SLOW_GCLK must always be higher or equal to CAN_CLK (DEV_OSC) (derived from the asynchronous clock source), in order to achieve a stable functionality of the DCAN. Here also the frequency shift of the modulated L3_SLOW_GCLK has to be considered:

$f_{0,\ L3_SLOW_GCLK}(OCP) \pm \Delta f_{FM,\ L3_SLOW_GCLK}(OCP) \geq f_{CANCLK}$


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** The CAN core has to be programmed to at least 8 clock cycles per bit time. To achieve a transfer rate of 1 MBaud when using the asynchronous clock domain as the clock source for CAN_CLK (DEV_OSC), an oscillator frequency of 8MHz or higher has to be used.

## 5.4 Interrupt Functionality

Interrupts can be generated on two interrupt lines: DCANINT0 and DCANINT1. These lines can be enabled by setting the IE0 and IE1 bits, respectively, in the CAN control register. The interrupts are level triggered at the chip level.

The DCAN provides three groups of interrupt sources: message object interrupts, status change interrupts, and error interrupts (see Figure 5-8 and Figure 5-9).

The source of an interrupt can be determined by the interrupt identifiers Int0ID/Int1ID in the interrupt register (see Section 5.14.1.5). When no interrupt is pending, the register will hold the value zero.

Each interrupt line remains active until the dedicated field in the interrupt register DCAN INT (Int0ID / Int1ID) again reach zero (this means the cause of the interrupt is reset), or until IE0 / IE1 are reset.

The value 0x8000 in the Int0ID field indicates that an interrupt is pending because the CAN core has updated (not necessarily changed) the Error and Status register (DCAN ES and PARITYERR_EOI). This interrupt has the highest priority. The CPU can update (reset) the status bits WakeUpPnd, RxOk, TxOk and LEC by reading the error and status register DCAN ES, but a write access of the CPU will never generate or reset an interrupt.

Values between 1 and the number of the last message object indicates that the source of the interrupt is one of the message objects, Int0ID resp. Int1ID will point to the pending message interrupt with the highest priority. The Message Object 1 has the highest priority; the last message object has the lowest priority.

An interrupt service routine that reads the message that is the source of the interrupt may read the message and reset the message object's IntPnd at the same time (ClrIntPnd bit in the IF1/IF2 command register). When IntPnd is cleared, the interrupt register will point to the next message object with a pending interrupt.

### *5.4.1 Message Object Interrupts*

Message object interrupts are generated by events from the message objects. They are controlled by the flags IntPND, TxIE and RxIE that are described in Section 5.12.1.

Message object interrupts can be routed to either DCANINT0 or DCANINT1 line, controlled by the interrupt multiplexer register (DCAN INTMUX12 to DCAN INTMUX78), see Section 5.14.1.17.

### *5.4.2 Status Change Interrupts*

The events WakeUpPnd, RxOk, TxOk and LEC in error and status register ( DCAN ES) belong to the status change interrupts. The status change interrupt group can be enabled by SIEbit in the CAN control register.

If SIE is set, a status change interrupt will be generated at each CAN frame, independent of bus errors or valid CAN communication, and also independent of the message RAM configuration.

Status change interrupts can only be routed to interrupt line DCAN0INT, which has to be enabled by setting IE0 in the CAN control register.

**NOTE:** Reading the error and status register will clear the WakeUpPnd flag. If in global power-down mode, the WakeUpPnd flag is cleared by such a read access before the DCAN module has been waken up by the system, the DCAN may re-assert the WakeUpPnd flag, and a second interrupt may occur.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.4.3 Error Interrupts

The events PER, BOff and EWarn, monitored in the Error and Status register, DCAN ES, belong to the error interrupts. The error interrupt group can be enabled by setting bit EIE in the CAN Control register.

Error interrupts can only be routed to interrupt line DCAN0INT, which has to be enabled by setting IE0 in the CAN Control register.

**Figure 5-8. CAN Interrupt Topology 1**

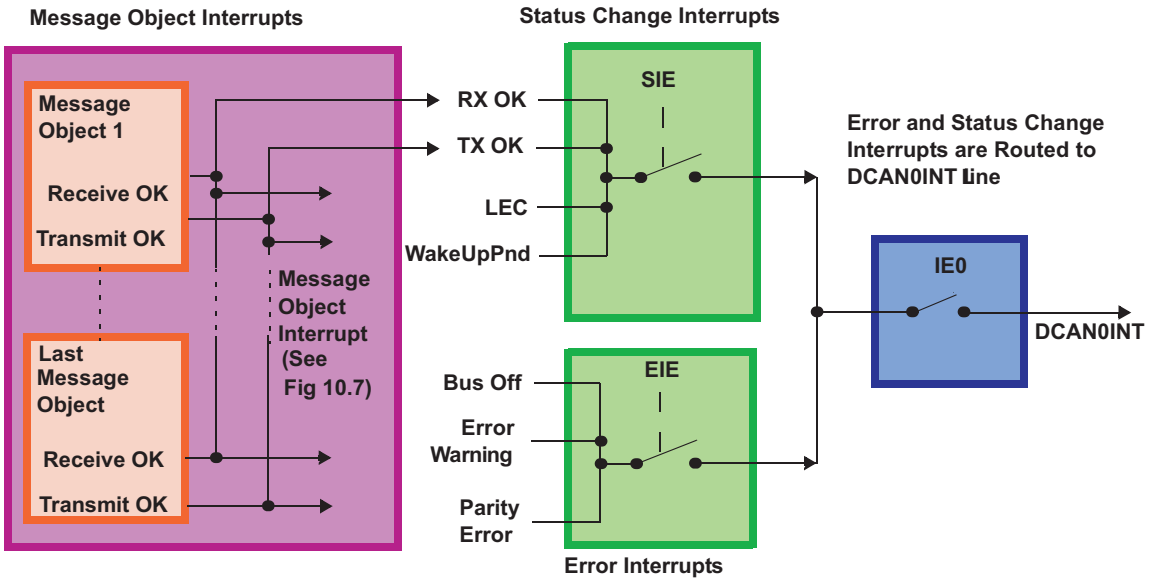
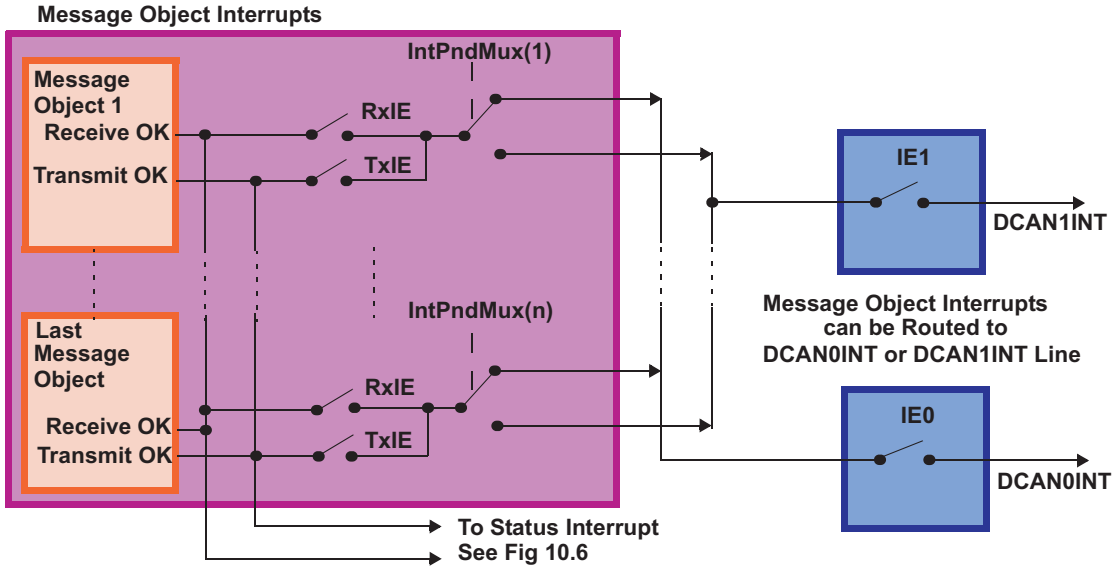



**Figure 5-9. CAN Interrupt Topology 2**





Copyright © 2011–2016, Texas Instruments Incorporated

## 5.5 Local Power-Down Mode

The DCAN supports a local power-down mode, which can be controlled within the DCAN control registers.

### 5.5.1 Entering Local Power-Down Mode

The local power-down mode is requested by setting the PDR bit in CAN Control register.

The DCAN then finishes all transmit requests of the message objects. When all requests are done, DCAN waits until a bus idle state is recognized. Then it will automatically set the Init bit in CAN control register to prevent any further CAN transfers, and it will also set the PDA bit in CAN error and status register. With setting the PDA bits, the DCAN module indicates that the local power-down mode has been entered.

During local power-down mode, the internal clocks of the DCAN module are turned off, but there is a wakeup logic (see Section 5.5.2) that can be active, if enabled. Also, the actual contents of the control registers can be read back.

**NOTE:** In local low-power mode, the application should not clear the Init bit while PDR is set. If there are any messages in the message RAM which are configured as transmit messages and the application resets the init bit, these messages may get sent.

### 5.5.2 Wakeup From Local Power Down

There are two ways to wake up the DCAN from local power-down mode:

- The application could wake up the DCAN module manually by clearing the PDR bit and then clearing the Init bit in CAN Control register.
- Alternatively, a CAN bus activity detection circuit can be activated by setting the wakeup on bus activity bit ( WUBA) in the CAN control register. If this circuit is active, on occurrence of a dominant CAN bus level, the DCAN will automatically start the wakeup sequence. It will clear the PDR bit in the CAN Control register and also clear the PDA bit in the error and status register. The WakeUpPnd bit in CAN error and status register will be set. If status interrupts are enabled, also an interrupt will be generated. Finally the Init bit in CAN control register will be cleared.

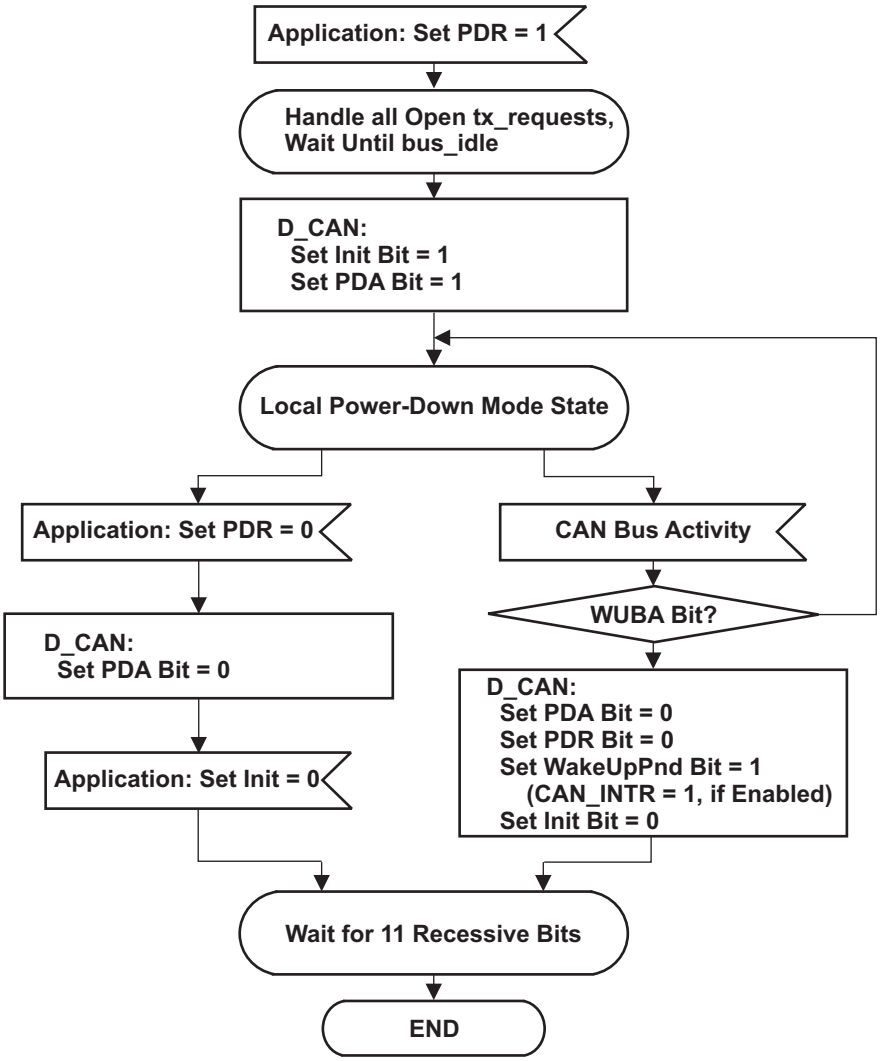
After the Init bit has been cleared, the module waits until it detects 11 consecutive recessive bits on the CAN_RX pin and then goes bus-active again.

**NOTE:** The CAN transceiver circuit has to stay active in order to detect any CAN bus activity while the DCAN is in local power down mode. The first CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power-down and automatic wake-up mode, is lost.

Figure 5-10 shows a flow diagram about entering and leaving local power-down mode.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 5-10. Local Power-Down Mode Flow Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

## 5.6 Parity Check Mechanism

The DCAN provides a parity check mechanism to ensure data integrity of message RAM data. For each word (32 bits) in message RAM, one parity bit will be calculated. The formation of the different words is according to the message RAM representation in RDA mode, see Section 5.12.4.

Parity information is stored in the message RAM on write accesses and will be checked against the stored parity bit from message RAM on read accesses.

The Parity check functionality can be enabled or disabled by PMD bit field in the CAN control register.

In case of disabled parity check, the parity bits in message RAM will be left unchanged on write access to data area and no check will be done on read access.

If parity checking is enabled, parity bits will be automatically generated and checked by the DCAN. The parity bits could be read in debug/suspend mode (see Section 5.12.3) or in RDA mode (see Section 5.12.4). However, direct write access to the parity bits is only possible in this two modes with parity check disabled.

A parity bit will be set, if the modulo-2-sum of the data bits is 1. This definition is equivalent to: The parity bit will be set, if the number of 1 bits in the data is odd.

**NOTE:** The parity scheme is tied to even parity at the device level.

### 5.6.1 Behavior on Parity Error

On any read access to message RAM (e.g., during start of a CAN frame transmission), the parity of the message object will be checked. If a parity error is detected, the PER bit in the error and status register will be set. If error interrupts are enabled, an interrupt would also be generated. In order to avoid the transmission of invalid data over the CAN bus, the MsgVal bit of the message object will be reset.

The message object data can be read by the host CPU, independently of parity errors. Thus, the application has to ensure that the read data is valid, e.g., by immediately checking the parity error code register (DCAN PERR) on parity error interrupt.

**NOTE:** During RAM initialization, no parity check will be done, but if the PMD bit is set, the parity bits will be generated.

### 5.6.2 Parity Testing

Testing the parity mechanism can be done by enabling the bit RamDirectAccess ( RDA) and manually writing the parity bits directly to the dedicated RAM locations. With this, data and parity bits could be checked when reading directly from RAM.

**NOTE:** If parity check was disabled, the application has to ensure correct parity bit handling in order to prevent parity errors later on when parity check is enabled.


Copyright © 2011–2016, Texas Instruments Incorporated

## 5.7 Debug/Suspend Mode

The module supports the usage of an external debug unit by providing functions like pausing DCAN activities and making message RAM content accessible via OCP interface.

Before entering debug/suspend mode, the circuit will either wait until a started transmission or reception will be finished and bus idle state is recognized, or immediately interrupt a current transmission or reception. This is depending on bit IDS in the CAN control register.

Afterwards, the DCAN enters debug/suspend mode, indicated by InitDbg flag in the CAN control register.

During debug/suspend mode, all DCAN registers can be accessed. Reading reserved bits will return '0'. Writing to reserved bits will have no effect.

Also, the message RAM will be memory mapped. This allows the external debug unit to read the message RAM. For the memory organization, see Section 5.12.3).

**NOTE:** During debug/suspend mode, the message RAM cannot be accessed via the IFx register sets.

**NOTE:** Writing to control registers in debug/suspend mode may influence the CAN state machine and further message handling.

For debug support, the auto clear functionality of the following DCAN registers is disabled:

- Error and status register (clear of status flags by read)
- IF1/IF2 command registers (clear of DMAActive flag by read/write)

## 5.8 Configuration of Message Objects

The whole message RAM should be configured before the end of the initialization, however it is also possible to change the configuration of message objects during CAN communication.

The CAN software driver library should offer subroutines that:

- Transfer a complete message structure into a message object. (Configuration)
- Transfer the data bytes of a message into a message object and set TxRqst and NewDat. (Start a new transmission)
- Get the data bytes of a message from a message object and clear NewDat (and IntPnd). (Read received data)
- Get the complete message from a message object and clear NewDat (and IntPnd). (Read a received message, including identifier, from a message object with UMask = '1')

Parameters of the subroutines are the Message Number and a pointer to a complete message structure or to the data bytes of a message structure.

Two examples of assigning the IFx interface register sets to these subroutines are shown here:

In the first method, the tasks of the application program that may access the module are divided into two groups. Each group is restricted to the use of one of the interface register sets. The tasks of one group may interrupt tasks of the other group, but not of the same group.

In the second method, which may be a special case of the first method, there are only two tasks in the application program that access the module. A Read_Message task that uses IF2 or IF3 to get received messages from the message RAM and a Write_Message task that uses IF1 to write messages to be transmitted (or to be configured) into the message RAM. Both tasks may interrupt each other.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.8.1 Configuration of a Transmit Object for Data Frames

Table 5-1 shows how a transmit object can be initialized.

**Table 5-1. Initialization of a Transmit Object**

| MsgVal | Arb | Data | Mask | EoB | Dir | NewDat | MsgLst | RxIE | TxIE | IntPnd | RmtEn | TxRqst |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | appl. | appl. | appl. | 1 | 1 | 0 | 0 | 0 | appl. | 0 | appl. | 0 |

The arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of the outgoing message. If an 11-bit identifier (standard frame) is used (Xtd = '0'), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored.

The data registers (DLC[3:0] and Data0-7) are given by the application. TxRqst and RmtEn should not be set before the data is valid.

If the TxIE bit is set, the IntPnd bit will be set after a successful transmission of the message object.

If the RmtEn bit is set, a matching received remote frame will cause the TxRqst bit to be set; the remote frame will autonomously be answered by a data frame.

The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask='1') to allow groups of remote frames with similar identifiers to set the TxRqst bit. The Dir bit should not be masked. For details, see Section 5.9.8. Identifier masking must be disabled (UMask = '0') if no remote frames are allowed to set the TxRqst bit (RmtEn = '0').

### 5.8.2 Configuration of a Transmit Object for Remote Frames

It is not necessary to configure transmit objects for the transmission of remote frames. Setting TxRqst for a receive object causes the transmission of a remote frame with the same identifier as the data frame for which this receive object is configured.

### 5.8.3 Configuration of a Single Receive Object for Data Frames

Table 5-2 shows how a receive object for data frames can be initialized.

**Table 5-2. Initialization of a single Receive Object for Data Frames**

| MsgVal | Arb | Data | Mask | EoB | Dir | NewDat | MsgLst | RxIE | TxIE | IntPnd | RmtEn | TxRqst |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | appl. | appl. | appl. | 1 | 0 | 0 | 0 | appl. | 0 | 0 | 0 | 0 |

The arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of accepted received messages. If an 11-bit Identifier (standard frame) is used (Xtd = '0'), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a data frame with an 11-bit Identifier is received, ID[17:0] is set to '0'.

The data length code (DLC[3:0]) is given by the application. When the message handler stores a data frame in the message object, it will store the received data length code and eight data bytes. If the data length code is less than 8, the remaining bytes of the message object may be overwritten by non specified values.

The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask = '1') to allow groups of data frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. If some bits of the mask bits are set to "don't care," the corresponding bits of the arbitration register will be overwritten by the bits of the stored data frame.

If the RxIE bit is set, the IntPnd bit will be set when a received data frame is accepted and stored in the message object.

If the TxRqst bit is set, the transmission of a remote frame with the same identifier as actually stored in the arbitration bits will be triggered. The content of the arbitration bits may change if the mask bits are used (UMask = '1') for acceptance filtering.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.8.4 Configuration of a Single Receive Object for Remote Frames

Table 5-3 shows how a receive object for remote frames can be initialized.

**Table 5-3. Initialization of a Single Receive Object for Remote Frames**

| MsgVal | Arb | Data | Mask | EoB | Dir | NewDat | MsgLst | RxIE | TxIE | IntPnd | RmtEn | TxRqst |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | appl. | appl. | appl. | 1 | 1 | 0 | 0 | appl. | 0 | 0 | 0 | 0 |

A receive object for remote frames may be used to monitor remote frames on the CAN bus. The remote frame stored in the receive object will not trigger the transmission of a data frame. Receive objects for remote frames may be expanded to a FIFO buffer (see Section 5.8.5).

UMask must be set to '1.' The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be set to "must-match" or to "don't care," to allow groups of remote frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. For details, see Section 5.9.8.

The arbitration bits (ID[28:0] and Xtd bit) may be given by the application. They define the identifier and type of accepted received remote frames. If some bits of the mask bits are set to "don't care," the corresponding bits of the arbitration bits will be overwritten by the bits of the stored remote frame. If an 11-bit Identifier (standard frame) is used (Xtd = '0'), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a remote frame with an 11-bit Identifier is received, ID[17:0] will be set to '0.'

The data length code (DLC[3:0]) may be given by the application. When the message handler stores a remote frame in the message object, it will store the received data length code. The data bytes of the message object will remain unchanged.

If the RxIE bit is set, the IntPnd bit will be set when a received remote frame is accepted and stored in the message object.

### 5.8.5 Configuration of a FIFO Buffer

With the exception of the EoB bit, the configuration of receive objects belonging to a FIFO buffer is the same as the configuration of a single receive object.

To concatenate multiple message objects to a FIFO buffer, the identifiers and masks (if used) of these message objects have to be programmed to matching values. Due to the implicit priority of the message objects, the message object with the lowest number will be the first message object of the FIFO buffer. The EoB bit of all message objects of a FIFO buffer except the last one have to be programmed to zero. The EoB bits of the last message object of a FIFO Buffer is set to one, configuring it as the end of the block.


Copyright © 2011–2016, Texas Instruments Incorporated

## 5.9 Message Handling

When initialization is finished, the DCAN module synchronizes itself to the traffic on the CAN bus. It does acceptance filtering on received messages and stores those frames that are accepted into the designated message objects. The application has to update the data of the messages to be transmitted and to enable and request their transmission. The transmission is requested automatically when a matching remote frame is received.

The application may read messages which are received and accepted. Messages that are not read before the next messages is accepted for the same message object will be overwritten.

Messages may be read interrupt-driven or after polling of NewDat.

### 5.9.1 Message Handler Overview

The message handler state machine controls the data transfer between the Rx/Tx shift register of the CAN core and the message RAM. It performs the following tasks:

- Data transfer from message RAM to CAN core (messages to be transmitted)
- Data transfer from CAN core to the message RAM (received messages)
- Data transfer from CAN core to the acceptance filtering unit
- Scanning of message RAM for a matching message object (acceptance filtering)
- Scanning the same message object after being changed by IF1/IF2 registers when priority is the same or higher as the message the object found by last scanning
- Handling of TxRqst flags
- Handling of interrupt flags

The message handler registers contains status flags of all message objects grouped into the following topics:

- Transmission Request Flags
- New Data Flags
- Interrupt Pending Flags
- Message Valid Registers

Instead of collecting above listed status information of each message object via IFx registers separately, these message handler registers provides a fast and easy way to get an overview, for example, about all pending transmission requests.

All message handler registers are read-only.

### 5.9.2 Receive/Transmit Priority

The receive/transmit priority for the message objects is attached to the message number, not to the CAN identifier. Message object 1 has the highest priority, while the last implemented message object has the lowest priority. If more than one transmission request is pending, they are serviced due to the priority of the corresponding message object so messages with the highest priority, for example, can be placed in the message objects with the lowest numbers.

The acceptance filtering for received data frames or remote frames is also done in ascending order of message objects, so a frame that has been accepted by a message object cannot be accepted by another message object with a higher message number. The last message object may be configured to accept any data frame or remote frame that was not accepted by any other message object, for nodes that need to log the complete message traffic on the CAN bus.

### 5.9.3 Transmission of Messages in Event Driven CAN Communication

If the shift register of the CAN core is ready for loading and if there is no data transfer between the IFx registers and message RAM, the MsgVal bits in the Message Valid register (DCAN MSGVAL12 to DCAN MSGVAL78) and the TxRqst bits in the transmission request register are evaluated. The valid message object with the highest priority pending transmission request is loaded into the shift register by the message handler and the transmission is started. The message object's NewDat bit is reset.


Copyright © 2011–2016, Texas Instruments Incorporated

After a successful transmission and if no new data was written to the message object (NewDat = ‘0’) since the start of the transmission, the TxRqst bit will be reset. If TxIE is set, IntPnd will be set after a successful transmission. If the DCAN has lost the arbitration or if an error occurred during the transmission, the message will be retransmitted as soon as the CAN bus is free again. If meanwhile the transmission of a message with higher priority has been requested, the messages will be transmitted in the order of their priority.

If automatic retransmission mode is disabled by setting the DAR bit in the CAN control register, the behavior of bits TxRqst and NewDat in the Message Control register of the interface register set is as follows:

- When a transmission starts, the TxRqst bit of the respective interface register set is reset, while bit NewDat remains set.
- When the transmission has been successfully completed, the NewDat bit is reset.

When a transmission failed (lost arbitration or error) bit NewDat remains set. To restart the transmission, the application has to set TxRqst again.

Received remote frames do not require a receive object. They will automatically trigger the transmission of a data frame, if in the matching transmit object the RmtEn bit is set.

### 5.9.4 Updating a Transmit Object

The CPU may update the data bytes of a transmit object any time via the IF1/IF2 interface registers, neither d nor TxRqst have to be reset before the update.

Even if only a part of the data bytes are to be updated, all four bytes in the corresponding IF1/IF2 Data A register (DCAN IF1DATA/DCAN IF2DATA) or IF1/IF2 Data B register (DCAN IF1DATB/DCAN IF2DATB) have to be valid before the content of that register is transferred to the message object. Either the CPU has to write all four bytes into the IF1/IF2 data register or the message object is transferred to the IF1/IF2 data register before the CPU writes the new data bytes.

When only the data bytes are updated, first 0x87 can be written to bits [23:16] of the IF1/IF2 Command register (DCAN IF1CMD/DCAN IF2CMD) and then the number of the message object is written to bits [7:0] of the command register, concurrently updating the data bytes and setting TxRqst with NewDat.

To prevent the reset of TxRqst at the end of a transmission that may already be in progress while the data is updated, NewDat has to be set together with TxRqst in event driven CAN communication. For details, see Section 5.9.3.

When NewDat is set together with TxRqst, NewDat will be reset as soon as the new transmission has started.

### 5.9.5 Changing a Transmit Object

If the number of implemented message objects is not sufficient to be used as permanent message objects only, the transmit objects may be managed dynamically. The CPU can write the whole message (arbitration, control, and data) into the interface register. The bits [23:16] of the command register can be set to 0xB7 for the transfer of the whole message object content into the message object. Before changing the configuration of a message object, MsgVal has to be reset.

If a previously requested transmission of this message object is not completed but already in progress, it will be continued; however, it will not be repeated if it is disturbed.

To only update the data bytes of a message to be transmitted, bits [23:16] of the command register should be set to 0x87.

**NOTE:** After the update of the transmit object, the interface register set will contain a copy of the actual contents of the object, including the part that had not been updated.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.9.6 Acceptance Filtering of Received Messages

When the arbitration and control bits (Identifier + IDE + RTR + DLC) of an incoming message are completely shifted into the shift register of the CAN core, the message handler starts the scan of the message RAM for a matching valid message object:

- The acceptance filtering unit is loaded with the arbitration bits from the CAN core shift register.
- Then the arbitration and mask bits (including MsgVal, UMask, NewDat, and EoB) of message object 1 are loaded into the acceptance filtering unit and are compared with the arbitration bits from the shift register. This is repeated for all following message objects until a matching message object is found, or until the end of the message RAM is reached.
- If a match occurs, the scanning is stopped and the message handler proceeds depending on the type of the frame (data frame or remote frame) received.

### 5.9.7 Reception of Data Frames

The message handler stores the message from the CAN core shift register into the respective message object in the message RAM. Not only the data bytes, but all arbitration bits and the data length code are stored into the corresponding message object. This ensures that the data bytes stay associated to the identifier even if arbitration mask registers are used.

The NewDat bit is set to indicate that new data (not yet seen by the CPU) has been received. The CPU should reset the NewDat bit when it reads the message object. If at the time of the reception the NewDat bit was already set, MsgLst is set to indicate that the previous data (supposedly not seen by the CPU) is lost. If the RxIE bit is set, the IntPnd bit is set, causing the interrupt register to point to this message object.

The TxRqst bit of this message object is reset to prevent the transmission of a remote frame, while the requested data frame has just been received.

### 5.9.8 Reception of Remote Frames

When a remote frame is received, three different configurations of the matching message object have to be considered:

- Dir = ‘1’ (direction = transmit), RmtEn = ‘1’, UMask = ‘1’ or ‘0’

  The TxRqst bit of this message object is set at the reception of a matching remote frame. The rest of the message object remains unchanged.
- Dir = ‘1’ (direction = transmit), RmtEn = ‘0’, UMask = ‘0’

  The remote frame is ignored, this message object remains unchanged.
- Dir = ‘1’ (direction = transmit), RmtEn = ‘0’, UMask = ‘1’

  The remote frame is treated similar to a received data frame. At the reception of a matching Remote Frame, the TxRqst bit of this message object is reset. The arbitration and control bits (Identifier + IDE + RTR + DLC) from the shift register are stored in the message object in the message RAM and the NewDat bit of this message object is set. The data bytes of the message object remain unchanged.

### 5.9.9 Reading Received Messages

The CPU may read a received message any time via the IFx interface register. The data consistency is guaranteed by the message handler state machine. Typically the CPU will write first 0x7F to bits [23:16] and then the number of the message object to bits [7:0] of the command register. That combination will transfer the entire received message from the message RAM into the interface register set. Additionally, the bits NewDat and IntPnd are cleared in the message RAM (not in the interface register set). The values of these bits in the message control register always reflect the status before resetting the bits. If the message object uses masks for acceptance filtering, the arbitration bits show which of the different matching messages has been received.

The actual value of NewDat shows whether a new message has been received since last time when this message object was read. The actual value of MsgLst shows whether more than one message has been received since the last time when this message object was read. MsgLst will not be automatically reset.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.9.10 Requesting New Data for a Receive Object

By means of a remote frame, the CPU may request another CAN node to provide new data for a receive object. Setting the TxRqst bit of a receive object will cause the transmission of a remote frame with the identifier of the receive object. This remote frame triggers the other CAN node to start the transmission of the matching data frame. If the matching data frame is received before the remote frame could be transmitted, the TxRqst bit is automatically reset. Setting the TxRqst bit without changing the contents of a message object requires the value 0x84 in bits [23:16] of the command register.

### 5.9.11 Storing Received Messages in FIFO Buffers

Several message objects may be grouped to form one or more FIFO buffers. Each FIFO buffer configured to store received messages with a particular (group of) identifier(s). arbitration and mask registers of the FIFO buffer's message objects are identical. The end of buffer (EoB) bits of all but the last of the FIFO buffer's message objects are '0'; in the last one the EoB bit is '1.'

Received messages with identifiers matching to a FIFO buffer are stored into a message object of this FIFO buffer, starting with the message object with the lowest message number. When a message is stored into a message object of a FIFO buffer, the NewDat bit of this message object is set. By setting NewDat while EoB is '0', the message object is locked for further write accesses by the message handler until the CPU has cleared the NewDat bit.

Messages are stored into a FIFO buffer until the last message object of this FIFO buffer is reached. If none of the preceding message objects is released by writing NewDat to '0,' all further messages for this FIFO buffer will be written into the last message object of the FIFO buffer (EoB = '1') and therefore overwrite previous messages in this message object.

### 5.9.12 Reading From a FIFO Buffer

Several messages may be accumulated in a set of message objects which are concatenated to form a FIFO buffer before the application program is required (in order to avoid the loss of data) to empty the buffer.

A FIFO buffer of length N will store N –1 plus A FIFO buffer of length N will store –1 plus the last received message since last time it was cleared.

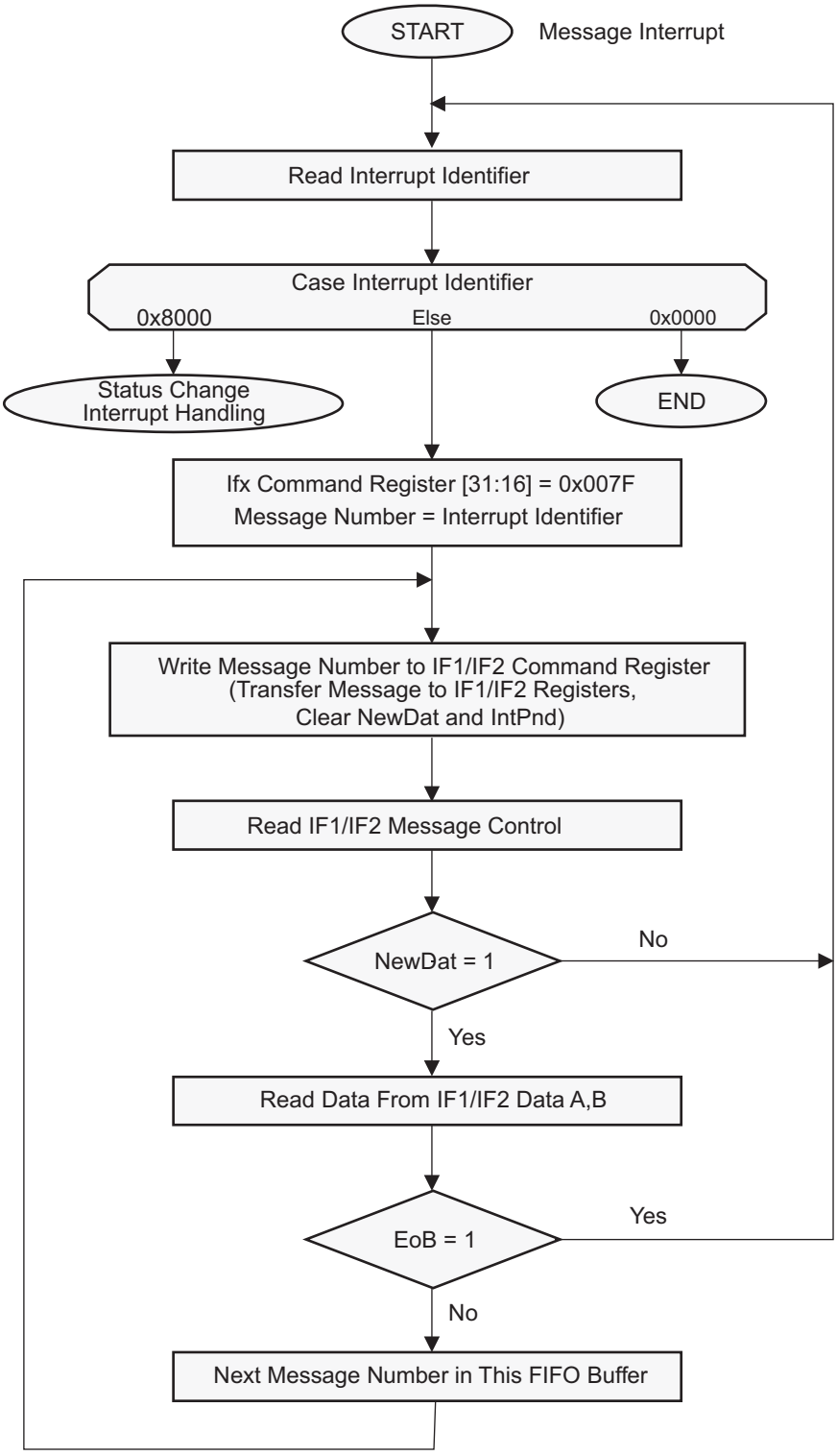
A FIFO buffer is cleared by reading and resetting the NewDat bits of all its message objects, starting at the FIFO Object with the lowest message number. This should be done in a subroutine following the example shown in Figure 5-11.

**NOTE:** All message objects of a FIFO buffer needs to be read and cleared before the next batch of messages can be stored. Otherwise, true FIFO functionality can not be guaranteed, since the message objects of a partly read buffer will be re-filled according to the normal (descending) priority.

Reading from a FIFO buffer message object and resetting its NewDat bit is handled the same way as reading from a single message object.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 5-11. CPU Handling of a FIFO Buffer (Interrupt Driven)**





Copyright © 2011–2016, Texas Instruments Incorporated

## 5.10 CAN Bit Timing

The DCAN supports bit rates between less than 1 kBit/s and 1000 kBit/s.

Each member of the CAN network has its own clock generator, typically derived from a crystal oscillator. The bit timing parameters can be configured individually for each CAN node, creating a common bit rate even though the CAN nodes' oscillator periods ($f_{osc}$) may be different.

The frequencies of these oscillators are not absolutely stable. Small variations are caused by changes in temperature or voltage and by deteriorating components. As long as the variations remain inside a specific oscillator tolerance range (df), the CAN nodes are able to compensate for the different bit rates by resynchronizing to the bit stream.

In many cases, the CAN bit synchronization will amend a faulty configuration of the CAN bit timing to such a degree that only occasionally an error frame is generated. In the case of arbitration however, when two or more CAN nodes simultaneously try to transmit a frame, a misplaced sample point may cause one of the transmitters to become error passive.

The analysis of such sporadic errors requires a detailed knowledge of the CAN bit synchronization inside a CAN node and of the CAN nodes' interaction on the CAN bus.

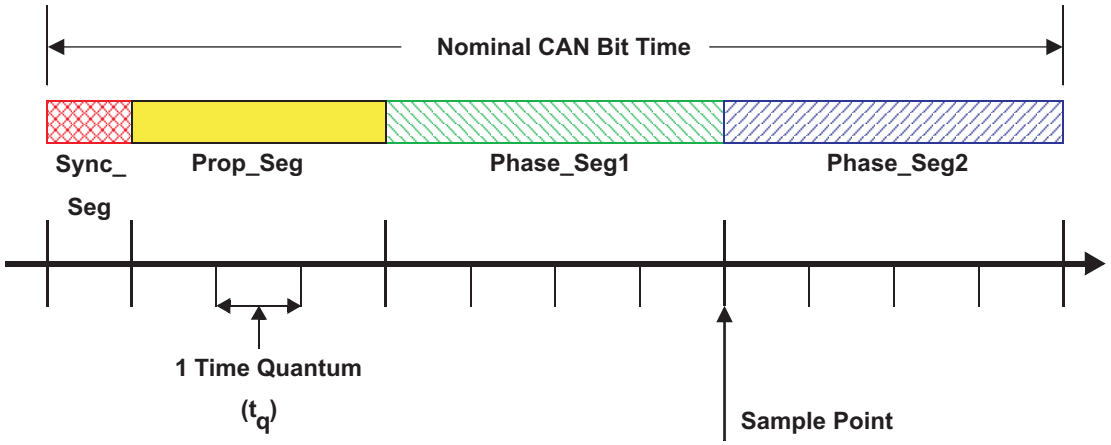
Even if minor errors in the configuration of the CAN bit timing do not result in immediate failure, the performance of a CAN network can be reduced significantly.

### 5.10.1 Bit Time and Bit Rate

According to the CAN specification, the bit time is divided into four segments (see Figure 5-12):

- Synchronization segment (Sync_Seg)
- Propagation time segment (Prop_Seg)
- Phase buffer segment 1 (Phase_Seg1)
- Phase buffer segment 2 (Phase_Seg2)

**Figure 5-12. Bit Timing**




Each segment consists of a specific number of time quanta. The length of one time quantum ($t_q$), which is the basic time unit of the bit time, is given by the CAN_CLK and the baud rate prescalers (BRPE and BRP). With these two baud rate prescalers combined, divider values from 1 to 1024 can be programmed:

$t_q$ = Baud Rate Prescaler / CAN_CLK

Apart from the fixed length of the synchronization segment, these numbers are programmable. Table 5-4 describes the minimum programmable ranges required by the CAN protocol.

A given bit rate may be met by different bit time configurations.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-4. Parameters of the CAN Bit Time**

| Parameter | Range | Remark |
|---|---|---|
| Sync_Seg | 1 $t_q$ (fixed) | Synchronization of bus input to CAN_CLK |
| Prop_Seg | [1 … 8] $t_q$ | Compensates for the physical delay times |
| Phase_Seg1 | [1 … 8] $t_q$ | May be lengthened temporarily by synchronization |
| Phase_Seg2 | [1 … 8] $t_q$ | May be shortened temporarily by synchronization |
| Synchronization Jump Width (SJW) | [1 … 4] $t_q$ | May not be longer than either Phase Buffer Segment |

**NOTE:** For proper functionality of the CAN network, the physical delay times and the oscillator's tolerance range have to be considered.

#### 5.10.1.1 Synchronization Segment

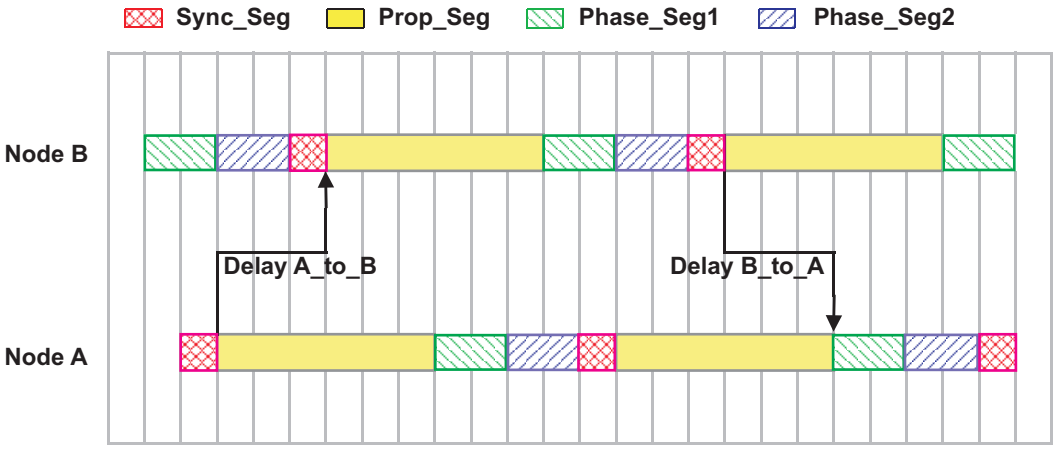
The synchronization segment (Sync_Seg) is the part of the bit time where edges of the CAN bus level are expected to occur. If an edge occurs outside of Sync_Seg, its distance to the Sync_Seg is called the phase error of this edge.

#### 5.10.1.2 Propagation Time Segment

This part of the bit time is used to compensate physical delay times within the CAN network. These delay times consist of the signal propagation time on the bus and the internal delay time of the CAN nodes.

Any CAN node synchronized to the bit stream on the CAN bus can be out of phase with the transmitter of the bit stream, caused by the signal propagation time between the two nodes. The CAN protocol's non-destructive bitwise arbitration and the dominant acknowledge bit provided by receivers of CAN messages require that a CAN node transmitting a bit stream must also be able to receive dominant bits transmitted by other CAN nodes that are synchronized to that bit stream. The example in Figure 5-13 shows the phase shift and propagation times between two CAN nodes.

**Figure 5-13. The Propagation Time Segment**




**Delay A_to_B >= node output delay(A) + bus line delay(AÆB) + node input delay(B)**

**Prop_Seg >= Delay A_to_B + Delay B_to_A**

**Prop_Seg >= 2 • [max(node output delay+bus line delay + node input delay)]**


Copyright © 2011–2016, Texas Instruments Incorporated

In this example, both nodes A and B are transmitters performing an arbitration for the CAN bus. The node A has sent its start of frame bit less than one bit time earlier than node B, therefore node B has synchronized itself to the received edge from recessive to dominant. Since node B has received this edge delay(A_to_B) after it has been transmitted, node B's bit timing segments are shifted with regard to node A. Node B sends an identifier with higher priority and so it will win the arbitration at a specific identifier bit when it transmits a dominant bit while node A transmits a recessive bit. The dominant bit transmitted by node B will arrive at node A after the delay (B_to_A).

Due to oscillator tolerances, the actual position of node A's sample point can be anywhere inside the nominal range of node A's Phase Buffer Segments, so the bit transmitted by node B must arrive at node A before the start of Phase_Seg1. This condition defines the length of Prop_Seg.

If the edge from recessive to dominant transmitted by node B would arrive at node A after the start of Phase_Seg1, it could happen that node A samples a recessive bit instead of a dominant bit, resulting in a bit error and the destruction of the current frame by an error flag.

This error only occurs when two nodes arbitrate for the CAN bus which have oscillators of opposite ends of the tolerance range and are separated by a long bus line; this is an example of a minor error in the bit timing configuration (Prop_Seg too short) that causes sporadic bus errors.

Some CAN implementations provide an optional 3 Sample Mode. The DCAN does not. In this mode, the CAN bus input signal passes a digital low-pass filter, using three samples and a majority logic to determine the valid bit value. This results in an additional input delay of 1 $t_q$, requiring a longer Prop_Seg.

#### 5.10.1.3 Phase Buffer Segments and Synchronization

The phase buffer segments (Phase_Seg1 and Phase_Seg2) and the synchronization jump width (SJW) are used to compensate for the oscillator tolerance.

The phase buffer segments surround the sample point and may be lengthened or shortened by synchronization.

The synchronization jump width (SJW) defines how far the resynchronizing mechanism may move the sample point inside the limits defined by the phase buffer segments to compensate for edge phase errors.

Synchronizations occur on edges from recessive to dominant. Their purpose is to control the distance between edges and sample points.

Edges are detected by sampling the actual bus level in each time quantum and comparing it with the bus level at the previous sample point. A synchronization may be done only if a recessive bit was sampled at the previous sample point and if the actual time quantum's bus level is dominant.

An edge is synchronous if it occurs inside of Sync_Seg; otherwise, its distance to the Sync_Seg is the edge phase error, measured in time quanta. If the edge occurs before Sync_Seg, the phase error is negative, else it is positive.

Two types of synchronization exist: hard synchronization and resynchronizing. A hard synchronization is done once at the start of a frame; inside a frame, only resynchronization is possible.

- Hard Synchronization

  After a hard synchronization, the bit time is restarted with the end of Sync_Seg, regardless of the edge phase error. Thus hard synchronization forces the edge which has caused the hard synchronization, to lie within the synchronization segment of the restarted bit time.

- Bit Resynchronizations

  Resynchronization leads to a shortening or lengthening of the Bit time such that the position of the sample point is shifted with regard to the edge.

  When the phase error of the edge which causes resynchronization is positive, Phase_Seg1 is lengthened. If the magnitude of the phase error is less than SJW, Phase_Seg1 is lengthened by the magnitude of the phase error, else it is lengthened by SJW.

  When the phase error of the edge which causes Resynchronization is negative, Phase_Seg2 is shortened. If the magnitude of the phase error is less than SJW, Phase_Seg2 is shortened by the magnitude of the phase error, else it is shortened by SJW.


Copyright © 2011–2016, Texas Instruments Incorporated

If the magnitude of the phase error of the edge is less than or equal to the programmed value of SJW, the results of hard synchronization and resynchronization are the same. If the magnitude of the phase error is larger than SJW, the resynchronization cannot compensate the phase error completely, and an error of (phase error - SJW) remains.

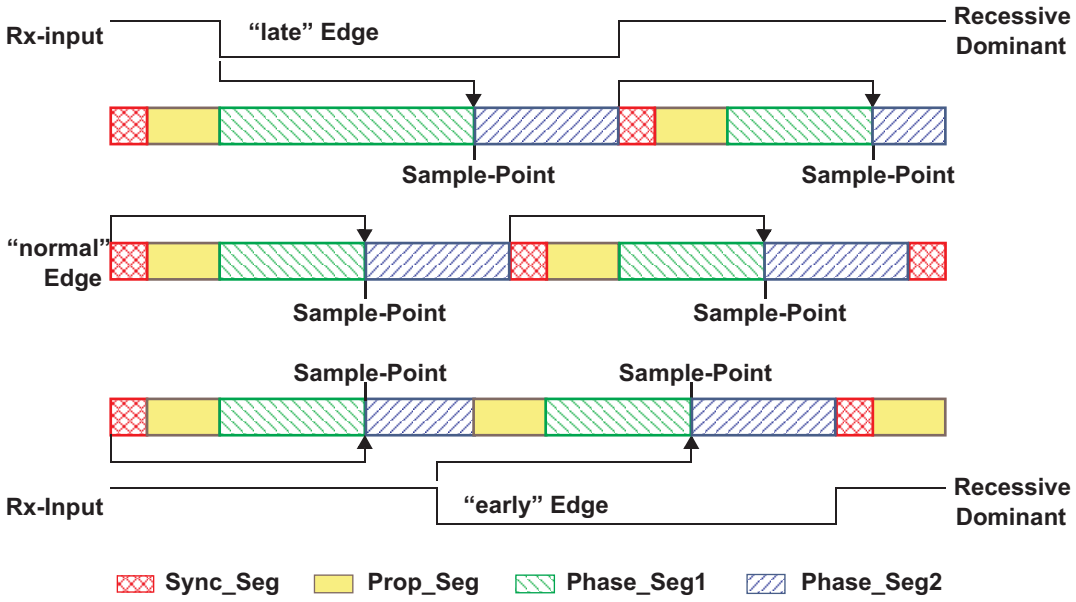
Only one synchronization may be done between two sample points. The synchronizations maintain a minimum distance between edges and sample points, giving the bus level time to stabilize and filtering out spikes that are shorter than (Prop_Seg + Phase_Seg1).

Apart from noise spikes, most synchronizations are caused by arbitration. All nodes synchronize “hard” on the edge transmitted by the “leading” transceiver that started transmitting first, but due to propagation delay times, they cannot become ideally synchronized. The leading transmitter does not necessarily win the arbitration; therefore, the receivers have to synchronize themselves to different transmitters that subsequently take the lead and that are differently synchronized to the previously leading transmitter. The same happens at the acknowledge field, where the transmitter and some of the receivers will have to synchronize to that receiver that takes the lead in the transmission of the dominant acknowledge bit.

Synchronizations after the end of the arbitration will be caused by oscillator tolerance, when the differences in the oscillator’s clock periods of transmitter and receivers sum up during the time between synchronizations (at most ten bits). These summarized differences may not be longer than the SJW, limiting the oscillator’s tolerance range.

Figure 5-14 shows how the phase buffer segments are used to compensate for phase errors. There are three drawings of each two consecutive bit timings. The upper drawing shows the synchronization on a “late” edge, the lower drawing shows the synchronization on an “early” edge, and the middle drawing is the reference without synchronization.

**Figure 5-14. Synchronization on Late and Early Edges**




In the first example, an edge from recessive to dominant occurs at the end of Prop_Seg. The edge is "late" since it occurs after the Sync_Seg. Reacting to the late edge, Phase_Seg1 is lengthened so that the distance from the edge to the sample point is the same as it would have been from the Sync_Seg to the sample point if no edge had occurred. The phase error of this late edge is less than SJW, so it is fully compensated and the edge from dominant to recessive at the end of the bit, which is one nominal bit time long, occurs in the Sync_Seg.

In the second example, an edge from recessive to dominant occurs during Phase_Seg2. The edge is "early" since it occurs before a Sync_Seg. Reacting to the early edge, Phase_Seg2 is shortened and Sync_Seg is omitted, so that the distance from the edge to the sample point is the same as it would have been from a Sync_Seg to the sample point if no edge had occurred. As in the previous example, the magnitude of this early edge’s phase error is less than SJW, so it is fully compensated.


Copyright © 2011–2016, Texas Instruments Incorporated

The phase buffer segments are lengthened or shortened temporarily only; at the next bit time, the segments return to their nominal programmed values.

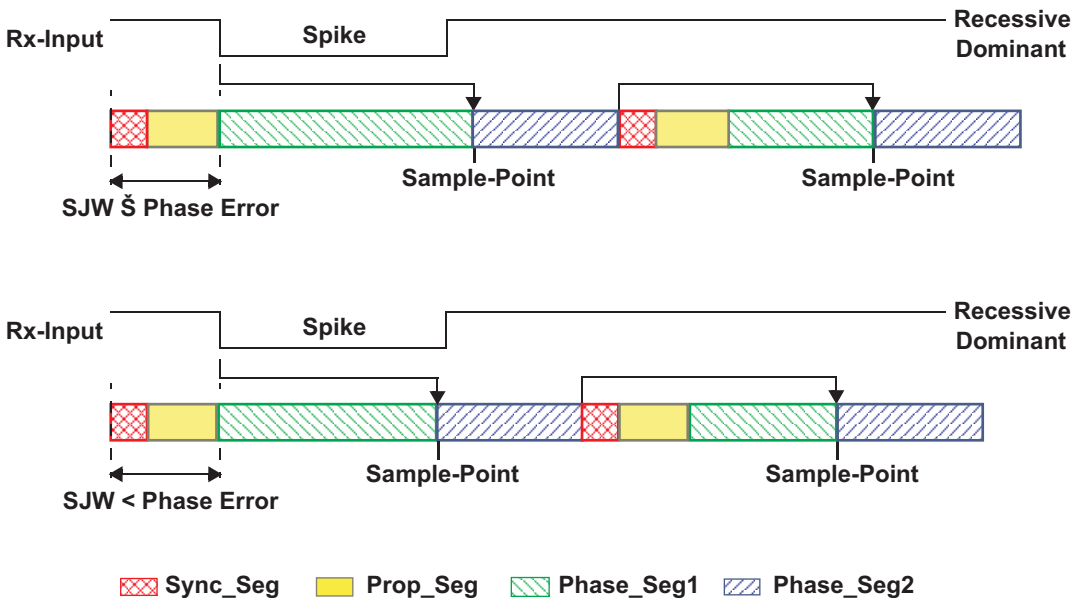
In these examples, the bit timing is seen from the point of view of the CAN implementation's state machine, where the bit time starts and ends at the sample points. The state machine omits Sync_Seg when synchronizing on an early edge because it cannot subsequently redefine that time quantum of Phase_Seg2 where the edge occurs to be the Sync_Seg.

Figure 5-15 shows how short dominant noise spikes are filtered by synchronizations. In both examples, the spike starts at the end of Prop_Seg and has the length of (Prop_Seg + Phase_Seg1).

In the first example, the synchronization jump width is greater than or equal to the phase error of the spike's edge from recessive to dominant. Therefore the sample point is shifted after the end of the spike; a recessive bus level is sampled.

In the second example, SJW is shorter than the phase error, so the sample point cannot be shifted far enough; the dominant spike is sampled as actual bus level.

**Figure 5-15. Filtering of Short Dominant Spikes**




### 5.10.1.4 Oscillator Tolerance Range

With the introduction of CAN protocol version 1.2, the option to synchronize on edges from dominant to recessive became obsolete. Only edges from recessive to dominant are considered for synchronization. The protocol update to version 2.0 (A and B) had no influence on the oscillator tolerance.

The tolerance range df for an oscillator's frequency $f_{osc}$ around the nominal frequency $f_{nom}$ with:

$$(1 - df) \bullet f_{nom} \le f_{osc} \le (1 + df) \bullet f_{nom}$$

depends on the proportions of Phase_Seg1, Phase_Seg2, SJW, and the bit time. The maximum tolerance df is the defined by two conditions (both shall be met):

$$\text{I:} \quad df \le \frac{\min(TSeg1, TSeg2)}{2x\left(13x\left(bit_time - TSeg2\right)\right)}$$

$$\text{II:} \quad df \le \frac{SJW}{20xbit_time}$$

It has to be considered that SJW may not be larger than the smaller of the phase buffer segments and that the propagation time segment limits that part of the bit time that may be used for the phase buffer segments.

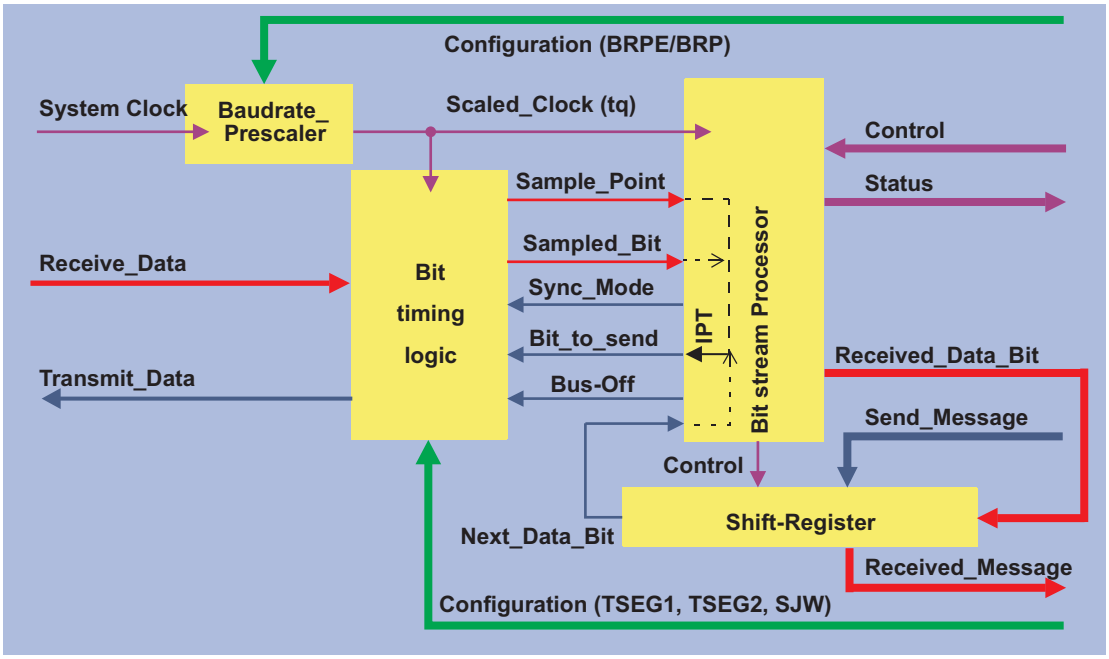

Copyright © 2011–2016, Texas Instruments Incorporated

The combination Prop_Seg = 1 and Phase_Seg1 = Phase_Seg2 = SJW = 4 allows the largest possible oscillator tolerance of 1.58%. This combination with a propagation time segment of only 10% of the bit time is not suitable for short bit times; it can be used for bit rates of up to 125 kBit/s (bit time = 8 µs) with a bus length of 40 m.

### 5.10.2 DCAN Bit Timing Registers

In the DCAN, the bit timing configuration is programmed in two register bytes, additionally a third byte for a baud rate prescaler extension of four bits ( BRPE) is provided. The sum of Prop_Seg and Phase_Seg1 (as TSEG1) is combined with Phase_Seg2 (as TSEG2) in one byte, SJW and BRP (plus BRPE in third byte) are combined in the other byte (see Figure 5-16).

**Figure 5-16. Structure of the CAN Core's CAN Protocol Controller**




In this bit timing register, the components TSEG1, TSEG2, SJW and BRP have to be programmed to a numerical value that is one less than its functional value; so instead of values in the range of [1…n], values in the range of [0…n–1] are programmed. That way, for example, SJW (functional range of [1…4]) is represented by only two bits.

Therefore, the length of the bit time is (programmed values) [TSEG1 + TSEG2 + 3] $t_q$ or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] $t_q$.

The data in the bit timing register ( BTR) is the configuration input of the CAN protocol controller. The baud rate prescaler (configured by BRPE/BRP) defines the length of the time quantum (the basic time unit of the bit time); the bit timing logic (configured by TSEG1, TSEG2, and SJW) defines the number of time quanta in the bit time.

The processing of the bit time, the calculation of the position of the sample point, and occasional synchronizations are controlled by the bit timing state machine, which is evaluated once each time quantum. The rest of the CAN protocol controller, the bit stream processor (BSP) state machine, is evaluated once each bit time, at the sample point.

The shift register serializes the messages to be sent and parallelizes received messages. Its loading and shifting is controlled by the BSP. The BSP translates messages into frames and vice versa. It generates and discards the enclosing fixed format bits, inserts and extracts stuff bits, calculates and checks the CRC code, performs the error management, and decides which type of synchronization is to be used. It is evaluated at the sample point and processes the sampled bus input bit. The time after the sample point that is needed to calculate the next bit to be sent (for example, data bit, CRC bit, stuff bit, error flag, or idle) is called the information processing time (IPT), which is 0 tq for the DCAN.


Copyright © 2011–2016, Texas Instruments Incorporated

Generally, the IPT is CAN controller-specific, but may not be longer than 2 tq. The IPC length is the lower limit of the programmed length of Phase_Seg2. In case of a synchronization, Phase_Seg2 may be shortened to a value less than IPT, which does not affect bus timing.

#### 5.10.2.1 Calculation of the Bit Timing Parameters

Usually, the calculation of the bit timing configuration starts with a desired bit rate or bit time. The resulting bit time (1 / Bit rate) must be an integer multiple of the CAN clock period.

**NOTE:** 8 MHz is the minimum CAN clock frequency required to operate the DCAN at a bit rate of 1 MBit/s.

The bit time may consist of 8 to 25 time quanta. The length of the time quantum $t_q$ is defined by the baud rate prescaler with $t_q$ = (Baud Rate Prescaler) / CAN_CLK. Several combinations may lead to the desired bit time, allowing iterations of the following steps.

First part of the bit time to be defined is the Prop_Seg. Its length depends on the delay times measured in the system. A maximum bus length as well as a maximum node delay has to be defined for expandible CAN bus systems. The resulting time for Prop_Seg is converted into time quanta (rounded up to the nearest integer multiple of $t_q$).

The Sync_Seg is 1 $t_q$ long (fixed), leaving (bit time – Prop_Seg – 1) $t_q$ for the two Phase Buffer Segments. If the number of remaining $t_q$ is even, the Phase Buffer Segments have the same length, Phase_Seg2 = Phase_Seg1, else Phase_Seg2 = Phase_Seg1 + 1.

The minimum nominal length of Phase_Seg2 has to be regarded as well. Phase_Seg2 may not be shorter than any CAN controller's Information Processing Time in the network, which is device dependent and can be in the range of [0…2] $t_q$.

The length of the synchronization jump width is set to its maximum value, which is the minimum of four (4) and Phase_Seg1.

The oscillator tolerance range necessary for the resulting configuration is calculated by the formulas given in Section 5.10.2.3.

If more than one configurations are possible to reach a certain Bit rate, it is recommended to choose the configuration which allows the highest oscillator tolerance range.

CAN nodes with different clocks require different configurations to come to the same bit rate. The calculation of the propagation time in the CAN network, based on the nodes with the longest delay times, is done once for the whole network.

The CAN system's oscillator tolerance range is limited by the node with the lowest tolerance range.

The calculation may show that bus length or bit rate have to be decreased or that the oscillator frequencies' stability has to be increased in order to find a protocol compliant configuration of the CAN bit timing.

The resulting configuration is written into the bit timing register:

Tseg2 = Phase_Seg2-1

Tseg1 = Phase_Seg1+ Prop_Seg-1

SJW = SynchronizationJumpWidth-1

BRP = Prescaler-1


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.10.2.2 Example for Bit Timing at High Baud Rate

In this example, the frequency of CAN_CLK is 10 MHz, BRP is 0, the bit rate is 1 MBit/s.

| | | | |
|---|---|---|---|
| $t_q$ | 100 ns | = | $t_{CAN_CLK}$ |
| delay of bus driver | 90 ns | = | |
| delay of receiver circuit | 40 ns | = | |
| delay of bus line (40 m) | 220 ns | = | |
| $t_{Prop}$ | 700 ns | = | INT (2*delays + 1) = 7 • $t_q$ |
| $t_{SJW}$ | 100 ns | = | 1 • $t_q$ |
| $t_{TSeg1}$ | 800 ns | = | $t_{Prop}$ + $t_{SJW}$ |
| $t_{TSeg2}$ | 100 ns | = | Information Processing Time + 1 • $t_q$ |
| $t_{Sync\text{-}Seg}$ | 100 ns | = | 1 • $t_q$ |
| bit time | 1000 ns | = | $t_{Sync\text{-}Seg}$ + $t_{TSeg1}$ + $t_{TSeg2}$ |
| tolerance for CAN_CLK | 0.43 % | = | $\frac{(\min(TSeg1, TSeg2))}{2x(13x(bit_time - TSeg2))}$ |
| | | = | $\frac{0.1\,\mu s}{2x(13x(1\,\mu s - 0.1\,\mu s))}$ |

In this example, the concatenated bit time parameters are $(1\text{-}1)_3\&(8\text{-}1)_4\&(1\text{-}1)_2\&(1\text{-}1)_6$, so the bit timing register is programmed to = 0x00000700.

### 5.10.2.3 Example for Bit Timing at Low Baud Rate

In this example, the frequency of CAN_CLK is 2 MHz, BRP is 1, the bit rate is 100 KBit/s.

| | | | |
|---|---|---|---|
| $t_q$ | 1 µs | = | $t_{CAN_CLK}$ |
| Delay of bus driver | 200 ns | = | |
| Delay of receiver circuit | 80 ns | = | |
| Delay of bus line (40 m) | 220 ns | = | |
| $t_{Prop}$ | 1 µs | = | 1 • $t_q$ |
| $t_{SJW}$ | 4 µs | = | 4 • $t_q$ |
| $t_{TSeg1}$ | 5 µs | = | $t_{Prop}$ + $t_{SJW}$ |
| $t_{TSeg2}$ | 3 µs | = | Information Processing Time + 3 • $t_q$ |
| $t_{Sync\text{-}Seg}$ | 1 µs | = | 1 • $t_q$ |
| Bit time | 9 µs | = | $t_{Sync\text{-}Seg}$ + $t_{TSeg1}$ + $t_{TSeg2}$ |
| Tolerance for CAN_CLK | | = | $\frac{(\min(TSeg1, TSeg2))}{2x(13x(bit_time - TSeg2))}$ |
| | | = | $\frac{0.1\,\mu s}{2x(13x(1\,\mu s - 0.1\,\mu s))}$ |

In this example, the concatenated bit time parameters are $(3\text{-}1)_3\&(5\text{-}1)_4\&(4\text{-}1)_2\&(2\text{-}1)_6$, so the bit timing register is programmed to = 0x000024C1.


Copyright © 2011–2016, Texas Instruments Incorporated

## 5.11 Message Interface Register Sets

The interface register sets control the CPU read and write accesses to the message RAM. There are two interface registers sets for read/write access, IF1 and IF2 and one interface register set for read access only, IF3.

Due to the structure of the message RAM, it is not possible to change single bits or bytes of a message object. Instead, always a complete message object in the message RAM is accessed. Therefore the data transfer from the IF1/IF2 registers to the message RAM requires the message handler to perform a read-modify-write cycle: First those parts of the message object that are not to be changed are read from the message RAM into the interface register set, and after the update the whole content of the interface register set is written into the message object.

After the partial write of a message object, those parts of the interface register set that are not selected in the command register, will be set to the actual contents of the selected message object.

After the partial read of a message object, those parts of the interface register set that are not selected in the command register, will be left unchanged.

By buffering the data to be transferred, the interface register sets avoid conflicts between concurrent CPU accesses to the message RAM and CAN message reception and transmission. A complete message object (see Section 5.12.1) or parts of the message object may be transferred between the message RAM and the IF1/IF2 register set (see Section 5.14.1.19) in one single transfer. This transfer, performed in parallel on all selected parts of the message object, guarantees the data consistency of the CAN message.

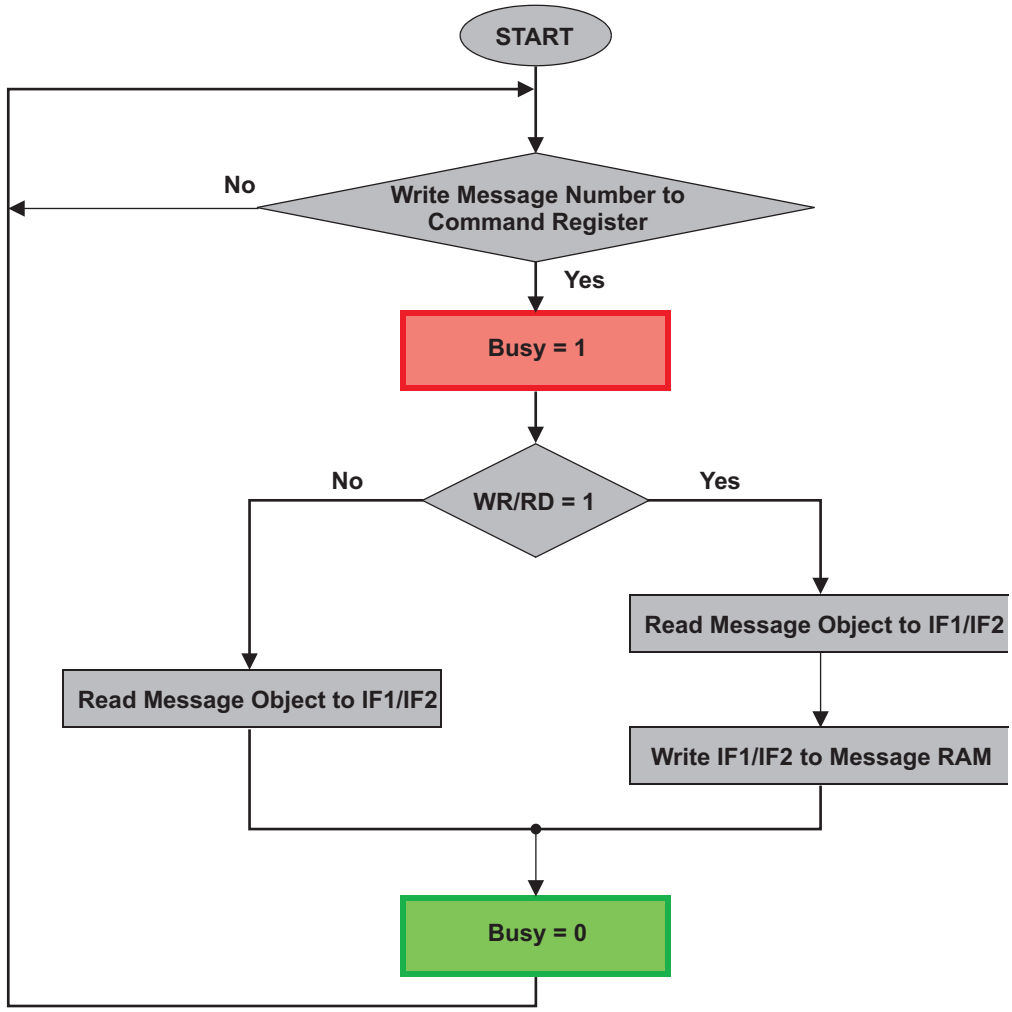
### *5.11.1 Message Interface Register Sets 1 and 2*

The IF1 and IF2 register sets control the data transfer to and from the message object. The command register addresses the desired message object in the message RAM and specifies whether a complete message object or only parts should be transferred. The data transfer is initiated by writing the message number to the bits [7:0] of the command register.

When the CPU initiates a data transfer between the IF1/IF2 registers and message RAM, the message handler sets the busy bit in the respective command register to ‘1’. After the transfer has completed, the busy bit is set back to ‘0’ (see Figure 5-17).


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 5-17. Data Transfer Between IF1/IF2 Registers and Message RAM**




### 5.11.2 IF3 Register Set

The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by CPU. The intention of this feature of IF3 is to provide an interface for the DMA to read packets efficiently. The automatic update functionality can be programmed for each message object (Section 5.14.1.28).

All valid message objects in message RAM which are configured for automatic update, will be checked for active NewDat flags. If such a message object is found, it will be transferred to the IF3 register (if no previous DMA transfers are ongoing), controlled by IF3 Observation register. If more than one NewDat flag is active, the message object with the lowest number has the highest priority for automatic IF3 update.

The NewDat bit in the message object will be reset by a transfer to IF3.

If DCAN internal IF3 update is complete, a DMA request is generated. The DMA request stays active until first read access to one of the IF3 registers. The DMA functionality has to be enabled by setting bit DE3 in CAN control register. Please refer to the device datasheet to find out if this DMA source is available.

**NOTE:** The IF3 register set can not be used for transferring data into message objects.


Copyright © 2011–2016, Texas Instruments Incorporated

## 5.12 Message RAM

**CAUTION**

Writes to the DCAN RAM must not be done while it is in low-power mode. If such writes occur, the behavior is undefined and RAM content is corrupted. It has to be ensured in software that the low-power mode is removed from the DCAN RAM before writing to it.

The DCAN message RAM contains message objects and parity bits for the message objects. There are up to 64 message objects in the message RAM.

During normal operation, accesses to the message RAM are performed via the interface register sets, and the CPU cannot directly access the message RAM.

The interface register sets IF1 and IF2 provide indirect read/write access from the CPU to the message RAM. The IF1 and IF2 register sets can buffer control and user data to be transferred to and from the message objects.

The third interface register set IF3 can be configured to automatically receive control and user data from the message RAM when a message object has been updated after reception of a CAN message. The CPU does not need to initiate the transfer from message RAM to IF3 register set.

The message handler avoids potential conflicts between concurrent accesses to message RAM and CAN frame reception/transmission.

There are two modes where the message RAM can be directly accessed by the CPU:

- Debug/Suspend mode (see Section 5.12.3)
- RAM Direct Access (RDA) mode (see Section 5.12.4)

For the message RAM base address, see the device-specific data sheet.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.12.1 Structure of Message Objects

Table 5-5 shows the structure of a message object. Table 5-6 provides a description of the fields.

The grayed fields are those parts of the message object which are represented in dedicated registers. For example, the transmit request flags of all message objects are represented in centralized transmit request registers.

**Table 5-5. Structure of a Message Object**

| Message Object | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UMask | Msk[28:0] | MXtd | MDir | EoB | unused | NewDat | MsgLst | RxIE | TxIE | IntPnd | RmtEn | TxRqst |
| MsgVal | ID[28:0] | Xtd | Dir | DLC[3:0] | Data 0 | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 |

**Table 5-6. Field Descriptions**

| Name | Value | Description |
|---|---|---|
| MsgVal | | Message valid. The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets the Init bit in the CAN control register. |
| | 0 | The message object is ignored by the message handler. |
| | 1 | The message object is to be used by the message handler. |
| | | Note: This bit may be kept at level '1' even when the identifier bits ID[28:0], the control bits Xtd, Dir, or the data length code are changed. It should be reset if the Messages Object is no longer required. |
| UMask | | Use acceptance mask |
| | 0 | Mask bits (Msk[28:0], MXtd and MDir) are ignored and not used for acceptance filtering. |
| | 1 | Mask bits are used for acceptance filtering. |
| | | Note: If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one. |
| ID[28:0] | | Message identifier |
| | ID[28:0] | 29-bit ("extended") identifier bits |
| | ID[28:18] | 11-bit ("standard") identifier bits |
| Msk[28:0] | | Identifier mask |
| | 0 | The corresponding bit in the message identifier is not used for acceptance filtering (don't care). |
| | 1 | The corresponding bit in the message identifier is used for acceptance filtering. |
| Xtd | | Extended identifier |
| | 0 | The 11-bit ("standard") identifier will be used for this message object. |
| | 1 | The 29-bit ("extended") identifier will be used for this message object. |
| MXtd | | Mask extended identifier |
| | 0 | The extended identifier bit (IDE) has no effect on the acceptance filtering. |
| | 1 | The extended identifier bit (IDE) is used for acceptance filtering. |
| | | Note: When 11-bit ("standard") Identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered. |
| Dir | | Message direction |
| | 0 | Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, the message is stored in this message object. |
| | 1 | Direction = transmit: On TxRqst, a data frame is transmitted. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = one). |
| MDir | | Mask message direction |
| | 0 | The message direction bit (Dir) has no effect on the acceptance filtering. |
| | 1 | The message direction bit (Dir) is used for acceptance filtering. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-6. Field Descriptions (continued)**

| Name | Value | Description |
|---|---|---|
| EOB | | End of block |
| | 0 | The message object is part of a FIFO Buffer block and is not the last message object of this FIFO Buffer block. |
| | 1 | The message object is a single message object or the last message object in a FIFO Buffer Block. |
| | | Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one. |
| NewDat | | New data |
| | 0 | No new data has been written into the data bytes of this message object by the message handler since the last time when this flag was cleared by the CPU. |
| | 1 | The message handler or the CPU has written new data into the data bytes of this message object. |
| MsgLst | | Message lost (only valid for message objects with direction = receive) |
| | 0 | No message was lost since the last time when this bit was reset by the CPU. |
| | 1 | The message handler stored a new message into this message object when NewDat was still set, so the previous message has been overwritten. |
| RxIE | | Receive interrupt enable |
| | 0 | IntPnd will not be triggered after the successful reception of a frame. |
| | 1 | IntPnd will be triggered after the successful reception of a frame. |
| TxIE | | Transmit interrupt enable |
| | 0 | IntPnd will not be triggered after the successful transmission of a frame. |
| | 1 | IntPnd will be triggered after the successful transmission of a frame. |
| IntPnd | | Interrupt pending |
| | 0 | This message object is not the source of an interrupt. |
| | 1 | This message object is the source of an interrupt. The interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority. |
| RmtEn | | Remote enable |
| | 0 | At the reception of a remote frame, TxRqst is not changed. |
| | 1 | At the reception of a remote frame, TxRqst is set. |
| TxRqst | | Transmit request |
| | 0 | This message object is not waiting for a transmission. |
| | 1 | The transmission of this message object is requested and is not yet done. |
| DLC[3:0] | | Data length code |
| | 0-8 | Data frame has 0-8 data bits. |
| | 9-15 | Data frame has 8 data bytes. |
| | | Note: The data length code of a message object must be defined to the same value as in the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message. |
| Data 0 | | 1st data byte of a CAN data frame |
| Data 1 | | 2nd data byte of a CAN data frame |
| Data 2 | | 3rd data byte of a CAN data frame |
| Data 3 | | 4th data byte of a CAN data frame |
| Data 4 | | 5th data byte of a CAN data frame |
| Data 5 | | 6th data byte of a CAN data frame |
| Data 6 | | 7th data byte of a CAN data frame |
| Data 7 | | 8th data byte of a CAN data frame |
| | | Note: Byte Data 0 is the first data byte shifted into the shift register of the CAN core during a reception, byte Data 7 is the last. When the message handler stores a data frame, it will write all the eight data bytes into a message object. If the data length code is less than 8, the remaining bytes of the message object may be overwritten by undefined values. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.12.2 Addressing Message Objects in RAM

The starting location of a particular message object in RAM is:
Message RAM base address + (message object number) * 0x20

This means that message object 1 starts at offset 0x0020; message object 2 starts at offset 0x0040, etc.

‘0’ is not a valid message object number.

The base address for DCAN0 RAM is 0x481C_D000 and DCAN1 RAM is 0x481D_1000.

Message object number 1 has the highest priority.

**Table 5-7. Message RAM addressing in Debug/Suspend and RDA Mode**

| Message Object Number | Offset From Base Address | Word Number | Debug/Suspend Mode, see Section 5.12.3 | RDA mode, see Section 5.12.4 |
|---|---|---|---|---|
| 1 | 0x0020 | 1 | Parity | Data Bytes 4-7 |
| | 0x0024 | 2 | MXtd,MDir,Mask | Data Bytes 0-3 |
| | 0x0028 | 3 | Xtd,Dir,ID | ID[27:0],DLC |
| | 0x002C | 4 | Ctrl | Mask,Xtd,Dir,ID[28] |
| | 0x0030 | 5 | Data Bytes 3-0 | Parity,Ctrl,MXtd,MDir |
| | 0x0034 | 6 | Data Bytes 7-4 | … |
| … | … | … | … | … |
| 31 | 0x03E0 | 1 | Parity | Data Bytes 4-7 |
| | 0x03E4 | 2 | MXtd,MDir,Mask | Data Bytes 0-3 |
| | 0x03E8 | 3 | Xtd,Dir,ID | ID[27:0],DLC |
| | 0x03EC | 4 | Ctrl | Mask,Xtd,Dir,ID[28] |
| | 0x03F0 | 5 | Data Bytes 3-0 | Parity,Ctrl,MXtd,MDir |
| | 0x03F4 | 6 | Data Bytes 7-4 | … |
| … | … | … | … | … |
| 63 | 0x07E0 | 1 | Parity | Data Bytes 4-7 |
| | 0x07E4 | 2 | MXtd,MDir,Mask | Data Bytes 0-3 |
| | 0x07E8 | 3 | Xtd,Dir,ID | ID[27:0],DLC |
| | 0x07EC | 4 | Ctrl | Mask,Xtd,Dir,ID[28] |
| | 0x07F0 | 5 | Data Bytes 3-0 | Parity,Ctrl,MXtd,MDir |
| | 0x07F4 | 6 | Data Bytes 7-4 | … |
| last implemented ( 32(DCAN3) or 64) | 0x0000 | 1 | Parity | Data Bytes 4-7 |
| | 0x0004 | 2 | MXtd,MDir,Mask | Data Bytes 0-3 |
| | 0x0008 | 3 | Xtd,Dir,ID | ID[27:0],DLC |
| | 0x000C | 4 | Ctrl | Mask,Xtd,Dir,ID[28] |
| | 0x0010 | 5 | Data Bytes 3-0 | Parity,Ctrl,MXtd,MDir |
| | 0x0014 | 6 | Data Bytes 7-4 | … |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.12.3 Message RAM Representation in Debug/Suspend Mode

In debug/suspend mode, the message RAM will be memory mapped. This allows the external debug unit to access the message RAM.

**NOTE:** During debug/suspend mode, the message RAM cannot be accessed via the IFx register sets.

**Table 5-8. Message RAM Representation in Debug/Suspend Mode**

| Bit # | 31/15 | 30/14 | 29/13 | 29/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MsgAddr + 0x00 | Reserved | | | | | | | | | | | | | | | |
| | Reserved | | | | | | | | | | | Parity[4:0] | | | | |
| MsgAddr + 0x04 | MXtd | MDir | Rsvd | Msk[28:16] | | | | | | | | | | | | |
| | Msk[15:0] | | | | | | | | | | | | | | | |
| MsgAddr + 0x08 | Rsvd | Xtd | Dir | ID[28:16] | | | | | | | | | | | | |
| | ID[15:0] | | | | | | | | | | | | | | | |
| MsgAddr + 0x0C | Reserved | | | | | | | | | | | | | | | |
| | Rsvd | MsgLst | Rsvd | UMask | TxIE | RxTE | RmtEn | Rsvd | EOB | Reserved | | | DLC[3:0] | | | |
| MsgAddr + 0x10 | Data 3 | | | | | | | | Data 2 | | | | | | | |
| | Data 1 | | | | | | | | Data 0 | | | | | | | |
| MsgAddr + 0x14 | Data 7 | | | | | | | | Data 6 | | | | | | | |
| | Data 5 | | | | | | | | Data 4 | | | | | | | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.12.4 Message RAM Representation in Direct Access Mode

When the RDA bit in the test register is set while the DCAN module is in test mode (test bit in the CAN control register is set), the CPU has direct access to the message RAM. Due to the 32-bit bus structure, the RAM is split into word lines to support this feature. The CPU has access to one word line at a time only.

In RAM direct access mode, the RAM is represented by a continuous memory space within the address frame of the DCAN module, starting at the message RAM base address.

Note: During direct access mode, the message RAM cannot be accessed via the IFx register sets.

Any read or write to the RAM addresses for RamDirectAccess during normal operation mode (TestMode bit or RDA bit not set) will be ignored.

**Table 5-9. Message RAM Representation in RAM Direct Access Mode**

| Bit # | 31/15 | 30/14 | 29/13 | 29/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MsgAddr + 0x00 | Data 4 | | | | | | | | Data 5 | | | | | | | |
| | Data 6 | | | | | | | | Data 7 | | | | | | | |
| MsgAddr + 0x04 | Data 0 | | | | | | | | Data 1 | | | | | | | |
| | Data 2 | | | | | | | | Data 3 | | | | | | | |
| MsgAddr + 0x08 | ID[27:12] | | | | | | | | | | | | | | | |
| | ID[11:0] | | | | | | | | | | | | DLC[3:0] | | | |
| MsgAddr + 0x0C | Msk[28:13] | | | | | | | | | | | | | | | |
| | Msk[12:0] | | | | | | | | | | | | | Xtd | Dir | ID[28] |
| MsgAddr + 0x10 | Reserved | | | | | | | | | | | | | | | Parity [4] |
| | Parity[3:0] | | | | Unused | | | | Msg Lst | UMask | TxIE | RxTE | Rmt En | EOB | MXtd | MDir |

**NOTE:** Writes to unused bits has no effect.

## 5.13 GIO Support

The CAN_RX and CAN_TX pins of the DCAN module can be used as general purpose IO pins, if CAN functionality is not needed. This function is controlled by the CAN TX IO control register (DCAN TIOC) (see Section 5.14.1.29) and the CAN RX IO control register (DCAN RIOC) (see Section 5.14.1.30).

## 5.14 Registers

The base address for the:

- DCAN0 registers is 0x481C_C000
- DCAN1 registers is 0x481D_0000


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1 DCAN Control Registers

After hardware reset, the registers of the DCAN hold the values shown in the register descriptions.

Additionally, the Bus-Off state is reset and the CAN_TX pin is set to recessive (HIGH). The Init bit in the CAN control register is set to enable the software initialization. The DCAN will not influence the CAN bus until the CPU resets Init to '0'.

**Table 5-10. DCAN Control Register Summary Table**

| Offset | Acronym | Description | Section |
|---|---|---|---|
| 0x00 | DCAN CTL | CAN Control Register | Section 5.14.1.1 |
| 0x04 | DCAN ES and PARITYERR_EOI | Error and Status Register<br>Parity Error EOI Register | Section 5.14.1.2 |
| 0x08 | DCAN ERRC | Error Counter Register | Section 5.14.1.3 |
| 0x0C | DCAN BTR | Bit Timing Register | Section 5.14.1.4 |
| 0x10 | DCAN INT | Interrupt Register | Section 5.14.1.5 |
| 0x14 | DCAN TEST | Test Register | Section 5.14.1.6 |
| 0x1C | DCAN PERR | Parity Error Code Register | Section 5.14.1.7 |
| 0x80 | DCAN ABOTR | Auto-Bus-On Time Register | Section 5.14.1.8 |
| 0x84 | DCAN TXRQ X | Transmission Request X Register | Section 5.14.1.9 |
| 0x88 to 0x94 | DCAN TXRQ12 to DCAN TXRQ78 | Transmission Request Registers | Section 5.14.1.10 |
| 0x98 | DCAN NWDAT X | New Data X Register | Section 5.14.1.11 |
| 0x9C to 0xA8 | DCAN NWDAT12 to DCAN NWDAT78 | New Data Registers | Section 5.14.1.12 |
| 0xAC | DCAN INTPND X | Interrupt Pending X Register | Section 5.14.1.13 |
| 0xB0 to 0xBC | DCAN INTPND12 to DCAN INTPND78) | Interrupt Pending Registers | Section 5.14.1.14 |
| 0xC0 | DCAN MSGVAL X | Message Valid X Register | Section 5.14.1.15 |
| 0xC4 to 0xD0 | DCAN MSGVAL12 to DCAN MSGVAL78 | Message Valid Registers | Section 5.14.1.16 |
| 0xD8 to 0xE4 | DCAN INTMUX12 to DCAN INTMUX78 | Interrupt Multiplexer Registers | Section 5.14.1.17 |
| 0x100 | DCAN IF1CMD | IF1 Command Registers | Section 5.14.1.18 |
| 0x120 | DCAN IF2CMD | IF2 Command Registers | Section 5.14.1.18 |
| 0x104 | DCAN IF1MSK | IF1 Mask Register | Section 5.14.1.19 |
| 0x124 | DCAN IF2MSK | IF2 Mask Register | Section 5.14.1.19 |
| 0x108 | DCAN IF1ARB | IF1 Arbitration Register | Section 5.14.1.20 |
| 0x128 | DCAN IF2ARB | IF2 Arbitration Register | Section 5.14.1.20 |
| 0x10C | DCAN IF1MCTL | IF1 Message Control Register | Section 5.14.1.21 |
| 0x12C | DCAN IF2MCTL | IF2 Message Control Register | Section 5.14.1.21 |
| 0x110 | DCAN IF1DATA | IF1 Data A Register | Section 5.14.1.22 |
| 0x114 | DCAN IF1DATB | IF1 Data B Register | Section 5.14.1.22 |
| 0x130 | DCAN IF2DATA | IF2 Data A Register | Section 5.14.1.22 |
| 0x134 | DCAN IF2DATB | IF2 Data B Register | Section 5.14.1.22 |
| 0x140 | DCAN IF3OBS | IF3 Observation Register | Section 5.14.1.23 |
| 0x144 | DCAN IF3MSK | IF3 Mask Register | Section 5.14.1.24 |
| 0x148 | DCAN IF3ARB | IF3 Arbitration Register | Section 5.14.1.25 |
| 0x14C | DCAN IF3MCTL | IF3 Message Control Register | Section 5.14.1.26 |
| 0x150 | DCAN IF3DATA | IF3 Data A Register | Section 5.14.1.27 |
| 0x154 | DCAN IF3DATB | IF3 Data B Register | Section 5.14.1.27 |
| 0x160 to 0x16C | DCAN IF3UPD12 to IF3UPD78 | IF3 Update Enable Registers | Section 5.14.1.28 |
| 0x1E0 | DCAN TIOC | CAN TX IO Control Register | Section 5.14.1.29 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-10. DCAN Control Register Summary Table (continued)**

| Offset | Acronym | Description | Section |
|---|---|---|---|
| 0x1E4 | DCAN RIOC | CAN RX IO Control Register | Section 5.14.1.30 |

### 5.14.1.1 CAN Control Register (DCAN CTL)

The CAN control register (DCAN CTL) is shown in Figure 5-18 and described in Table 5-11.

**Figure 5-18. CAN Control Register (DCAN CTL)**

| 31 | | | | | | | | | | | | | 26 | 25 | 24 | 23 | | | | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | WUBA | PDR | Reserved | | | | | DE3 | DE2 | DE1 | IE1 | InitDbg |
| R-0 | | | | | | | | | | | | | | R/W-0 | R/W-0 | R-0 | | | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 |
| 15 | 14 | 13 | | | | | | | | | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | |
| SWR | Rsvd | PMD | | | | | | | | | | ABO | IDS | Test | CCE | DAR | Rsvd | EIE | SIE | IE0 | Init | | | | |
| R/WP-0 | R-0 | R/W-0x5 | | | | | | | | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-1 | | | | |

LEGEND: R/W = Read/Write; R = Read only; WP = Write protected by init bit; -*n* = value after reset

**Table 5-11. CAN Control Register (DCAN CTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-26 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 26 | WUBA | | Automatic wake up on bus activity when in local power-down mode |
| | | 0 | No detection of a dominant CAN bus level while in local power-down mode. |
| | | 1 | Detection of a dominant CAN bus level while in local power-down mode is enabled. On occurrence of a dominant CAN bus level, the wake up sequence is started (Additional information can be found in *Local Power-Down Mode*). |
| | | | Note: The CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power down and automatic wake-up mode, will be lost. |
| 24 | PDR | | Request for local low power-down mode |
| | | 0 | No application request for local low power-down mode. If the application has cleared this bit while DCAN in local power-down mode, also the Init bit has to be cleared. |
| | | 1 | Local power-down mode has been requested by application. The DCAN will acknowledge the local power-down mode by setting bit PDA in the error and status register. The local clocks will be turned off by DCAN internal logic (Additional information can be found in *Local Power-Down Mode*). |
| 23-21 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 20 | DE3 | | Enable DMA request line for IF3 |
| | | 0 | Disabled |
| | | 1 | Enabled |
| | | | Note: A pending DMA request for IF3 remains active until first access to one of the IF3 registers. |
| 19 | DE2 | | Enable DMA request line for IF2 |
| | | 0 | Disabled |
| | | 1 | Enabled |
| | | | Note: A pending DMA request for IF2 remains active until first access to one of the IF2 registers. |
| 18 | DE1 | | Enable DMA request line for IF1 |
| | | 0 | Disabled |
| | | 1 | Enabled |
| | | | Note: A pending DMA request for IF1 remains active until first access to one of the IF1 registers. |
| 17 | IE1 | | Interrupt line 1 enable |
| | | 0 | Disabled - Module interrupt DCAN1INT is always low. |
| | | 1 | Enabled - interrupts will assert line DCAN1INT to one; line remains active until pending interrupts are processed. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-11. CAN Control Register (DCAN CTL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 16 | InitDbg | | Internal init state while debug access |
| | | 0 | Not in debug mode, or debug mode requested but not entered. |
| | | 1 | Debug mode requested and internally entered; the DCAN is ready for debug accesses. |
| 15 | SWR | | SW reset enable |
| | | 0 | Normal Operation |
| | | 1 | Module is forced to reset state. This bit will automatically get cleared after execution of SW reset after one OCP clock cycle.<br>Note: To execute software reset, the following procedure is necessary:<br>1. Set Init bit to shut down CAN communication.<br>2. Set SWR bit additionally to Init bit. |
| 14 | Reserved | 0 | This bit is always read as 0. Writes have no effect. |
| 13-10 | PMD | | Parity on/off |
| | | 0x5 | Parity function disabled |
| | | Others | Parity function enabled |
| 9 | ABO | | Auto-Bus-On enable |
| | | 0 | The Auto-Bus-On feature is disabled |
| | | 1 | The Auto-Bus-On feature is enabled |
| 8 | IDS | | Interruption debug support enable |
| | | 0 | When Debug/Suspend mode is requested, DCAN will wait for a started transmission or reception to be completed before entering Debug/Suspend mode |
| | | 1 | When Debug/Suspend mode is requested, DCAN will interrupt any transmission or reception, and enter Debug/Suspend mode immediately. |
| 7 | Test | | Test mode enable |
| | | 0 | Normal Operation |
| | | 1 | Test Mode |
| 6 | CCE | | Configuration change enable |
| | | 0 | The CPU has no write access to the configuration registers. |
| | | 1 | The CPU has write access to the configuration registers (when Init bit is set). |
| 5 | DAR | | Disable automatic retransmission |
| | | 0 | Automatic retransmission of not successful messages enabled. |
| | | 1 | Automatic retransmission disabled. |
| 4 | Reserved | 0 | This bit is always read as 0. Writes have no effect. |
| 3 | EIE | | Error interrupt enable |
| | | 0 | Disabled - PER, BOff and EWarn bits can not generate an interrupt. |
| | | 1 | Enabled - PER, BOff and EWarn bits can generate an interrupt at DCAN0INT line and affect the interrupt register. |
| 2 | SIE | | Status change interrupt enable |
| | | 0 | Disabled - WakeUpPnd, RxOk, TxOk and LEC bits can not generate an interrupt. |
| | | 1 | Enabled - WakeUpPnd, RxOk, TxOk and LEC can generate an interrupt at DCAN0INT line and affect the interrupt register. |
| 1 | IE0 | | Interrupt line 0 enable |
| | | 0 | Disabled - Module interrupt DCAN0INT is always low. |
| | | 1 | Enabled - interrupts will assert line DCAN0INT to one; line remains active until pending interrupts are processed. |
| 0 | Init | | Initialization |
| | | 0 | Normal operation |
| | | 1 | Initialization mode is entered |


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** The Bus-Off recovery sequence (refer to CAN specification) cannot be shortened by setting or resetting Init bit. If the module goes Bus-Off, it will automatically set the Init bit and stop all bus activities.

When the Init bit is cleared by the application again, the module will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the bus-off recovery sequence, the error counters will be reset.

After the Init bit is reset, each time when a sequence of 11 recessive bits is monitored, a Bit0 error code is written to the error and status register, enabling the CPU to check whether the CAN bus is stuck at dominant or continuously disturbed, and to monitor the proceeding of the bus-off recovery sequence.

### 5.14.1.2 Error and Status Register (DCAN ES) and DCAN Parity Error EOI Register (PARITYERR_EOI)

The error and status register (DCAN ES) is shown in Figure 5-19 and described in Table 5-12. The DCAN Parity Error EOI register is shown in Figure 5-20 and described in Table 5-13.

**Figure 5-19. Error and Status Register (DCAN ES)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | PDA | Wake UpPnd | PER | BOff | EWarn | EPass | RxOK | TxOK | LEC | |
| R-0 | | R-0 | R/C-0 | R/C-0 | R-0 | R-0 | R-0 | R/C-0 | R/C-0 | R/S-111 | |

LEGEND: R = Read only; S = Set by Read; C = Clear by Read; -*n* = value after reset

**Figure 5-20. Parity Error End of Interrupt Register (PARITYERR_EOI)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 9 | 8 | 0 |
|---|---|---|---|
| Reserved | PARITYERR_EOI | Reserved | |
| R-0 | | R-0 | |

LEGEND: R = Read only; S = Set by Read; C = Clear by Read; -*n* = value after reset

**Table 5-12. Error and Status Register (DCAN ES) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 10 | PDA | | Local power-down mode acknowledge |
| | | 0 | DCAN is not in local power-down mode. |
| | | 1 | Application request for setting DCAN to local power-down mode was successful. DCAN is in local power-down mode. |
| 9 | WakeUp Pnd | | Wake up pending. This bit can be used by the CPU to identify the DCAN as the source to wake up the system. |
| | | 0 | No Wake Up is requested by DCAN. |
| | | 1 | DCAN has initiated a wake up of the system due to dominant CAN bus while module power down. |
| | | | This bit will be reset if error and status register is read. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-12. Error and Status Register (DCAN ES) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 8 | PER | | Parity error detected |
| | | 0 | No parity error has been detected since last read access. |
| | | 1 | The parity check mechanism has detected a parity error in the Message RAM. |
| | | | This bit will be reset if error and status register is read. |
| 7 | BOff | | Bus-Off state |
| | | 0 | The CAN module is not bus-off state. |
| | | 1 | The CAN module is in bus-off state. |
| 6 | EWarn | | Warning state |
| | | 0 | Both error counters are below the error warning limit of 96. |
| | | 1 | At least one of the error counters has reached the error warning limit of 96. |
| 5 | EPass | | Error passive state |
| | | 0 | On CAN Bus error, the DCAN could send active error frames. |
| | | 1 | The CAN core is in the error passive state as defined in the CAN Specification. |
| 4 | RxOk | | Received a message successfully |
| | | 0 | No message has been successfully received since the last time when this bit was read by the CPU. This bit is never reset by DCAN internal events. |
| | | 1 | A message has been successfully received since the last time when this bit was reset by a read access of the CPU (independent of the result of acceptance filtering). |
| | | | This bit will be reset if error and status register is read. |
| 3 | TxOk | | Transmitted a message successfully |
| | | 0 | No message has been successfully transmitted since the last time when this bit was read by the CPU. This bit is never reset by DCAN internal events. |
| | | 1 | A message has been successfully transmitted (error free and acknowledged by at least one other node) since the last time when this bit was reset by a read access of the CPU. |
| | | | This bit will be reset if error and status register is read. |
| 2-0 | LEC | | Last error code. The LEC field indicates the type of the last error on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. |
| | | 0 | No error |
| | | 1 | Stuff error: More than five equal bits in a row have been detected in a part of a received message where this is not allowed. |
| | | 2 | Form error: A fixed format part of a received frame has the wrong format. |
| | | 3 | Ack error: The message this CAN core transmitted was not acknowledged by another node. |
| | | 4 | Bit1 error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant. |
| | | 5 | Bit0 error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (logical value '0'), but the monitored bus level was recessive. During Bus-Off recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus-Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). |
| | | 6 | CRC error: In a received message, the CRC check sum was incorrect. (CRC received for an incoming message does not match the calculated CRC for the received data). |
| | | 7 | No CAN bus event was detected since the last time the CPU read the error and status register. Any read access to the error and status register re-initializes the LEC to value '7.' |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-13. Parity Error End of Interrupt Register (PARITYERR_EOI) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 8 | PARITYERR_ EOI | | End of interrupt indication for parity error |
| | | 0 | Has no effect writing |
| | | 1 | Indicated end of interrupt for parity error |
| 7-0 | Reserved | | These bits are always read as 0. Writes have no effect. |

Interrupts are generated by bits PER, BOff and EWarn (if EIE bit in CAN control register is set) and by bits WakeUpPnd, RxOk, TxOk, and LEC (if SIE bit in CAN control register is set).

A change of bit EPass will not generate an interrupt.

PARITYERR_EOI is used to signal an end of interrupt of the parity error interrupt (dcan_parity) to the IP. PARITYERR_EOI register is a *Write only* register.

A read from this offset address will result in a read from the DCAN ES register, where as a write will result in a write to the PARITYERR_EOI register.

**NOTE:** Reading the error and status register clears the WakeUpPnd, PER, RxOk and TxOk bits and set the LEC to value '7.' Additionally, the status interrupt value (0x8000) in the interrupt register will be replaced by the next lower priority interrupt value. The EOI for all other interrupts (DCANINT0 and DCANINT1) are automatically handled by hardware.

**NOTE:** For debug support, the auto clear functionality of error and status register (clear of status flags by read) is disabled when in debug/suspend mode.

### 5.14.1.3 Error Counter Register (DCAN ERRC)

The error counter register (DCAN ERRC) is shown in Figure 5-21 and described in Table 5-14.

**Figure 5-21. Error Counter Register (DCAN ERRC)**

| 31 | | | | | | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |
| 15 | 14 | | | | | | 8 | 7 | | | | | | | 0 |
| RP | REC[6:0] | | | | | | | TEC[7:0] | | | | | | | |
| R-0 | R-0 | | | | | | | R-0 | | | | | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 5-14. Error Counter Register (DCAN ERRC) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 15 | RP | | Receive error passive |
| | | 0 | The receive error counter is below the error passive level. |
| | | 1 | The receive error counter has reached the error passive level as defined in the CAN specification. |
| 14-8 | REC[6:0] | | Receive error counter. Actual state of the receive error counter (values from 0 to 255). |
| 7-0 | TEC[7:0] | | Transmit error counter. Actual state of the transmit error counter (values from 0 to 255). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.4 Bit Timing Register (DCAN BTR)

The bit timing register (DCAN BTR) is shown in Figure 5-22 and described in Table 5-15.

**Figure 5-22. Bit Timing Register (DCAN BTR)**

| 31 | 20 | 19 | 16 |
|---|---|---|---|
| Reserved | | BRPE | |
| R-0 | | R-0 | |

| 15 | 14 12 | 11 8 | 7 6 | 5 0 |
|---|---|---|---|---|
| Reserved | TSeg2 | TSeg1 | SJW | BRP |
| R-0 | R/WP-0x2 | R/WP-0x3 | R/WP-0 | R/WP-0x1 |

LEGEND: R = Read only; WP = Write protected by CCE bit; *-n* = value after reset

**Table 5-15. Bit Timing Register (DCAN BTR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 19-16 | BRPE | 0x00-0x0F | Baud rate prescaler extension.<br>Valid programmed values are 0 to 15.<br>By programming BRPE the baud rate prescaler can be extended to values up to 1024. |
| 15 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 14-12 | TSeg2 | 0x0-0x7 | Time segment after the sample point<br>Valid programmed values are 0 to 7.<br>The actual TSeg2 value which is interpreted for the bit timing will be the programmed TSeg2 value + 1. |
| 11-8 | TSeg1 | 0x01-0x0F | Time segment before the sample point<br>Valid programmed values are 1 to15.<br>The actual TSeg1 value interpreted for the bit timing will be the programmed TSeg1 value + 1. |
| 7-6 | SJW | 0x0-0x3 | Synchronization Jump Width<br>Valid programmed values are 0 to 3.<br>The actual SJW value interpreted for the synchronization will be the programmed SJW value + 1. |
| 5-0 | BRP | 0x00-0x3F | Baud rate prescaler<br>Value by which the CAN_CLK frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta.<br>Valid programmed values are 0 to 63.<br>The actual BRP value interpreted for the bit timing will be the programmed BRP value + 1. |

**NOTE:** This register is only writable if CCE and Init bits in the CAN control register are set.

**NOTE:** The CAN bit time may be programmed in the range of 8 to 25 time quanta.

**NOTE:** The CAN time quantum may be programmed in the range of 1 to1024 CAN_CLK periods.

With a CAN_CLK of 8 MHz and BRPE = 0x00, the reset value of 0x00002301 configures the DCAN for a bit rate of 500kBit/s.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.5 Interrupt Register (DCAN INT)

The interrupt register (DCAN INT) is shown in Figure 5-23 and described in Table 5-16.

**Figure 5-23. Interrupt Register (DCAN INT)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| Reserved | | Int1ID[7:0] | |
| R-0 | | R-0 | |

| 15 | 0 |
|---|---|
| Int0ID[15:0] | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 5-16. Interrupt Register (DCAN INT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 23-16 | Int1ID[23:16] | | Interrupt 1 Identifier (indicates the message object with the highest pending interrupt) |
| | | 0x00 | No interrupt is pending |
| | | 0x01-0x80 | Number of message object which caused the interrupt. |
| | | 0x81-0xFF | Unused |
| | | | If several interrupts are pending, the CAN interrupt register will point to the pending interrupt with the highest priority. The DCAN1INT interrupt line remains active until Int1ID reaches value 0 (the cause of the interrupt is reset) or until IE1 is cleared. |
| | | | A message interrupt is cleared by clearing the message object's IntPnd bit. |
| | | | Among the message interrupts, the message object's interrupt priority decreases with increasing message number. |
| 15-0 | Int0ID[15:0] | | Interrupt Identifier (the number here indicates the source of the interrupt) |
| | | 0x0000 | No interrupt is pending |
| | | 0x0001-0x0080 | Number of message object which caused the interrupt. |
| | | 0x0081-0x7FFF | Unused |
| | | 0x8000 | Error and status register value is not 0x07. |
| | | 0x8001-0xFFFF | Unused |
| | | | If several interrupts are pending, the CAN interrupt register will point to the pending interrupt with the highest priority. The DCAN0INT interrupt line remains active until Int0ID reaches value 0 (the cause of the interrupt is reset) or until IE0 is cleared. |
| | | | The Status interrupt has the highest priority. Among the message interrupts, the message object's interrupt priority decreases with increasing message number. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.6 Test Register (DCAN TEST)

The test register (DCAN TEST) is shown in Figure 5-24 and described in Table 5-17.

**Figure 5-24. Test Register (DCAN TEST)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | RDA | EXL | Rx | Tx[1:0] | | LBack | Silent | Reserved | | |
| R-0 | | R/WP-0 | R/WP-0 | R-U | R/WP-0 | | R/WP-0 | R/WP-0 | R-0 | | |

LEGEND: R = Read only; WP = Write protected by test bit; *-n* = value after reset; U = Undefined

**Table 5-17. Test Register (DCAN TEST) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 9 | RDA | | RAM direct access enable |
| | | 0 | Normal pperation |
| | | 1 | Direct access to the RAM is enabled while in test mode |
| 8 | EXL | | External loopback mode |
| | | 0 | Disabled |
| | | 1 | Enabled |
| 7 | Rx | | Receive pin. Monitors the actual value of the CAN_RX pin |
| | | 0 | The CAN bus is dominant |
| | | 1 | The CAN bus is recessive |
| 6-5 | Tx[1:0] | | Control of CAN_TX pin |
| | | 00 | Normal operation, CAN_TX is controlled by the CAN core. |
| | | 01 | Sample point can be monitored at CAN_TX pin. |
| | | 10 | CAN_TX pin drives a dominant value. |
| | | 11 | CAN_TX pin drives a recessive value. |
| 4 | LBack | | Loopback mode |
| | | 0 | Disabled |
| | | 1 | Enabled |
| 3 | Silent | | Silent mode |
| | | 0 | Disabled |
| | | 1 | Enabled |
| 2-0 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |

For all test modes, the test bit in CAN control register needs to be set to one. If test bit is set, the RDA, EXL, Tx1, Tx0, LBack and Silent bits are writable. Bit Rx monitors the state of pin CAN_RX and therefore is only readable. All test register functions are disabled when test bit is cleared.

**NOTE:** The test register is only writable if test bit in CAN control register is set.

**NOTE:** Setting Tx[1:0] other than '00' will disturb message transfer.

**NOTE:** When the internal loop-back mode is active (bit LBack is set), bit EXL will be ignored.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.7 Parity Error Code Register (DCAN PERR)

The parity error code register (DCAN PERR) is shown in Figure 5-25 and described in Table 5-18.

**Figure 5-25. Parity Error Code Register (DCAN PERR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 11 | 10 | 8 | 7 | 0 |
|---|---|---|---|---|---|
| Reserved | | Word Number | | Message Number | |
| R-0 | | R-U | | R-U | |

LEGEND: R = Read only; -*n* = value after reset; U = Undefined

**Table 5-18. Parity Error Code Register (DCAN PERR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 10-8 | Word Number | 0x01-0x05 | Word number where parity error has been detected<br>RDA word number (1 to 5) of the message object (according to the message RAM representation in RDA mode). |
| 7-0 | Message Number | 0x01-0x80 | Message object number where parity error has been detected |

If a parity error is detected, the PER flag will be set in the error and status register. This bit is not reset by the parity check mechanism; it must be reset by reading the error and status register.

In addition to the PER flag, the parity error code register will indicate the memory area where the parity error has been detected (message number and word number).

If more than one word with a parity error was detected, the highest word number with a parity error will be displayed.

After a parity error has been detected, the register will hold the last error code until power is removed.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.8 Auto-Bus-On Time Register (DCAN ABOTR)

The Auto-Bus-On time register (DCAN ABOTR) is shown in Figure 5-26 and described in Table 5-19.

**Figure 5-26. Auto-Bus-On Time Register (DCAN ABOTR)**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| ABO Time[31:16] | | ABO Time[15:0] | |
| R/W-0 | | R/W-0 | |

LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 5-19. Auto-Bus-On Time Register (DCAN ABOTR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ABO Time | | Number of OCP clock cycles before a Bus-Off recovery sequence is started by clearing the Init bit.<br>This function has to be enabled by setting bit ABO in CAN control register.<br>The Auto-Bus-On timer is realized by a 32-bit counter that starts to count down to zero when the module goes Bus-Off.<br>The counter will be reloaded with the preload value of the ABO time register after this phase. |

**NOTE:** On write access to the CAN control register while Auto-Bus-On timer is running, the Auto-Bus-On procedure will be aborted.

**NOTE:** During Debug/Suspend mode, running Auto-Bus-On timer will be paused.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.9 Transmission Request X Register (DCAN TXRQ X)

With the transmission request X register (DCAN TXRQ X), the CPU can detect if one or more bits in the different transmission request registers are set. Each register bit represents a group of eight message objects. If at least one of the TxRqst bits of these message objects are set, the corresponding bit in the transmission request X register will be set. The transmission request X register is shown in Figure 5-27.

**Figure 5-27. Transmission Request X Register (DCAN TXRQ X)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TxRqstReg8 | | TxRqstReg7 | | TxRqstReg6 | | TxRqstReg5 | | TxRqstReg4 | | TxRqstReg3 | | TxRqstReg2 | | TxRqstReg1 | |
| R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Example 1.** Bit 0 of the transmission request X register represents byte 0 of the transmission request 1 register. If one or more bits in this byte are set, bit 0 of the transmission request X register will be set.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.10 Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78)

These registers hold the TxRqst bits of the implemented message objects. By reading out these bits, the CPU can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the CPU via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.

The transmission request registers are shown in Table 5-20 and described in Table 5-21.

**Table 5-20. Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78)**

| Offset/Register | 31/15 | 30/14 | 29/13 | 29/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x88 CAN TXRQ12 | TxRqst[32:17] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | TxRqst[16:1] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0x8C CAN TXRQ34 | TxRqst[64:49] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | TxRqst[48:33] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0x90 CAN TXRQ56 | TxRqst[96:81] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | TxRqst[80:65] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0x94 CAN TXRQ78 | TxRqst[128:113] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | TxRqst[112:97] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 5-21. Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | TxRqs[128:1] | | Transmission request bits (for all message objects) |
| | | 0 | No transmission has been requested for this message object. |
| | | 1 | The transmission of this message object is requested and is not yet done. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.11 New Data X Register (DCAN NWDAT X)

With the new data X register, the CPU can detect if one or more bits in the different new data registers are set. Each register bit represents a group of eight message objects. If at least on of the NewDat bits of these message objects are set, the corresponding bit in the new data X register will be set.

The new data X register (DCAN NWDAT X) is shown in Figure 5-28.

**Figure 5-28. Transmission Request X Register (DCAN TXRQ X)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NewDatReg8 | | NewDatReg7 | | NewDatReg6 | | NewDatReg5 | | NewDatReg4 | | NewDatReg3 | | NewDatReg2 | | NewDatReg1 | |
| R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Example 1.** Bit 0 of the new data X register represents byte 0 of the new data 1 register. If one or more bits in this byte are set, bit 0 of the new data X register will be set.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.12 New Data Registers (DCAN NWDAT12 to DCAN NWDAT78)

These registers hold the NewDat bits of the implemented message objects. By reading out these bits, the CPU can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.

The new data registers (DCAN NWDAT12 to DCAN NWDAT78) are shown in Table 5-22 and described in Table 5-23.

**Table 5-22. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78)**

| Offset/Register | 31/15 | 30/14 | 29/13 | 29/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x9C CAN NWDAT12 | NewDat[32:17] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | NewDat[16:1] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xA0 CAN NWDAT34 | NewDat[64:49] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | NewDat[48:33] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xA4 CAN NWDAT56 | NewDat[96:81] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | NewDat[80:65] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xA8 CAN NWDAT78 | NewDat[128:113] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | NewDat[112:97] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 5-23. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | NewDat[128:1] | | New Data Bits (for all message objects) |
| | | 0 | No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. |
| | | 1 | The message handler or the CPU has written new data into the data portion of this message object. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.13 Interrupt Pending X Register (DCAN INTPND X)

With the interrupt pending X register, the CPU can detect if one or more bits in the different interrupt pending registers are set. Each bit of this register represents a group of eight message objects. If at least one of the IntPnd bits of these message objects are set, the corresponding bit in the interrupt pending X register will be set.

The interrupt pending X register (DCAN INTPND X) is shown in Figure 5-29.

**Figure 5-29. Interrupt Pending X Register (DCAN INTPND X)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IntPndReg8 | | IntPndReg7 | | IntPndReg6 | | IntPndReg5 | | IntPndReg4 | | IntPndReg3 | | IntPndReg2 | | IntPndReg1 | |
| R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Example 2.** Bit 0 of the interrupt pending X register represents byte 0 of the interrupt pending 1 register. If one or more bits in this byte are set, bit 0 of the interrupt pending X register will be set.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.14 Interrupt Pending Registers (DCAN INTPND12 to DCAN INTPND78)

These registers hold the IntPnd bits of the implemented message objects. By reading out these bits, the CPU can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

The interrupt pending registers (DCAN INTPND12 to DCAN INTPND78) are shown in Table 5-24 and described in Table 5-25.

**Table 5-24. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78)**

| Offset/Register | 31/15 | 30/14 | 29/13 | 29/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0xB0 CAN INTPND12 | IntPnd[32:17] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | IntPnd[16:1] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xB4 CAN INTPND34 | IntPnd[64:49] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | IntPnd[48:33] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xB8 CAN INTPND56 | IntPnd[96:81] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | IntPnd[80:65] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xBC CAN INTPND78 | IntPnd[128:113] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | IntPnd[112:97] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 5-25. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | IntPnd[128:1] | | Interrupt Pending Bits (for all message objects) |
| | | 0 | This message object is not the source of an interrupt. |
| | | 1 | This message object is the source of an interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.15 Message Valid X Register (DCAN MSGVAL X)

With the message valid X register, the CPU can detect if one or more bits in the different message valid registers are set. Each bit of this register represents a group of eight message objects. If at least one of the MsgVal bits of these message objects are set, the corresponding bit in the message valid X register will be set.

The message valid X register (DCAN MSGVAL X) is shown in Figure 5-30.

**Figure 5-30. Message Valid X Register (DCAN MSGVAL X)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MsgValReg8 | | MsgValReg7 | | MsgValReg6 | | MsgValReg5 | | MsgValReg4 | | MsgValReg3 | | MsgValReg2 | | MsgValReg1 | |
| R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Example 3.** Bit 0 of the message valid X register represents byte 0 of the message valid 1 register. If one or more bits in this byte are set, bit 0 of the message valid X register will be set.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.16 Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78)

These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the CPU can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

The message valid registers (DCAN MSGVAL12 to MSGVAL78) is shown in Table 5-26 and described in Table 5-27.

**Table 5-26. Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78)**

| Offset/Register | 31/15 | 30/14 | 29/13 | 29/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0xC4 CAN MSGVAL12 | MsgVal[32:17] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | MsgVal[16:1] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xC8 CAN MSGVAL34 | MsgVal[64:49] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | MsgVal[48:33] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xCC CAN MSGVAL56 | MsgVal[96:81] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | MsgVal[80:65] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xD0 CAN MSGVAL78 | MsgVal[128:113] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | MsgVal[112:97] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 5-27. Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | MsgVal[128:1] | | Message valid bits (for all message objects) |
| | | 0 | This message object is ignored by the message handler. |
| | | 1 | This message object is configured and will be considered by the message handler. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.17 Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78)

The IntMux flag determine for each message object, which of the two interrupt lines (DCAN0INT or DCAN1INT) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register.

The IntPnd bit of a specific message object can be set or reset by the CPU via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the Int0ID resp Int1ID flags in the interrupt register.

The interrupt multiplexer registers are shown in Table 5-28 and described in Table 5-29.

**Table 5-28. Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78)**

| Offset/Register | 31/15 | 30/14 | 29/13 | 29/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0xD8 CAN INTMUX12 | IntMux[32:17] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | IntMux[16:1] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xDC CAN INTMUX34 | IntMux[64:49] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | IntMux[48:33] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xE0 CAN INTMUX56 | IntMux[96:81] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | IntMux[80:65] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| 0xE4 CAN INTMUX78 | IntMux[128:113] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |
| | IntMux[112:97] | | | | | | | | | | | | | | | |
| | R-0 | | | | | | | | | | | | | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 5-29. Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | IntMux[128:1] | | Multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines (for all message objects) |
| | | 0 | DCAN0INT line is active if corresponding IntPnd flag is one. |
| | | 1 | DCAN1INT line is active if corresponding IntPnd flag is one. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.18 IF1/IF2 Command Registers (DCAN IF1CMD, DCAN IF2CMD)

The IF1/IF2 Command Register (DCAN IF1CMD, DCAN IF2CMD) configure and initiate the transfer between the IF1/IF2 register sets and the message RAM. It is configurable which portions of the message object should be transferred.

A transfer is started when the CPU writes the message number to bits [7:0] of the IF1/IF2 command register. With this write operation, the Busy bit is automatically set to '1' to indicate that a transfer is in progress.

After 4 to 14 OCP clock cycles, the transfer between the interface register and the message RAM will be completed and the Busy bit is cleared. The maximum number of cycles is needed when the message transfer concurs with a CAN message transmission, acceptance filtering, or message storage.

If the CPU writes to both IF1/IF2 command registers consecutively (request of a second transfer while first transfer is still in progress), the second transfer will start after the first one has been completed.

The IF1/IF2 command registers are shown in Figure 5-31 and Figure 5-32 and described in Table 5-30.

**NOTE:** While Busy bit is one, IF1/IF2 register sets are write protected.

**NOTE:** For debug support, the auto clear functionality of the IF1/IF2 command registers (clear of DMAactive flag by r/w) is disabled during Debug/Suspend mode.

**NOTE:** If an invalid Message Number is written to bits [7:0] of the IF1/IF2 command register, the message handler may access an implemented (valid) message object instead.

**Figure 5-31. IF1 Command Registers (DCAN IF1CMD)**

| 31 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | WR/RD | Mask | Arb | Control | Clr IntPnd | TxRqst/ New-Dat | Data A | Data B |
| R-0 | | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 |

| 15 | 14 | 13 | 8 | 7 | 0 |
|---|---|---|---|---|---|
| Busy | DMA active | Reserved | | Message Number | |
| R-0 | R/WP/C-0 | R-0 | | R/WP-0x1 | |

LEGEND: R = Read only; WP = Write protected (protected by busy bit); C = Clear by IF1 access; -*n* = value after reset

**Figure 5-32. IF2 Command Registers (DCAN IF2CMD)**

| 31 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | WR/RD | Mask | Arb | Control | Clr IntPnd | TxRqst/ New-Dat | Data A | Data B |
| R-0 | | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 |

| 15 | 14 | 13 | 8 | 7 | 0 |
|---|---|---|---|---|---|
| Busy | DMA active | Reserved | | Message Number | |
| R-0 | R/WP/C-0 | R-0 | | R/WP-0x1 | |

LEGEND: R = Read only; WP = Write protected (protected by busy bit); C = Clear by IF2 access; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-30. IF1/IF2 Command Registers (DCAN IF1CMD, DCAN IF2CMD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 23 | WR/RD | | Write/Read |
| | | 0 | Direction = Read: Transfer direction is from the message object addressed by Message Number (Bits [7:0]) to the IF1/IF2 register set. |
| | | 1 | Direction = Write: Transfer direction is from the IF1/IF2 register set to the message object addressed by Message Number (Bits [7:0]). |
| 22 | Mask | | Access mask bits |
| | | 0 | Mask bits will not be changed |
| | | 1 | Direction = Read: The mask bits (identifier mask + MDir + MXtd) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the IF1/IF2 register set. |
| | | | Direction = Write: The mask bits (identifier mask + MDir + MXtd) will be transferred from the IF1/IF2 register set to the message object addressed by Message Number (Bits [7:0]). |
| 21 | Arb | | Access arbitration bits |
| | | 0 | Arbitration bits will not be changed |
| | | 1 | Direction = Read: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF1/IF2 register set. |
| | | | Direction = Write: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the IF1/IF2 register set to the message object addressed by Message Number (Bits [7:0]). |
| 20 | Control | | Access control bits |
| | | 0 | Control bits will not be changed |
| | | 1 | Direction = Read: The message control bits will be transferred from the message object addressed by message number (Bits [7:0]) to the IF1/IF2 registerset. |
| | | | Direction = Write: The message control bits will be transferred from the IF1/IF2 registerset to the message object addressed by message number (Bits [7:0]). |
| | | | If the TxRqst/NewDat bit in this register(Bit [18]) is set, the TxRqst/ NewDat bit in the IF1/IF2 message control register will be ignored. |
| 19 | ClrIntPnd | | Clear interrupt pending bit |
| | | 0 | IntPnd bit will not be changed |
| | | 1 | Direction = Read: Clears IntPnd bit in the message object. |
| | | | Direction = Write: This bit is ignored. Copying of IntPnd flag from IF1/IF2 Registers to message RAM can only be controlled by the control flag (Bit [20]). |
| 18 | TxRqst/NewDat | | Access transmission request bit |
| | | 0 | Direction = Read: NewDat bit will not be changed. Direction = Write: TxRqst/NewDat bit will be handled according to the control bit. |
| | | 1 | Direction = Read: Clears NewDat bit in the message object.<br>Direction = Write: Sets TxRqst/NewDat in message object. |
| | | | Note: If a CAN transmission is requested by setting TxRqst/NewDat in this register, the TxRqst/NewDat bits in the message object will be set to one independent of the values in IF1/IF2 message control Register. |
| | | | Note: A read access to a message object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the IF1/IF2 message control registeralways reflect the status before resetting them. |
| 17 | Data A | | Access Data Bytes 0-3 |
| | | 0 | Data Bytes 0-3 will not be changed. |
| | | 1 | Direction = Read: The data bytes 0-3 will be transferred from the message object addressed by the Message Number (Bits [7:0]) to the corresponding IF1/IF2 registerset. |
| | | | Direction = Write: The data bytes 0-3 will be transferred from the IF1/IF2 registerset to the message object addressed by the Message Number (Bits [7:0]). |
| | | | Note: The duration of the message transfer is independent of the number of bytes to be transferred. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-30. IF1/IF2 Command Registers (DCAN IF1CMD, DCAN IF2CMD) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 16 | Data B | | Access Data Bytes 4-7 |
| | | 0 | Data Bytes 4-7 will not be changed. |
| | | 1 | Direction = Read: The data bytes 4-7 will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF1/IF2 registerset. |
| | | | Direction = Write: The data bytes 4-7 will be transferred from the IF1/IF2 registerset to the message object addressed by message number (Bits [7:0]). |
| | | | Note: The duration of the message transfer is independent of the number of bytes to be transferred. |
| 15 | Busy | | Busy flag |
| | | 0 | No transfer between IF1/IF2 register set and message RAM is in progress. |
| | | 1 | Transfer between IF1/IF2 register set and message RAM is in progress. |
| | | | This bit is set to one after the message number has been written to bits [7:0]. IF1/IF2 register set will be write protected. The bit is cleared after read/write action has been finished. |
| 14 | DMAactive | | Activation of DMA feature for subsequent internal IF1/IF2 update |
| | | 0 | DMA request line is independent of IF1/IF2 activities. |
| | | 1 | DMA is requested after completed transfer between IF1/IF2 register set and message RAM. |
| | | | The DMA request remains active until the first read or write to one of the IF1/IF2 registers; an exception is a write to Message Number (Bits [7:0]) when DMAactive is one. |
| | | | Note: Due to the auto reset feature of the DMAactive bit, this bit has to be set for each subsequent DMA cycle separately. |
| 13-8 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 7-0 | Message Number | | Number of message object in message RAM which is used for data transfer |
| | | 0x00 | Invalid message number |
| | | 0x01-0x80 | Valid message numbers |
| | | 0x81-0xFF | Invalid message numbers |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.19 IF1/IF2 Mask Registers (DCAN IF1MSK, DCAN IF2MSK)

The bits of the IF1/IF2 mask registers mirror the mask bits of a message object. The function of the relevant message objects bits is described in *Structure of Message Objects*.

The IF1/IF2 mask registers (DCAN IF1MSK, DCAN IF2MSK) are shown in Figure 5-33 and Figure 5-34 and described in Table 5-31.

**NOTE:** While Busy bit of IF1/IF2 command register is one, IF1/IF2 register set is write protected.

**Figure 5-33. IF1 Mask Register (DCAN IF1MSK)**

| 31 | 30 | 29 | 28 | 16 |
|---|---|---|---|---|
| MXtd | MDir | Rsvd | Msk[28:16] | |
| R/WP-1 | R/WP-1 | R-1 | R/WP-0x1FFF | |

| 15 | 0 |
|---|---|
| Msk[15:0] | |
| R/WP-0xFFFF | |

LEGEND: R = Read only; WP = Write protected (protected by busy bit); -*n* = value after reset

**Figure 5-34. IF2 Mask Register (DCAN IF2MSK)**

| 31 | 30 | 29 | 28 | 16 |
|---|---|---|---|---|
| MXtd | MDir | Rsvd | Msk[28:16] | |
| R/WP-1 | R/WP-1 | R-1 | R/WP-0x1FFF | |

| 15 | 0 |
|---|---|
| Msk[15:0] | |
| R/WP-0xFFFF | |

LEGEND: R = Read only; WP = Write protected (protected by busy bit); -*n* = value after reset

**Table 5-31. IF1/IF2 Mask Registers (DCAN IF1MSK, DCAN IF2MSK) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | MXtd | | Mask Extended Identifier |
| | | 0 | The extended identifier bit (IDE) has no effect on the acceptance filtering. |
| | | 1 | The extended identifier bit (IDE) is used for acceptance filtering. |
| | | | When 11-bit (“standard”) identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered. |
| 30 | MDir | | Mask Message Direction |
| | | 0 | The message direction bit (Dir) has no effect on the acceptance filtering. |
| | | 1 | The message direction bit (Dir) is used for acceptance filtering. |
| 29 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 28-0 | Msk[28:0] | | Identifier Mask |
| | | 0 | The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). |
| | | 1 | The corresponding bit in the identifier of the message object is used for acceptance filtering. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.20 IF1/IF2 Arbitration Registers (DCAN IF1ARB, DCAN IF2ARB)

The bits of the IF1/IF2 arbitration registers mirror the arbitration bits of a message object. The function of the relevant message objects bits is described in *Structure of Message Objects*.

The IF1/IF2 arbitration registers (DCAN IF1ARB, DCAN IF2ARB) are shown in Figure 5-35 and Figure 5-36 and described in Table 5-32.

**NOTE:** While Busy bit of IF1/IF2 command register is one, IF1/IF2 register set is write protected.

**Figure 5-35. IF1 Arbitration Register (DCAN IF1ARB)**

| 31 | 30 | 29 | 28 ... 16 |
|---|---|---|---|
| MsgVal | Xtd | Dir | ID[28:16] |
| R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 |

| 15 ... 0 |
|---|
| ID[15:0] |
| R/WP-0 |

LEGEND: R = Read only; WP = Write protected (protected by busy bit); -*n* = value after reset

**Figure 5-36. IF2 Arbitration Register (DCAN IF2ARB)**

| 31 | 30 | 29 | 28 ... 16 |
|---|---|---|---|
| MsgVal | Xtd | Dir | ID[28:16] |
| R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 |

| 15 ... 0 |
|---|
| ID[15:0] |
| R/WP-0 |

LEGEND: R = Read only; WP = Write protected (protected by busy bit); -*n* = value after reset

**Table 5-32. IF1/IF2 Arbitration Registers (DCAN IF1ARB, DCAN IF2ARB) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | MsgVal | | Message valid |
| | | 0 | The message object is ignored by the message handler. |
| | | 1 | The message object is to be used by the message handler. |
| | | | The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN control register. This bit must also be reset before the identifier ID[28:0], the control bits Xtd, Dir or DLC[3:0] are modified, or if the messages object is no longer required. |
| 30 | Xtd | | Extended identifier |
| | | 0 | The 11-bit (“standard”) Identifier is used for this message object. |
| | | 1 | The 29-bit (“extended”) Identifier is used for this message object. |
| 29 | Dir | | Message direction |
| | | 0 | Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. |
| | | 1 | Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1). |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-32. IF1/IF2 Arbitration Registers (DCAN IF1ARB, DCAN IF2ARB) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 28-0 | ID[28:0] | | Message identifier |
| | | ID[28:0] | 29-bit identifier (extended frame) |
| | | ID[28:18] | 11-bit identifier (standard frame) |

The Arbitration bits ID[28:0], Xtd, and Dir are used to define the identifier and type of outgoing messages and (together with the mask bits Msk[28:0], MXtd, and MDir) for acceptance filtering of incoming messages.

A received message is stored into the valid message object with matching identifier and Direction = receive (data frame) or Direction = transmit (remote frame).

Extended frames can be stored only in message objects with Xtd = one, standard frames in message objects with Xtd = zero.

If a received message (data frame or remote frame) matches more than one valid message objects, it is stored into the one with the lowest message number.


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.21 IF1/IF2 Message Control Registers (DCAN IF1MCTL, DCAN IF2MCTL)

The bits of the IF1/IF2 message control registers mirror the message control bits of a message object. The function of the relevant message objects bits is described in *Structure of Message Objects*.

The IF1/IF2 message control registers (DCAN IF1MCTL, DCAN IF2MCTL) are shown in Figure 5-37 and Figure 5-38 and described in Table 5-33.

**NOTE:** While Busy bit of IF1/IF2 command register is one, IF1/IF2 register set is write protected.

**Figure 5-37. IF1 Message Control Register (DCAN IF1MCTL)**

| 31 | | | | | | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R/WP- 0 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| New Dat | Msg Lst | Int Pnd | UMask | TxIE | RxIE | Rmt En | Tx Rqst | EoB | Reserved | | | DLC[3:0] | | | |
| R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | | | R/WP-0 | | | |

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

**Figure 5-38. IF2 Message Control Register (DCAN IF2MCTL)**

| 31 | | | | | | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R/WP- 0 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| New Dat | Msg Lst | Int Pnd | UMask | TxIE | RxIE | Rmt En | Tx Rqst | EoB | Reserved | | | DLC[3:0] | | | |
| R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | R/WP-0 | | | R/WP-0 | | | |

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

**Table 5-33. IF1/IF2 Message Control Registers (DCAN IF1MCTL, DCAN IF2MCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 15 | NewDat | | New data |
| | | 0 | No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. |
| | | 1 | The message handler or the CPU has written new data into the data portion of this message object. |
| 14 | MsgLst | | Message lost (only valid for message objects with direction = receive) |
| | | 0 | No message lost since the last time when this bit was reset by the CPU. |
| | | 1 | The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten. |
| 13 | IntPnd | | Interrupt pending |
| | | 0 | This message object is not the source of an interrupt. |
| | | 1 | This message object is the source of an interrupt. The Interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority. |
| 12 | UMask | | Use acceptance mask |
| | | 0 | Mask ignored |
| | | 1 | Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering |
| | | | If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-33. IF1/IF2 Message Control Registers (DCAN IF1MCTL, DCAN IF2MCTL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 11 | TxIE | | Transmit interrupt enable |
| | | 0 | IntPnd will not be triggered after the successful transmission of a frame. |
| | | 1 | IntPnd will be triggered after the successful transmission of a frame. |
| 10 | RxIE | | Receive interrupt enable |
| | | 0 | IntPnd will not be triggered after the successful reception of a frame. |
| | | 1 | IntPnd will be triggered after the successful reception of a frame. |
| 9 | RmtEn | | Remote enable |
| | | 0 | At the reception of a remote frame, TxRqst is not changed. |
| | | 1 | At the reception of a remote frame, TxRqst is set. |
| 8 | TxRqst | | Transmit request |
| | | 0 | This message object is not waiting for a transmission. |
| | | 1 | The transmission of this message object is requested and is not yet done. |
| 7 | EoB | | End of Block |
| | | 0 | The message object is part of a FIFO Buffer block and is not the last message object of the FIFO Buffer block. |
| | | 1 | The message object is a single message object or the last message object in a FIFO Buffer Block. |
| | | | Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one. |
| 6-4 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 3-0 | DLC[3:0] | | Data length code |
| | | 0-8 | Data frame has 0-8 data bits. |
| | | 9-15 | Data frame has 8 data bytes. |
| | | | Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.22 IF1/IF2 Data A and Data B Registers (DCAN IF1DATA/DATB, DCAN IF2DATA/DATB)

The data bytes of CAN messages are stored in the IF1/IF2 registers in the following order:

1. In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received.
2. In CAN's serial bit stream, the MSB of each byte will be transmitted first.

The IF1/IF2 data A and data B registers (DCAN IF1DATA/DATB, DCAN IF2DATA/DATB) registers are shown in Figure 5-39, Figure 5-40, Figure 5-41 and Figure 5-42.

**Figure 5-39. IF1 Data A Register (DCAN IF1DATA)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| Data 3 | | Data 2 | |
| R/WP-0 | | R/WP-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Data 1 | | Data 0 | |
| R/WP-0 | | R/WP-0 | |

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

**Figure 5-40. IF1 Data B Register (DCAN IF1DATA)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| Data 7 | | Data 6 | |
| R/WP-0 | | R/WP-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Data 5 | | Data 4 | |
| R/WP-0 | | R/WP-0 | |

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

**Figure 5-41. IF2 Data A Register (DCAN IF2DATA)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| Data 7 | | Data 6 | |
| R/WP-0 | | R/WP-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Data 5 | | Data 4 | |
| R/WP-0 | | R/WP-0 | |

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

**Figure 5-42. IF2 Data B Register (DCAN IF2DATA)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| Data 7 | | Data 6 | |
| R/WP-0 | | R/WP-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Data 5 | | Data 4 | |
| R/WP-0 | | R/WP-0 | |

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.23 IF3 Observation Register (DCAN IF3OBS)

The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by CPU (Additional information can be found in *Structure of Message Objects*).

The observation flags (Bits [4:0]) in the IF3 observation register are used to determine, which data sections of the IF3 interface register set have to be read in order to complete a DMA read cycle. After all marked data sections are read, the DCAN is enabled to update the IF3 interface register set with new data.

Any access order of single bytes or half-words is supported. When using byte or half-word accesses, a data section is marked as completed, if all bytes are read.

**NOTE:** If IF3 Update Enable is used and no Observation flag is set, the corresponding message objects will be copied to IF3 without activating the DMA request line and without waiting for DMA read accesses.

A write access to this register aborts a pending DMA cycle by resetting the DMA line and enables updating of IF3 interface register set with new data. To avoid data inconsistency, the DMA controller should be disabled before reconfiguring IF3 observation register.

The status of the current read-cycle can be observed via status flags (Bits [12:8]).

If an interrupt line is available for IF3, an interrupt will be generated by IF3Upd flag. See the device-specific data sheet for the availability of this interrupt source.

With this interrupt, the observation status bits and the IF3Upd bit could be used by the application to realize the notification about new IF3 content in polling or interrupt mode.

The IF3 observation register (DCAN IF3OBS) is shown in Figure 5-43 and described in Table 5-34.

**Figure 5-43. IF3 Observation Register (DCAN IF3OBS)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IF3 Upd | Reserved | | IF3 SDB | IF3 SDA | IF3 SC | IF3 SA | IF3 SM | Reserved | | | DataB | DataA | Ctrl | Arb | Mask |
| R-0 | R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 5-34. IF3 Observation Register (DCAN IF3OBS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 15 | IF3 Upd | | IF3 Update Data |
| | | 0 | No new data has been loaded since last IF3 read. |
| | | 1 | New data has been loaded since last IF3 read. |
| 14-13 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 12 | IF3 SDB | | IF3 Status of Data B read access |
| | | 0 | All Data B bytes are already read out, or are not marked to be read. |
| | | 1 | Data B section has still data to be read out. |
| 11 | IF3 SDA | | IF3 Status of Data A read access |
| | | 0 | All Data A bytes are already read out, or are not marked to be read. |
| | | 1 | Data A section has still data to be read out. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-34. IF3 Observation Register (DCAN IF3OBS) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 10 | IF3 SC | | IF3 Status of control bits read access |
| | | 0 | All control section bytes are already read out, or are not marked to be read. |
| | | 1 | Control section has still data to be read out. |
| 9 | IF3 SA | | IF3 Status of Arbitration data read access |
| | | 0 | All Arbitration data bytes are already read out, or are not marked to be read. |
| | | 1 | Arbitration section has still data to be read out. |
| 8 | IF3 SM | | IF3 Status of Mask data read access |
| | | 0 | All mask data bytes are already read out, or are not marked to be read. |
| | | 1 | Mask section has still data to be read out. |
| 7-5 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 4 | DataB | | Data B read observation |
| | | 0 | Data B section has not to be read. |
| | | 1 | Data B section has to be read to enable next IF3 update. |
| 3 | DataA | | Data A read observation |
| | | 0 | Data A section has not to be read. |
| | | 1 | Data A section has to be read to enable next IF3 update. |
| 2 | Ctrl | | Ctrl read observation |
| | | 0 | Ctrl section has not to be read. |
| | | 1 | Ctrl section has to be read to enable next IF3 update. |
| 1 | Arb | | Arbitration data read observation |
| | | 0 | Arbitration data has not to be read. |
| | | 1 | Arbitration data has to be read to enable next IF3 update. |
| 0 | Mask | | Mask data read observation |
| | | 0 | Mask data has not to be read. |
| | | 1 | Mask data has to be read to enable next IF3 update. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.24 IF3 Mask Register (DCAN IF3MSK)

The IF3 mask register (DCAN IF3MSK) is shown in Figure 5-44 and described in Table 5-35.

**Figure 5-44. IF3 Mask Register (DCAN IF3MSK)**

| 31 | 30 | 29 | 28 | 16 |
|---|---|---|---|---|
| MXtd | MDir | Rsvd | Msk[28:16] | |
| R-1 | R-1 | R-1 | R/WP-0x1FFF | |

| 15 | 0 |
|---|---|
| Msk[15:0] | |
| R/WP-0xFFFF | |

LEGEND: R = Read only; -*n* = value after reset

**Table 5-35. IF3 Mask Register (DCAN IF3MSK) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | MXtd | | Mask Extended Identifier |
| | | 0 | The extended identifier bit (IDE) has no effect on the acceptance filtering. |
| | | 1 | The extended identifier bit (IDE) is used for acceptance filtering. |
| | | | When 11-bit (“standard”) identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered. |
| 30 | MDir | | Mask Message Direction |
| | | 0 | The message direction bit (Dir) has no effect on the acceptance filtering. |
| | | 1 | The message direction bit (Dir) is used for acceptance filtering. |
| 29 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 28-0 | Msk[28:0] | | Identifier Mask |
| | | 0 | The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). |
| | | 1 | The corresponding bit in the identifier of the message object is used for acceptance filtering. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.25 IF3 Arbitration Register (DCAN IF3ARB)

The IF3 arbitration register (DCAN IF3ARB) is shown in Figure 5-45 and described in Table 5-36.

**Figure 5-45. IF3 Arbitration Register (DCAN IF3ARB)**

| 31 | 30 | 29 | 28 | 16 |
|---|---|---|---|---|
| MsgVal | Xtd | Dir | ID[28:16] | |
| R-0 | R-0 | R-0 | R-0 | |

| 15 | 0 |
|---|---|
| ID[15:0] | |
| R-0 | |

LEGEND: R = Read only; WP = Write protected (protected by busy bit); -*n* = value after reset

**Table 5-36. IF3 Arbitration Register (DCAN IF3ARB) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | MsgVal | | Message Valid |
| | | 0 | The message object is ignored by the message handler. |
| | | 1 | The message object is to be used by the message handler. |
| | | | The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN control register. This bit must also be reset before the identifier ID[28:0], the control bits Xtd, Dir or DLC[3:0] are modified, or if the messages object is no longer required. |
| 30 | Xtd | | Extended Identifier |
| | | 0 | The 11-bit (“standard”) Identifier is used for this message object. |
| | | 1 | The 29-bit (“extended”) Identifier is used for this message object. |
| 29 | Dir | | Message Direction |
| | | 0 | Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. |
| | | 1 | Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1). |
| 28-0 | ID[28:0] | | Message Identifier |
| | | ID[28:0] | 29-bit Identifier (“extended frame”) |
| | | ID[28:18] | 11-bit Identifier (“standard frame”) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.26 IF3 Message Control Register (DCAN IF3MCTL)

The IF3 message control register (DCAN IF3MCTL) is shown in Figure 5-46 and described in Table 5-37.

**Figure 5-46. IF3 Message Control Register (DCAN IF3MCTL)**

| 31 | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | |
| R-0 | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 4 | 3 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| New Dat | Msg Lst | Int Pnd | UMask | TxIE | RxIE | Rmt En | Tx Rqst | EoB | Reserved | DLC[3:0] |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 5-37. IF3 Message Control Register (DCAN IF3MCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 15 | NewDat | | New Data |
| | | 0 | No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU. |
| | | 1 | The message handler or the CPU has written new data into the data portion of this message object. |
| 14 | MsgLst | | Message Lost (only valid for message objects with direction = receive) |
| | | 0 | No message lost since the last time when this bit was reset by the CPU. |
| | | 1 | The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten. |
| 13 | IntPnd | | Interrupt Pending |
| | | 0 | This message object is not the source of an interrupt. |
| | | 1 | This message object is the source of an interrupt. The Interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority. |
| 12 | UMask | | Use Acceptance Mask |
| | | 0 | Mask ignored |
| | | 1 | Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering |
| | | | If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one. |
| 11 | TxIE | | Transmit Interrupt enable |
| | | 0 | IntPnd will not be triggered after the successful transmission of a frame. |
| | | 1 | IntPnd will be triggered after the successful transmission of a frame. |
| 10 | RxIE | | Receive Interrupt enable |
| | | 0 | IntPnd will not be triggered after the successful reception of a frame. |
| | | 1 | IntPnd will be triggered after the successful reception of a frame. |
| 9 | RmtEn | | Remote enable |
| | | 0 | At the reception of a remote frame, TxRqst is not changed. |
| | | 1 | At the reception of a remote frame, TxRqst is set. |
| 8 | TxRqst | | Transmit Request |
| | | 0 | This message object is not waiting for a transmission. |
| | | 1 | The transmission of this message object is requested and is not yet done. |
| 7 | EoB | | End of Block |
| | | 0 | The message object is part of a FIFO Buffer block and is not the last message object of the FIFO Buffer block. |
| | | 1 | The message object is a single message object or the last message object in a FIFO Buffer Block. |
| | | | Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one. |
| 6-4 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-37. IF3 Message Control Register (DCAN IF3MCTL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 3-0 | DLC[3:0] | | Data Length Code |
| | | 0-8 | Data frame has 0-8 data bits. |
| | | 9-15 | Data frame has 8 data bytes. |
| | | | Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.27 IF3 Data A and Data B Registers (DCAN IF3DATA/DATB)

The data bytes of CAN messages are stored in the IF3 registers in the following order.

In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.

The IF3 data A and data B registers (DCAN IF3DATA/DATB) are shown in Figure 5-47 and Figure 5-48.

**Figure 5-47. IF3 Data A Register (DCAN IF3DATA)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| Data 3 | | Data 2 | |
| R-0 | | R-0 | |
| 15 | 8 | 7 | 0 |
| Data 1 | | Data 0 | |
| R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Figure 5-48. IF3 Data A Register (DCAN IF3DATB)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| Data 7 | | Data 6 | |
| R-0 | | R-0 | |
| 15 | 8 | 7 | 0 |
| Data 5 | | Data 4 | |
| R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.28 Update Enable Registers (DCAN IF3UPD12 to IF3UPD78)

The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UpdEn flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set.

The update enable registers (DCAN IF3UPD12 to IF3UPD78) are shown in Table 5-38 and described in Table 5-39.

**NOTE:** IF3 Update enable should not be set for transmit objects.

**Table 5-38. Update Enable Registers (DCAN IF3UPD12 to IF3UPD78))**

| Offset/Register | 31/15 | 30/14 | 29/13 | 29/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x160 CAN IF3UPD12 | IF3UpdEn[32:17] | | | | | | | | | | | | | | | |
| | R/W-0 | | | | | | | | | | | | | | | |
| | IF3UpdEn[16:1] | | | | | | | | | | | | | | | |
| | R/W-0 | | | | | | | | | | | | | | | |
| 0x164 CAN IF3UPD34 | IF3UpdEn[64:49] | | | | | | | | | | | | | | | |
| | R/W-0 | | | | | | | | | | | | | | | |
| | IF3UpdEn[48:33] | | | | | | | | | | | | | | | |
| | R/W-0 | | | | | | | | | | | | | | | |
| 0x168 CAN IF3UPD56 | IF3UpdEn[96:81] | | | | | | | | | | | | | | | |
| | R/W-0 | | | | | | | | | | | | | | | |
| | IF3UpdEn[80:65] | | | | | | | | | | | | | | | |
| | R/W-0 | | | | | | | | | | | | | | | |
| 0x16C CAN IF3UPD78 | IF3UpdEn[128:113] | | | | | | | | | | | | | | | |
| | R/W-0 | | | | | | | | | | | | | | | |
| | IF3UpdEn[112:97] | | | | | | | | | | | | | | | |
| | R/W-0 | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 5-39. Update Enable Registers (DCAN IF3UPD12 to IF3UPD78) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | IF3UpdEn[128:1] | | IF3 Update Enabled (for all message objects) |
| | | 0 | Automatic IF3 update is disabled for this message object. |
| | | 1 | Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.29 CAN TX I/O Control Register (DCAN TIOC)

The CAN_TX pin of the DCAN module can be used as general purpose IO pin if CAN function is not needed. The CAN TX I/O control register (DCAN TIOC) is shown in Figure 5-49 and described in Table 5-40.

**NOTE:** The values of the IO control registers are only writable if Init bit of the CAN control register is set.

**NOTE:** The OD, Func, Dir and Out bits of the CAN TX IO control register are forced to certain values when Init bit of CAN control register is reset (see bit descriptions).

**Figure 5-49. CAN TX I/O Control Register (DCAN TIOC)**

| 31 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|
| Reserved | | PU | PD | OD |
| R-0 | | R/W-D | R/W-D | R/WP-0 |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | Func | Dir | Out | In |
| R-0 | R/WP-0 | R/WP-0 | R/WP-0 | R-U |

LEGEND: R/W = Read/Write; R = Read only; WP = Protected Write (protected by Init bit), U = Undefined; D = Device dependent; -*n* = value after reset

**Table 5-40. CAN TX I/O Control Register (DCAN TIOC) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-19 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 18 | PU | | CAN_TX pull up/pull down select. This bit is only active when CAN_TX is configured to be an input. |
| | | 0 | CAN_TX pull down is selected, when pull logic is active (PD = 0). |
| | | 1 | CAN_TX pull up is selected, when pull logic is active(PD = 0). |
| 17 | PD | | CAN_TX pull disable. This bit is only active when CAN_TX is configured to be an input. |
| | | 0 | CAN_TX pull is active |
| | | 1 | CAN_TX pull is disabled |
| 16 | OD | | CAN_TX open drain enable. This bit is only active when CAN_TX is configured to be in GIO mode (TIOC.Func=0). |
| | | 0 | The CAN_TX pin is configured in push/pull mode. |
| | | 1 | The CAN_TX pin is configured in open drain mode. |
| | | | Forced to '0' if Init bit of CAN control register is reset. |
| 15-4 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 3 | Func | | CAN_TX function. This bit changes the function of the CAN_TX pin |
| | | 0 | CAN_TX pin is in GIO mode. |
| | | 1 | CAN_TX pin is in functional mode (as an output to transmit CAN data). |
| | | | Forced to '1' if Init bit of CAN control register is reset. |
| 2 | Dir | | CAN_TX data direction. This bit controls the direction of the CAN_TX pin when it is configured to be in GIO mode only (TIOC.Func=0) |
| | | 0 | The CAN_TX pin is an input. |
| | | 1 | The CAN_TX pin is an output |
| | | | Forced to '1' if Init bit of CAN control register is reset. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-40. CAN TX I/O Control Register (DCAN TIOC) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | Out | | CAN_TX data out write. This bit is only active when CAN_TX pin is configured to be in GIO mode (TIOC.Func = 0) and configured to be an output pin (TIOC.Dir = 1). The value of this bit indicates the value to be output to the CAN_TX pin. |
| | | 0 | The CAN_TX pin is driven to logic low |
| | | 1 | The CAN_TX pin is driven to logic high |
| | | | Forced to Tx output of the CAN core, if Init bit of CAN control register is reset. |
| 0 | In | | CAN_TX data in |
| | | 0 | The CAN_TX pin is at logic low |
| | | 1 | The CAN_TX pin is at logic high |
| | | | Note: When CAN_TX pin is connected to a CAN transceiver, an external pullup resistor has to be used to ensure that the CAN bus will not be disturbed (e.g. while reset of the DCAN module). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 5.14.1.30 CAN RX IO Control Register (DCAN RIOC)

The CAN_RX pin of the DCAN module can be used as general purpose IO pin if CAN function is not needed. The CAN RX I/O control register (DCAN RIOC) is shown in Figure 5-50 and described in Table 5-41.

**NOTE:** The values of the IO control registers are writable only if Init bit of CAN control register is set.

**NOTE:** The OD, Func and Dir bits of the CAN RX IO control register are forced to certain values when the Init bit of CAN control register is reset (see bit descriptions).

**Figure 5-50. CAN RX IO control register (DCAN RIOC)**

| 31 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|
| Reserved | | PU | PD | OD |
| R-0 | | R/W-D | R/W-D | R/WP-0 |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | Func | Dir | Out | In |
| R-0 | R/WP-0 | R/WP-0 | R/WP-0 | R-U |

LEGEND: R/W = Read/Write; R = Read only; WP = Protected Write (protected by Init bit), U = Undefined; D = Device dependent; -*n* = value after reset

**Table 5-41. CAN RX IO Control Register (DCAN RIOC) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-19 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 18 | PU | | CAN_RX pull up/pull down select. This bit is only active when CAN_RX is configured to be an input. |
| | | 0 | CAN_RX pull down is selected, when pull logic is active (PD = 0). |
| | | 1 | CAN_T=RX pull up is selected, when pull logic is active(PD = 0). |
| 17 | PD | | CAN_RX pull disable. This bit is only active when CAN_TX is configured to be an input. |
| | | 0 | CAN_RX pull is active |
| | | 1 | CAN_RX pull is disabled |
| 16 | OD | | CAN_RX open drain enable. This bit is only active when CAN_RX is configured to be in GIO mode (TIOC.Func=0). |
| | | 0 | The CAN_RX pin is configured in push/pull mode. |
| | | 1 | The CAN_RX pin is configured in open drain mode. |
| | | | Forced to '0' if Init bit of CAN control register is reset. |
| 15-4 | Reserved | 0 | These bits are always read as 0. Writes have no effect. |
| 3 | Func | | CAN_RX function. This bit changes the function of the CAN_RX pin |
| | | 0 | CAN_RX pin is in GIO mode. |
| | | 1 | CAN_RX pin is in functional mode (as an output to transmit CAN data). |
| | | | Forced to '1' if Init bit of CAN control register is reset. |
| 2 | Dir | | CAN_RX data direction. This bit controls the direction of the CAN_RX pin when it is configured to be in GIO mode only (TIOC.Func=0) |
| | | 0 | The CAN_RX pin is an input. |
| | | 1 | The CAN_RX pin is an output |
| | | | Forced to '1' if Init bit of CAN control register is reset. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 5-41. CAN RX IO Control Register (DCAN RIOC) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | Out | | CAN_RX data out write. This bit is only active when CAN_RX pin is configured to be in GIO mode (TIOC.Func = 0) and configured to be an output pin (TIOC.Dir = 1). The value of this bit indicates the value to be output to the CAN_RX pin. |
| | | 0 | The CAN_RX pin is driven to logic low |
| | | 1 | The CAN_RX pin is driven to logic high |
| | | | Forced to Tx output of the CAN core, if Init bit of CAN control register is reset. |
| 0 | In | | CAN_RX data in |
| | | 0 | The CAN_RX pin is at logic low |
| | | 1 | The CAN_RX pin is at logic high |
| | | | Note: When CAN_RX pin is connected to a CAN transceiver, an external pullup resistor has to be used to ensure that the CAN bus will not be disturbed (e.g. while reset of the DCAN module). |


Copyright © 2011–2016, Texas Instruments Incorporated

# DMM/TILER

This chapter describes the Dynamic Memory Manager (DMM) and its Tiling and Isometric Lightweight Engine for Rotation (TILER) submodule.

**Topic** **Page**

**6.1 Introduction** ..... **1153**
**6.2 Architecture** ..... **1156**
**6.3 Use Case** ..... **1191**
**6.4 DMM/TILER Registers** ..... **1206**


Copyright © 2011–2016, Texas Instruments Incorporated

## 6.1 Introduction

### 6.1.1 Overview

This section describes the Dynamic Memory Manager (DMM) and its Tiling and Isometric Lightweight Engine for Rotation (TILER) submodule.

As shown in Figure 6-1, the DMM is located in front to the SDRAM controllers and thus interfaces to the memory accesses generated by all the initiators.

The dynamic memory manager (DMM), is a module aimed at managing – in a broad sense – various aspects of memory accesses such as:

- Initiator-indexed priority generation
- Multizone SDRAM memory interleaving configuration
- Block object transfer optimization – tiling and sub-tiling
- Centralized low-latency page translation – MMU-like feature

The dynamic qualifier for memory management highlights the software configurability – and, hence, the run-time nature – of the four aspects of memory management handled by the DMM.

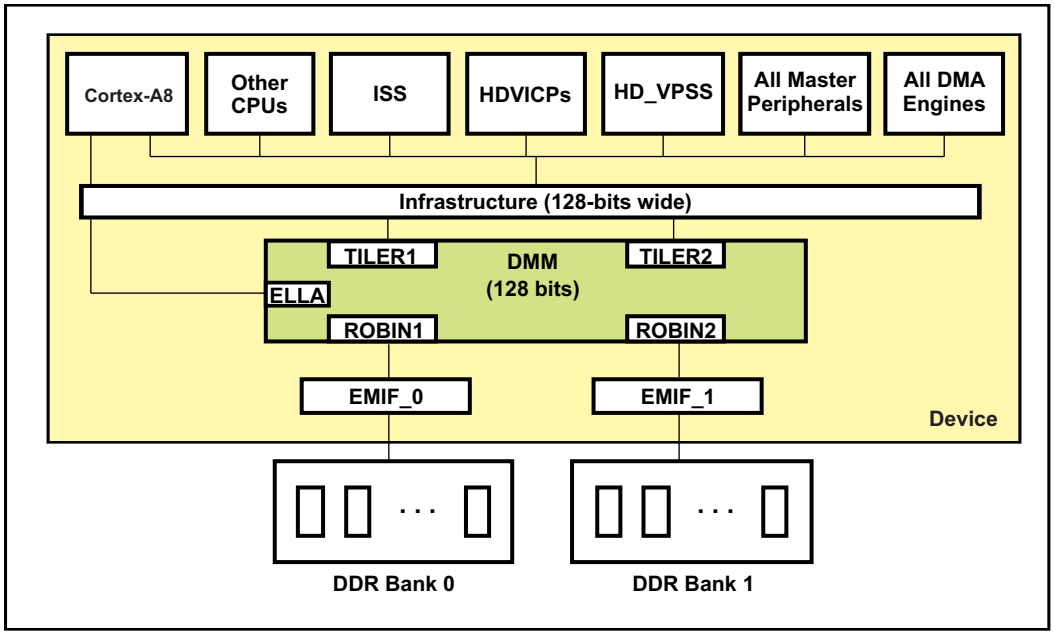
On a functional perspective, the role of the DMM is to:

- add initiator-based priority to any incoming requests
- perform to-and-fro tiling conversions of "tiled" requests
- provide an optional low-latency page-based translation to handle memory fragmentation - MMU
- distribute the traffic on both the attached memory controllers according to the interleaving configuration

The TILER is a submodule within the DMM aimed at efficient handling of two dimensional data, such as video/graphics accesses for HDVICP2, by the use of tiled format.

- optionally managing the memory fragmentation and zero-copy physical frame buffers swapping through a page-grained translation
- making on-the-fly, zero overhead transforms, such as 90°, 180°, or 270° rotations, with either a horizontal or vertical reflection

**Figure 6-1. DMM Integration**





Copyright © 2011–2016, Texas Instruments Incorporated

### 6.1.2 Features

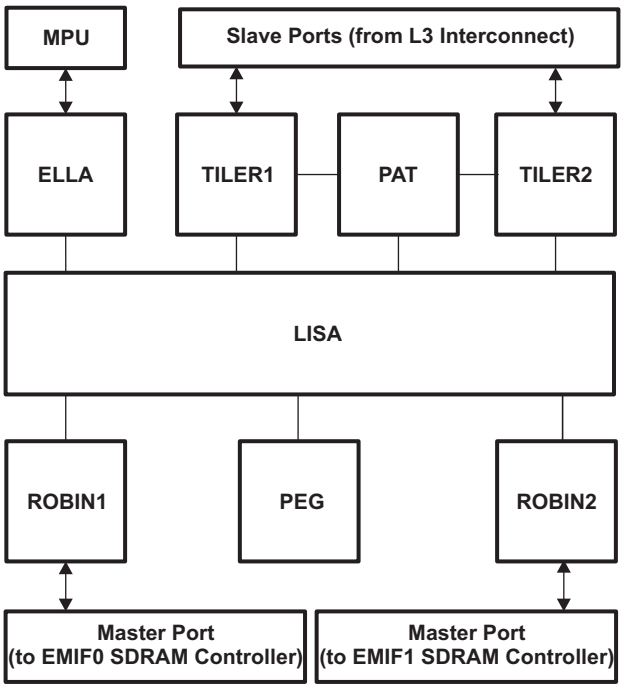
- Special low latency interconnect port. ELLA, only for Cortex-A8 accesses.
- Ability to interleave the DDR data between two EMIF banks, using the programmable multi-zone DRAM memory mapping . This increases the memory throughput by a factor of 2. Up-to four unique memory sections supported.
- Programmable Initiator based request priority extension, for up to 16 Initiator groups.
- Support for address translations of tiled data, on a 4KB page granularity using the PAT. This helps manage memory fragmentation.
- One internal address lookup table (LUT) with 256x128 entries. Four refill engines to program the LUT, with automatic synchronized reload.
- Supports up to 4 unique PAT views.

### 6.1.3 Functional Block Diagram

Figure 6-2 shows the DMM macro architecture. The DMM consists of six blocks:

- A Priority Extension Generator (PEG) to generate priorities required by the SDRAM controller, note that these priorities are not used in the DMM
- One Extra Low Latency Access (ELLA), having its own interconnect slave port, for providing lower latency accesses to the memory
- A Local Interconnect and Synchronization Agent (LISA) to synchronize all DMM subsystems and provide access to their configuration registers
- A Physical Address Translator (PAT) for managing the memory fragmentation
- Two Re-Ordering Buffer and Initiator Nodes (ROBIN), having each their own interconnect master port, to initiate requests to the SDRAM controller and allow tiled data, tiled response and split response reconstruction. The ROBIN block is only managing the re-ordering buffer and performing the data re-ordering due to the orientation.
- Two Tiling and Isometric Lightweight Engine for Rotation (TILER), having each their own interconnect slave port, for converting requests to-and-fro between the input virtual addressing mode and the output physical tiled addressing mode. Note that, the tiling conversions of requests, write data and responses is fully performed in the TILER blocks.

**Figure 6-2. DMM Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

### 6.1.4 Terminologies and Acronyms Used in this Document

The following is a brief explanation of some terms used in this document:

**bpp—** Bits per pixel

**DMM—** Dynamic Memory Manager

**ELLA—** Extra Low Latency Access

**GB, GiB—** Both imply Giga Byte

**Initiator—** A node inside the Device and is either a CPU, peripheral or DMA engine, which can be internal bus Master. Each initiator is identified by a ConnID (Connection ID). With a limit of only 16 ConnIDs, some of the Initiators are grouped together with same ConnID.

**Interlaced—** Qualifier for accesses skipping one line every line

**IVA—** Image Video Accelerator. Also called HDVICP2, IVA_HD.

**LISA—** Local Interconnect and Synchronisation Agent

**KB, KiB—** Both imply Kilo Byte

**LUT—** Look Up Table

**MMU—** Memory Management Unit

**MPU—** Main Processing Unit. For this device, it is Cortex-A8

**PAT—** Physical Address Translator

**PEG—** Priority Extension Generator

**Progressive—** Qualifier for line-by-line accesses

**ROBIN—** Re-Ordering Buffer and Initiator Node

**Tiled access—** 1D or 2D Access to the tiled area, where the image data is read and written in two dimensional format, to improve the efficiency of accesses of 2D accesses, example image macro block. TILER Tiling and Isometric Lightweight Engine for Rotation 1D access A simple linear read or write access request. The DMM responds with read/write from/to the contiguous memory starting with address specified in the request.

**2D access—** HDVICP2 and HD_VPSS, can generate a special access to 2D image buffers, with read/write request, with Height and Width information. The DMM-TILER decodes the access type based on the Height, width and address, and responds with read/write from/to data in the physical memory, which has been co-located, with sub-tile granularity.


Copyright © 2011–2016, Texas Instruments Incorporated

## 6.2 Architecture

### 6.2.1 DMM Functional Description

#### 6.2.1.1 Priority Extension Generator (PEG)

The priority extension generator (PEG) is a dynamic software-programmable initiator-indexed table of priorities. Its unique role is to generates priorities for each access forwarded by the DMM to the SDRAM controller. For initiators that do not generate their own per-transaction priority, the priority value that is programmed in the table, will be assigned to all SDRAM accesses generated by that initiator. These priorities are not used by the DMM, for internal arbitrations.

The 64 priority entries are software-programmable with DMM_PEG_PRIO0 (for the first eight entries of the ConnID table) through DMM_PEG_PRIO7 (for the last eight entries) registers and set on a 3-bit scale, ranging from values 0-7, with Priority=0, being the highest priority.

When a request is passed to the SDRAM controller, the priority from the corresponding DMM_PEG_PRIOx field for an initiator is passed to SDRAM. For example, an access from the C674x DSP is sent to SDRAM with the priority captured in DMM_PEG_PRIOn. Note that the priority does not impact the behavior in DMM itself. It is passed to the SDRAM controller and the controller may use it in the prioritization of requests. However, for HD_VPSS, the peripheral itself generates a priority. The DMM_PEG_PRIOx field for HD_VPSS is bypassed and the priority indicated by HD_VPSS access is sent to SDRAM controller.

#### 6.2.1.2 ELLA

The Extra Low Latency Access (ELLA) of the DMM is a simple interface port for all accesses made by Cortex-A8. As the name suggests, this interface is simplified to reduce latency to the request from Cortex-A8. In this respect, the ELLA block:

- Is limited to 1D bursts only
- Is not capable of performing tiling conversions
- Does not interact with the PAT block

The ELLA block main role is to split incoming requests at DMM atomic unit boundaries to ensure that requests sent to the SDRAM controller fit in a single SDRAM page. More precisely, the ELLA block is responsible for:

- Allocating an internal response context to timely generate the appropriate responses
- Splitting the incoming requests at DMM atomic section boundaries
- Requesting internal buffer allocation in the appropriate ROBIN
- In case of a write request, allocating and updating an internal write context to subsequently direct the incoming write data into the relevant re-ordering buffer

**NOTE:** In the device, Cortex-A8 accesses in the system address space of 8000 0000h-FFFF FFFFh, are only routed through ELLA port. Accesses made by Cortex-A8, in the all four tiled modes, including Paged mode, will be routed through the TILER ports, thus not getting benefit of lower latency of the ELLA port.

ELLA sub-module is not software configurable.

#### 6.2.1.3 LISA

The Local Interconnect and Synchronisation Agent (LISA) is a hardware interconnect module, aimed at setting priorities, managing tags and memory mapping. LISA maps the system addresses on the incoming DMM requests to the SDRAM addresses, as per the LISA sections programming.

LISA interconnect routes:

1. ELLA and TILER requests on the ROBIN initiator nodes.
2. ELLA and TILER write data on the ROBIN write buffers.


Copyright © 2011–2016, Texas Instruments Incorporated

3. ROBIN read data to the relevant TILER initiators or to ELLA initiator.

The LISA block registers are DMM_LISA_MAP_i (i = 0 to 3) and DMM_LISA_LOCK.

LISA block manages all the incoming requests to ELLA, TILER and PAT and translates then to requests to Registers and ROBIN ports. Highest priority is given to accesses coming to the low-latency port (ELLA). When both DMM-ROBIN ports are in use, LISA module interleaves the two ports at a programmable boundary from 128 Bytes or more, as programmed in the DMM_LISA_MAPn registers. See sub-section on Section Mapping for details about LISA sections programming.

#### *6.2.1.3.1 LISA LOCK*

The DMM_LISA_LOCK register is used to lock the configuration once set. If written to one, the LOCK bit prevents further writes to all DMM_LISA_MAP_i registers. The LOCK itself cannot be written back to 0. A reset is required to re-program the sections.

It is expected that LISA MAP is set up once during system configuration and not changed subsequently. The LOCK feature helps accomplish this and is expected to be set once the configuration is done.

### 6.2.1.4 PAT, Physical Address Translator Engine

The physical address translation engine (PAT) of the DMM is composed of two 32-k entry physical address translation vector table and a set of four refill engines. The refill engine is a specialized DMA aimed at refilling the content of the physical address translation table.

The address translation mechanism is only available when the incoming request is hitting a page mode or tiled mode container, that is, when the incoming address is targeting the TILER or its aliased view in the system addressing space. Otherwise the physical address translation logic is bypassed so that the resulting physical address corresponds to the input address.

The PAT is supporting multiple address translation schemes, called views, which can be bound to one or more initiator through a view mapping mechanism.

The usage of the Refill Engines is described in detail in the later section.

#### *6.2.1.4.1 PAT Views*

A PAT view defines the kind of physical address translation to perform for each tiled mode accesses (page, 8-bit, 16-bit and 32-bit). Each mode of each PAT view can be configured to either use PAT Direct Access Translation or PAT In-Direct Access Translation.

The PAT supports up to four unique Views, using the DMM_PAT_VIEW_MAP[0..3] Registers.

#### *6.2.1.4.2 PAT View Mappings*

The 16 groups of initiators of the Device, each identified by their ConnID, share a set of 4 PAT views. The connection from an initiator to a PAT view is made through the DMM_PAT_VIEW register. The register DMM_PAT_VIEW0 is used for the first eight initiator groups and DMM_PAT_VIEW1 for the second eight. Thus each initiator can choose one of the four configurable PAT views in the system.

#### *6.2.1.4.3 PAT View Map Base Address*

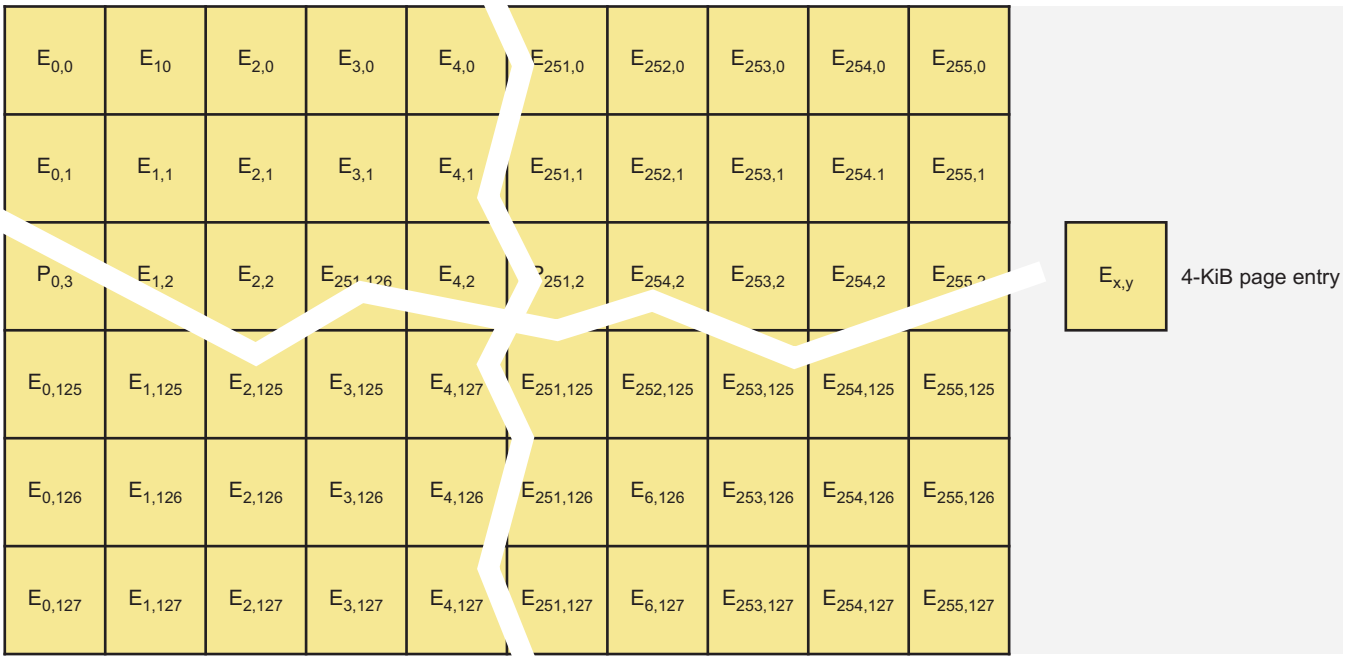
The PAT view map base address is defining the base address of all PAT translated addresses. The bit 31 of all PAT translated address is set to BASE_ADDR. In the Device, the bit 31 should be set to 1, which corresponds top 2GB assigned for external SDRAM in the system memory map. Hence the addresses translated by PAT range in addresses 8000 0000h-FFFF FFFFh.

#### *6.2.1.4.4 PAT - Look Up Tables (LUTs)*

The PAT contains one LUT, having 32K (256 × 128) entries. This geometry corresponds to that of the Virtual TILER container. So incoming address maps the actual entry in the LUT. The PAT shall then translate to a physical memory mapped to any 4K page the DDR. Each entry of the PAT address corresponds to the page in the DMM container that has the same location. For example - The entry (74, 42) in the table corresponds to the page (74, 42) in any DMM container.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 6-3. DMM Look-Up Table**




Each of this entry points to a 4K page in the memory. Hence a total of 128 MB of tiled memory can be mapped using one LUT. With upto 2G of DDR space, supported in the Device, each LUT entry is 19 bits wide.

The PAT uses the LUTs only in case of PAT In-Direct Access Translation.

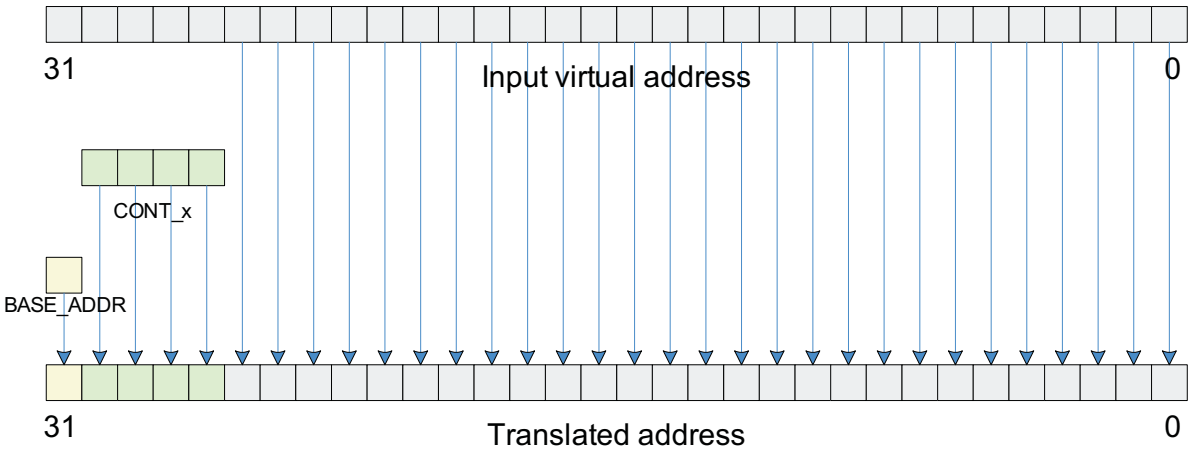
#### *6.2.1.4.5 PAT Direct Access Translation*

The container-grained translation is named the direct access. In this mode, the translation vector is directly given by the CONT_x field, in DMM_PAT_VIEW_MAP register, corresponding to the accessed mode(8-bit/16-bit/32-bit/paged mode).

With PAT Direct Access translation, the 128 MB virtual containers for all the four modes, 8-bit, 16-bit, 32-bit, and paged modes can be mapped to their unique System Address, by defining their base addresses.

Figure 6-4 describes the actual translation that happens.

**Figure 6-4. DMM PAT Direct Access Translation**





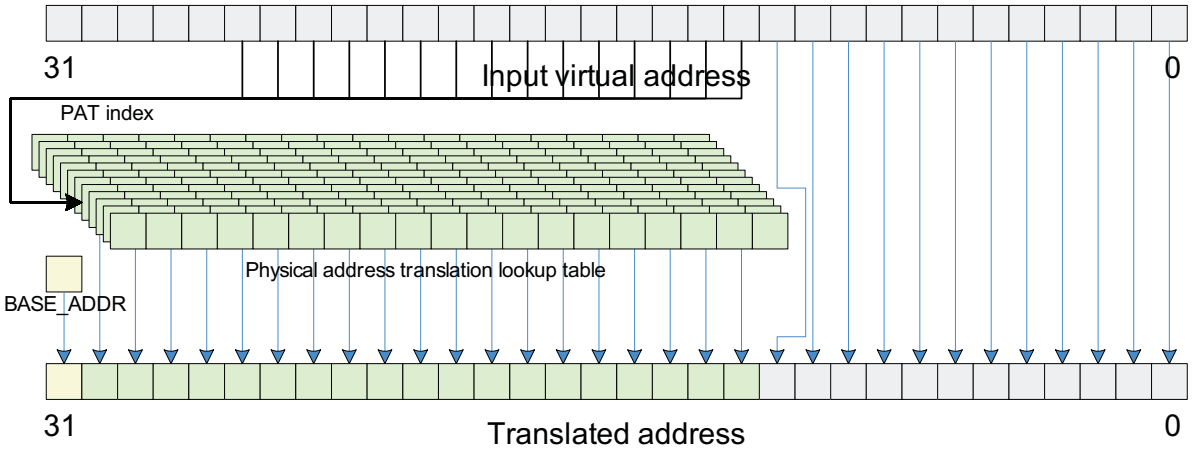
Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.1.4.6 PAT In-Direct Access Translation - Paged Mode Access

DMM_PAT_VIEW[0..3] specify, whether to do Direct or In-direct access translation. In case of In-direct access translation, the single LUT in the system is used for address lookup. Then bits [19:12] of input address specify the X co-ordinate and bits [26:20] of input address specify the Y co-ordinate LUT entry. The PAT replaces bits[30:12] of input address, with 19-bit value specified in LUT entry.

This translation happens on the 4K page boundary and hence the lower 12 bits are not translated. Figure 6-5 describes the actual translation that happens.

**Figure 6-5. DMM PAT In-Direct Access Translation**




#### 6.2.1.4.7 PAT In-Direct Access Translation - 8-bit, 16-bit, 32-bit Mode Access

DMM_PAT_VIEW[0..3] specify, whether to do Direct or In-direct access translation. Per each of the 8-bit,16-bit ,32-bit and paged container mode, user specifies which LUT to use for address lookup, by using CONT_x field, in DMM_PAT_VIEW_MAP registers.

How the 26 LSBs of this incoming virtual address is interpreted by the PAT is different for 8-bit,16-bit ,32-bit mode accesses, as compared to the paged mode access described in the preceding section. Please refer the TILER section describes the rationale behind decoding the address this way.

Virtual address decoding in 8-bit mode access mode is:

- Bits 0:5 - 6 bits offset into the horizontal line of the page
- Bits 6:13 - 8 bits, that select horizontal page in the tiler container as well as the X co-ordinate of the LUT table
- Bits 14:19 - 6 bits offset to select the line inside the page
- Bits 20:26 - 7 bits, that select vertical page in the tiler container and also Y co-ordinate of the LUT table
- Bits 27:31 - In 8 bit mode, their value is binary (01100), that is address in range 6000 0000h-67FF FFFFh

Virtual address decoding in 16-bit mode access mode is:

- Bit 0 - is always 0, as accesses are at-least 16-bit aligned
- Bits 1:6 - 6 bits offset into the horizontal line of the page
- Bits 7:14 - 8 bits, that select horizontal page in the tiler container as well as the X co-ordinate of the LUT table
- Bits 15:19 - 5 bits offset to select the line inside the page
- Bits 20:26 - 7 bits, that select vertical page in the tiler container and also Y co-ordinate of the LUT table
- Bits 27:31 - In 16 bit mode, their value is binary (01101), that is address in range 6800 0000h-


Copyright © 2011–2016, Texas Instruments Incorporated

6FFF FFFFh

Virtual address decoding in 32-bit mode access mode is:

- Bit 0,1 - are always 0, as accesses are at-least 32-bit aligned
- Bits 2:6 - 5 bits offset into the horizontal line of the page
- Bits 7:14 - 8 bits, that select horizontal page in the tiler container as well as the X co-ordinate of the LUT table
- Bits 15:19 - 5 bits offset to select the line inside the page
- Bits 20:26 - 7 bits, that select vertical page in the tiler container and also Y co-ordinate of the LUT table
- Bits 27:31 - In 32 bit mode, their value is binary (01110), that is address in range 7000 0000h-77FF FFFFh

#### 6.2.1.5 ROBIN

The re-ordering buffer and initiator node (ROBIN) is a block aimed at providing some working buffering for converting data and responses to-and-fro between raster and tiled organizations and a master port to connect to the SDRAM controller.

The ROBIN block does:

- Request Forwarding
- Buffering of Write access data and buffering of Read access Response data
- Keeps write data ordering
- Intra-word tiling and orientation transforms
- Tag handling

**NOTE:** Both ROBIN sub-modules are not software configurable.

#### 6.2.1.6 Section Mapping

In the device, DMM supports two unique SDRAM controllers, with a software configurable option to interleave data between both banks, at granularity of 128 Bytes, 256 Bytes and 512 Bytes. When accessing tiled data in 8-bit, 16-bit and 32-bit modes, which in the interleaved section of the memory, the interleaving will happen at the tile boundary of 1KB, over-riding the interleaving definition of the section.

For optimal system performance, it is recommended to enable interleaving between the 2 EMIF banks and thus have same sized memory on both the EMIF banks. Example : 512MB of DDR3 on EMIF bank 0 and 512MB of DDR3 on EMIF bank 1, for a total system DDR3 memory of 1GB.

**NOTE:** This interleaving is supported only if DDRs system connected to both the SDRAM controllers have same electrical characteristics.

If for cost reduction, one chooses to build a system with asymmetrical memory on both EMIF banks, then a few limited configurations can be supported by using the LISA section programming feature of DMM.

The address mapping inside the DMM is configurable through up to four sections. Each section is defined based on:

- Its system address: the base address of the decoding range for the section. This is the address of the incoming access to the DMM, also referred to as System Address.
- Its size: the encoding is the number of bits actually used in the upper 8-bits of the incoming system address. Hence the size of the section can be a value 16MB-2GB and power of 2. With interleaving enabled, the minimum size of the section is 32 MB.
- Its physical address: the base address of the memory range access in the external memory controller. It is also referred to as SDRC address.
- The target memory controller: A section may hit either of EMIF banks or both.

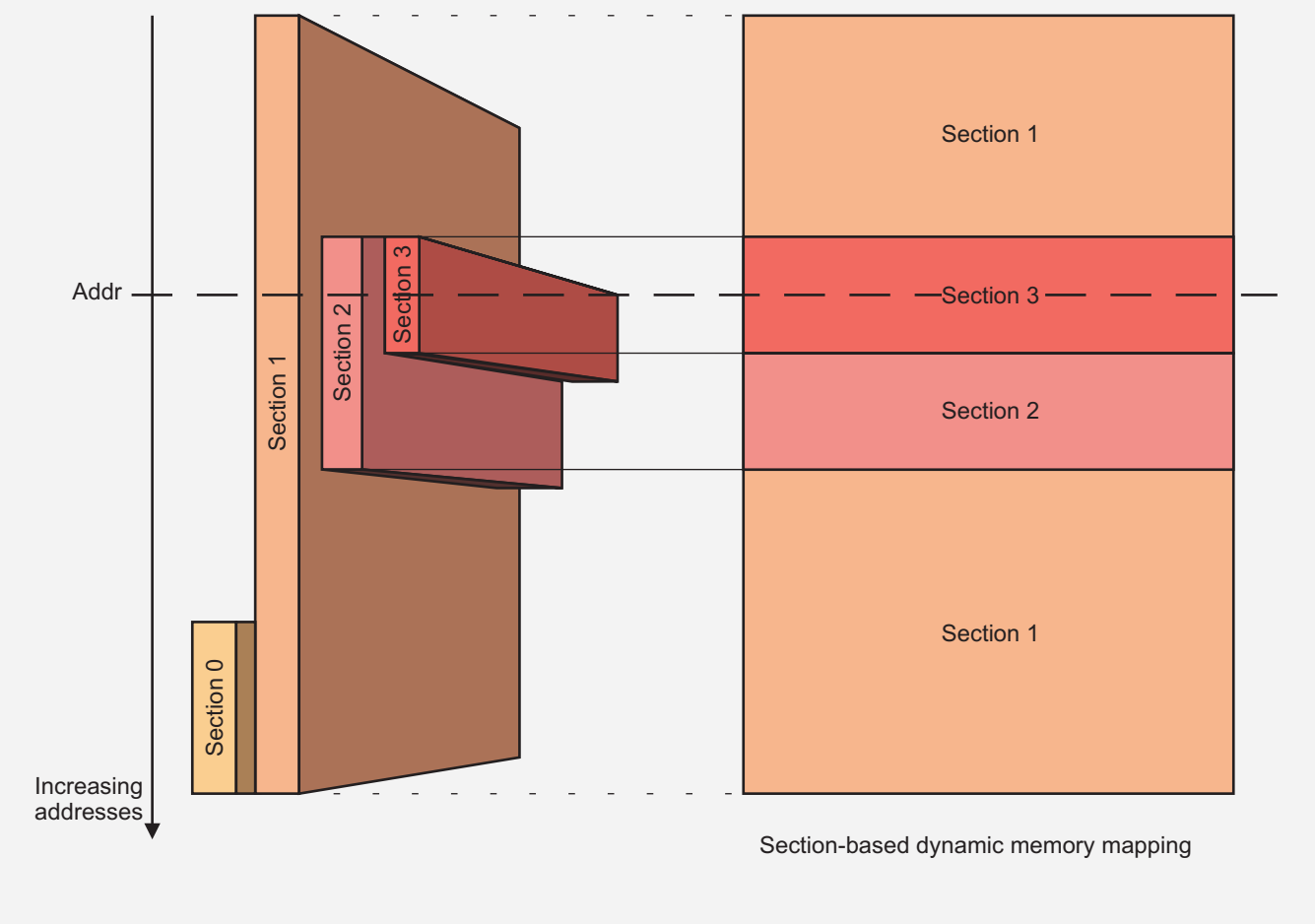

Copyright © 2011–2016, Texas Instruments Incorporated

- Its interleaving definition : Interleave at 128Bytes, 256 Bytes, 512 Bytes or no interleaving.

A LISA section is:

- a memory segment of size 16MB – 2GB, which is a power of two and aligned to that size in the system map. If the section is configured to interleave data between 2 EMIF, then the minimum section size is 32 MB.
- An area with constant interleaving scheme, with fixed interleaving granularity. Interleaving between 2 EMIF banks can be also disabled, for this section.
- If two sections overlap, then the property of higher section number will be applied, as shown in Figure 6-6. There can be a total of 4 unique section definition in DMM.

**Figure 6-6. DMM Section and Memory Mapping**




In Figure 6-6, a request at the system address Addr will follow the interleaving scheme of section 3, although hitting sections 1, 2 and 3. Similarly, this DMM configuration also prevents any request to use the interleaving scheme of section 0, since section 0 is fully masked by section 1 – which has a higher priority.

The DMM section specifies the range of incoming (to DMM) System address (granularity of 16 MB) and how it maps to the physical SDRC address of EMIF0 and/or EMIF 1, with similarly aligned granularity.

The interleaving of data between two EMIF banks is transparent to all the initiators of the system and will not require special address conversions. Figure 6-7 and Figure 6-8 give examples of all the interleaving schemes.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 6-7. 128B and 256B Interleaving**

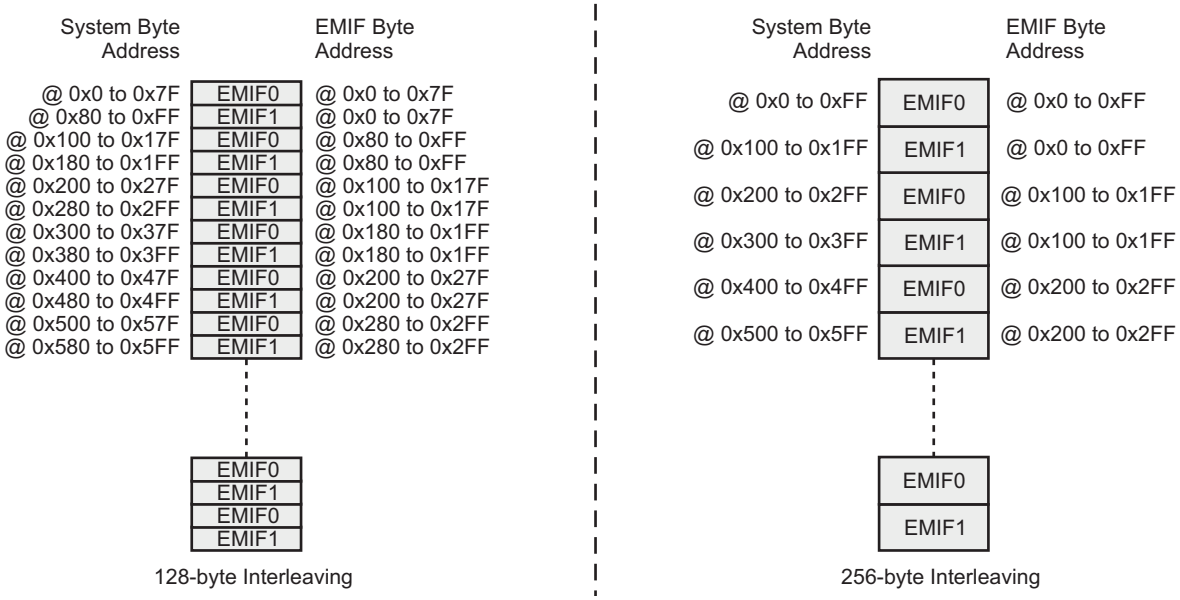
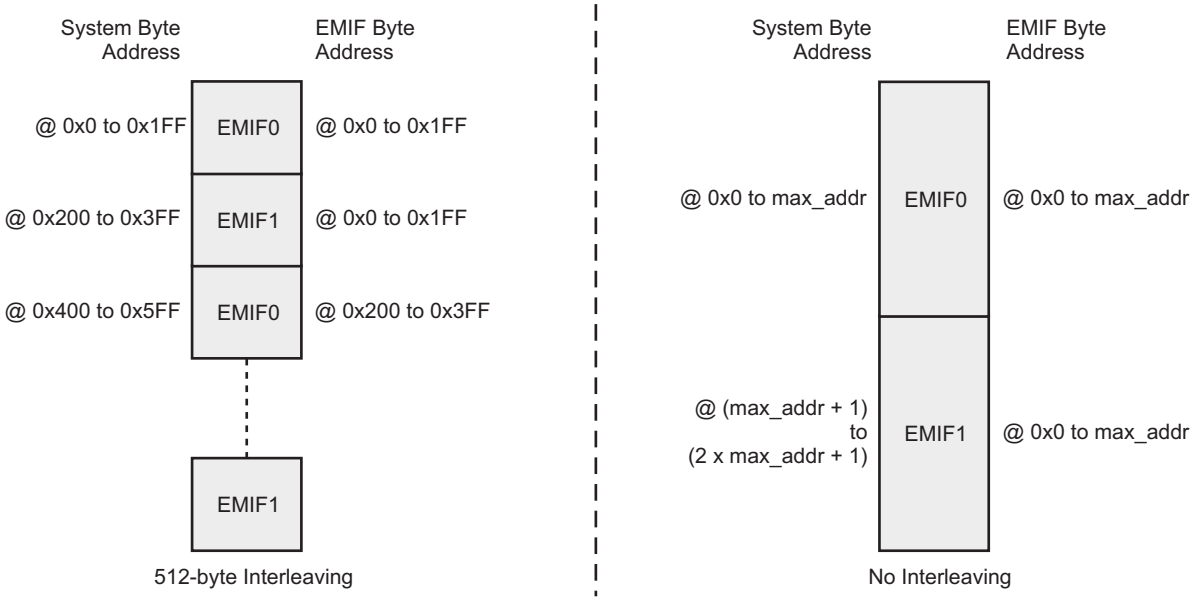



**Figure 6-8. 512KB and 1KB Interleaving**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.1.7 TILER

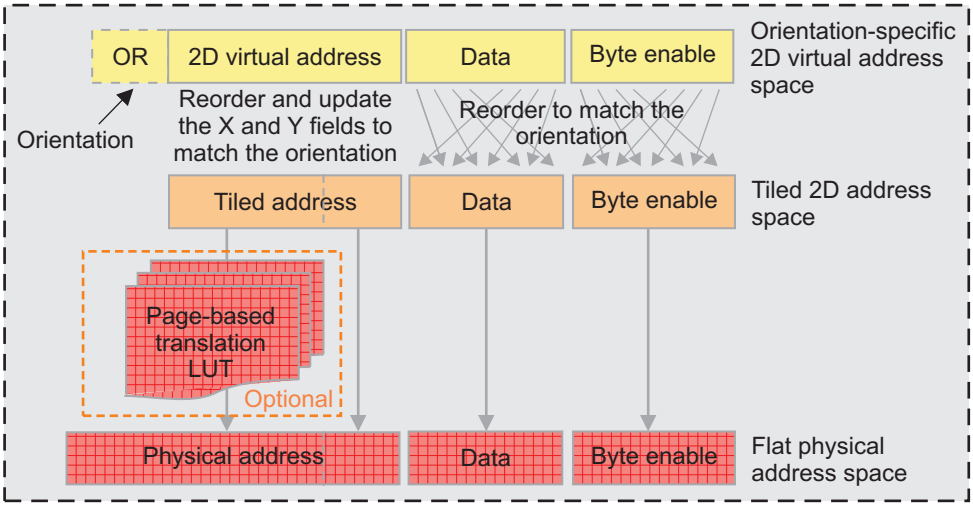
As can be seen in Figure 6-2, the two TILER ports interface to the incoming accesses to DMM. Its primary function is increase efficiency of a 2D Tiled Block access. Mentioned below are the main functions of the TILER module:

1. primary handling efficiently 2 dimensional data mapped in tiles, such as video or graphics macro blocks.
2. optionally managing the memory fragmentation and zero-copy physical frame buffers swapping through a page-grained translation, on 4KB page granularity.
3. making isometric – distance preserving – transforms, such as 90°, 180°, or 270° rotations, with either a horizontal or vertical reflection, with no overhead.

Written differently, the functionality of this TILER module is to map a 2D virtually-addressed incoming request into one or more physically-addressed SDRAM requests by:

1. Decoding the incoming access address to qualify whether the request targets the TILER or the memory directly.
2. If the incoming access is a TILER-specific requests, then transforming to the virtual address, data and byte enable to match the requested 0°, 90°, 180°, or 270° orientation in a tiled 2D addressing space.
3. optionally translating the oriented tiled address by a page-specific vector to manage memory fragmentation and physical object aliasing, as shown in Figure 6-9.
4. Splitting tiled requests at tile boundaries and non-tiled requests at DMM atomic section boundaries
5. Requesting the page-based address translation
6. Requesting buffer allocation in the appropriate ROBIN.
7. In case of a write request, allocating and updating an internal write context to subsequently direct the incoming write data into the relevant re-ordering buffer.

**Figure 6-9. Overview of Request Conversion**





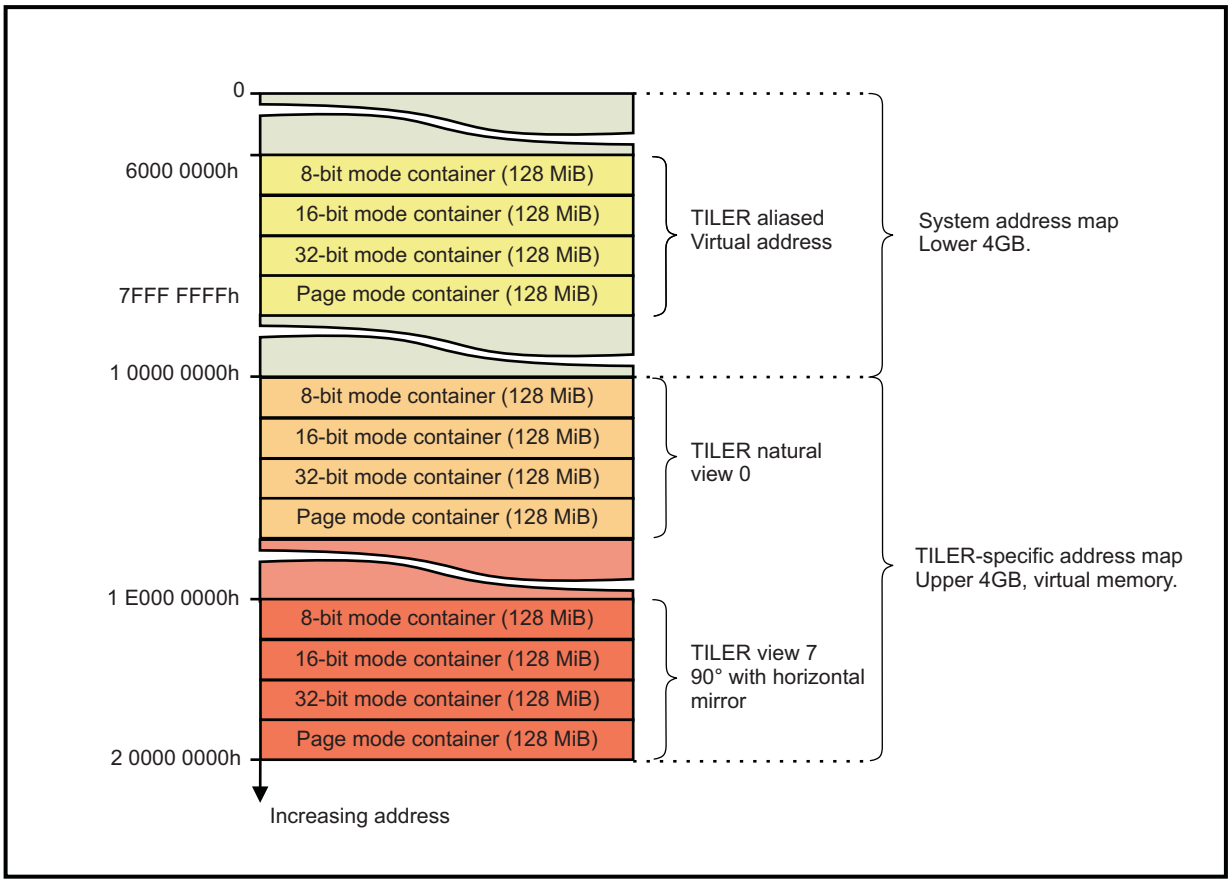
Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.1.7.1 Device Memory Map - For Tiled Data

In the Device, the initiators read and write tiled data by generating accesses to the tiled space by the one of the two methods:

1. For all initiators except, HD_VPSS, the tiled space is accessed in the 512 MB address range of (6000 0000h to 7FFF FFFFh). These initiators use the access the any of 8 views by using the respective DMM_TILER_OR0 or DMM_TILER_OR1 registers.
2. For HD_VPSS, the TILER module supports its own 4-GB virtual addressing space, from address (1 0000 0000h to 1 FFFF FFFFh), and allows any of the eight 512-MB oriented sub-spaces – or views – to be remapped in the system virtual addressing space.

**Figure 6-10. Memory Map**




**NOTE:** Despite the larger virtual address windows, 512 MB in the system memory map and 4GB in the TILER memory map, the maximum size of physical tiled data is only 128MB. This 128MB is co-located in a 128MB aligned space, if using PAT direct access translation. In case of using the optional PAT Indirect access translation, then this 128MB is mapped to any location in the SDRAM, using the LUT tables, at the granularity of 4KB page.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.1.8 Clock

The DMM is a synchronous design and operates from the same clock as the internal L3 interconnect. All timings use this clock as a reference.

#### 6.2.1.9 Interrupts

DMM module generates interrupt to Cortex-A8, for status and error conditions related to PAT refills engines. There 8 different conditions which can be individually enabled or disabled. With 4 Refill engines, there are total 32 unique sources of interrupts.

Refer to Section 6.3.3 for detailed description on using the PAT refill engines.

Per Refill engine, status and error conditions are mentioned below:

1. FILL_DSCn: Refill of any descriptor, for engine n, complete.
2. FILL_LSTn: Refill of the last descriptor, for engine n, complete.
3. ERR_INV_DSCn: Invalid descriptor pointer, for engine n.
4. ERR_INV_DATAn: Invalid data table pointer, for engine n.
5. ERR_UPD_AREAn: Error caused by area register update while refilling, for engine n.
6. ERR_UPD_CTRLn: Error caused by control register update while refilling, for engine n.
7. ERR_UPD_DATAn: Error caused by data register update while refilling, for engine n.
8. ERR_LUT_MISSn: Access to a yet-to-be-filled entry, that is part of the area being refilled, for engine n.

The six error conditions mentioned above are also reported in the respective DMM_PAT_STATUSn register.

#### 6.2.1.10 Address Translations Within DMM, For Tiled Accesses

This section describes address translations that happen at various interfaces within DMM, for tiled accesses.

From a user perspective, the PAT can be seen as an additional step in the process applied to the tiled addresses by the TILER:

- the TILER first transforms a tiled address into a 27 bit physical address which fits within a 128MB address range.
- the PAT then processes the high order bits of the physical address output by the TILER, and outputs a 32-bit physical address which covers a 4GB address range. Section 6.2.1.4 describes the exact details for this address translation.

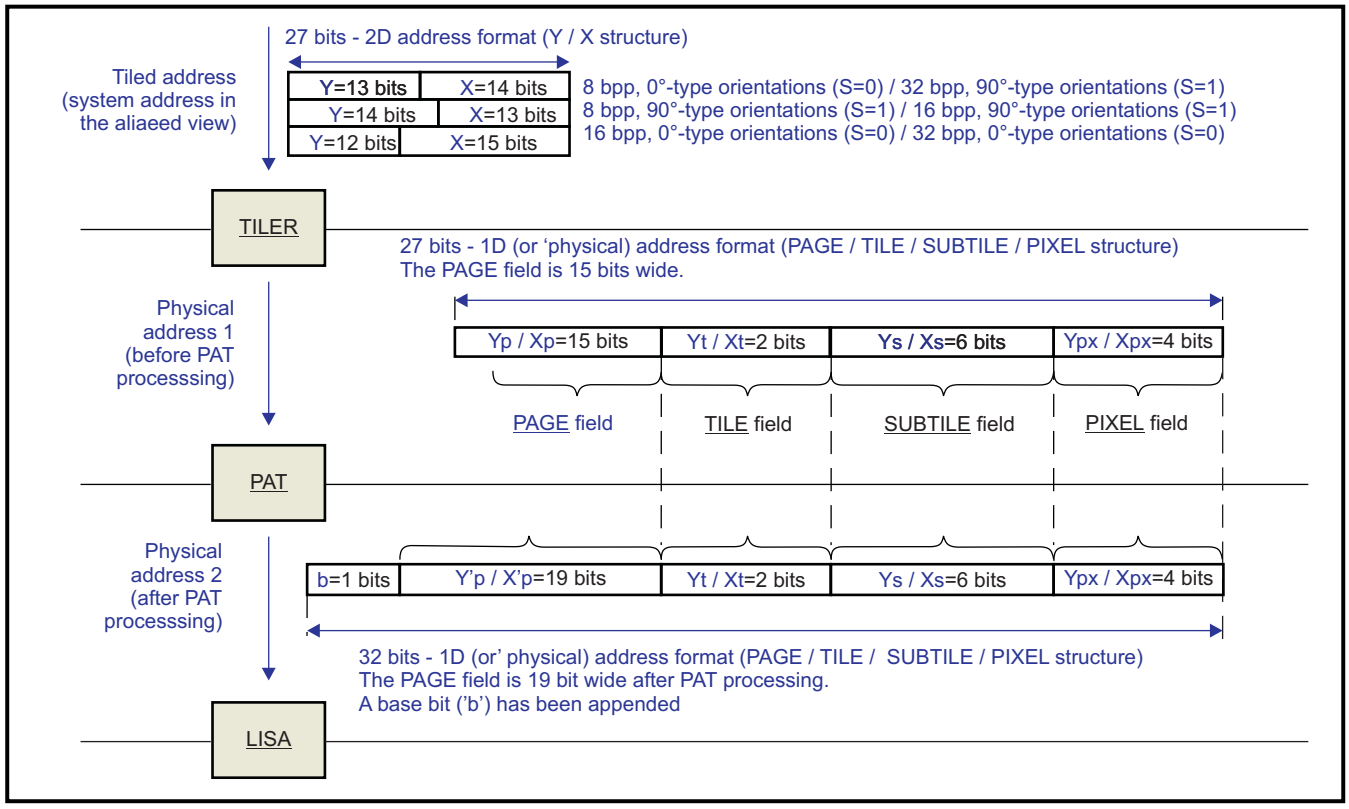
The purpose of the PAT is to map the tiled data anywhere in the 4GB physical address range, with a PAGE granularity (The TILER page is the granularity of physical memory allocation in TILER container. Each page is 4KB). This can be summed-up in Figure 6-11 where the PAT process appears in red.

Only the high order bits of the physical address are modified by the PAT: the 12 low-order bits remain unchanged. This means that the data ordering within each 4kB PAGE remains as calculated by the TILER.

A PAT view is defining the kind of physical address translation to perform for each of the page, 8-bit, 16-bit and 32-bit mode accesses. Each mode in each PAT view can be programmed in 2 different modes (direct translation and indirect translation) which will be described in PAT section. Note that indirect mode is the most commonly used. Direct mode is used only for debug or in case of a DMM without PAT module.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 6-11. DMM Address Translations**





Copyright © 2011–2016, Texas Instruments Incorporated

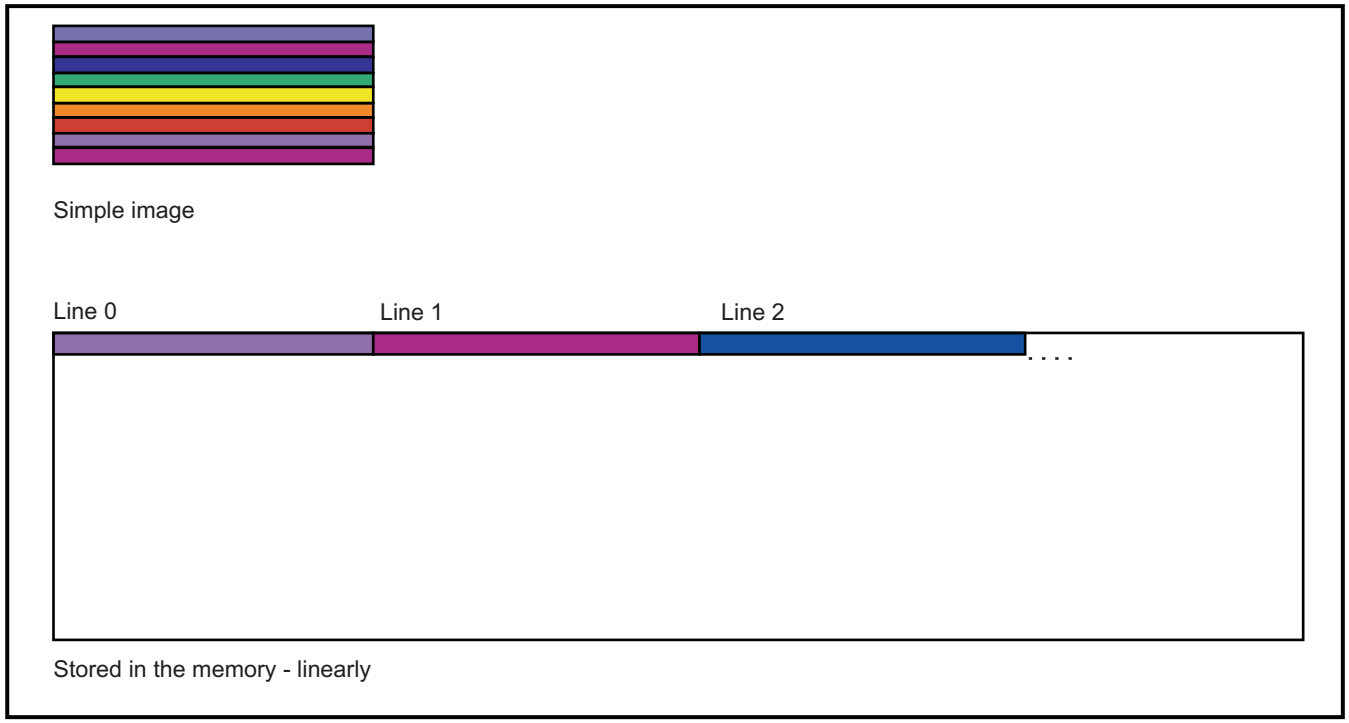
## 6.2.2 TILER Functional Description

### 6.2.2.1 Introduction to TILER

#### 6.2.2.1.1 Need for a TILER

Image processing algorithms like H.264 encode and decode, which works on the principle of spatial locality, access the image data in two-dimensional format, for example, 16 pixels wide by 16 pixels high. If the image is stored in the memory in linear format then it would appear as in Figure 6-12. Thus, to access the macroblock, multiple memory accesses would need to be made. In the device, each memory access can be up to 128 bytes large. However, because macroblock is not stored contiguously in the memory, only partial pixels is useful in each memory access and the rest is discarded and a new access is generated to fetch rest of the data.

**Figure 6-12. Image Stored in the Memory Linearly**



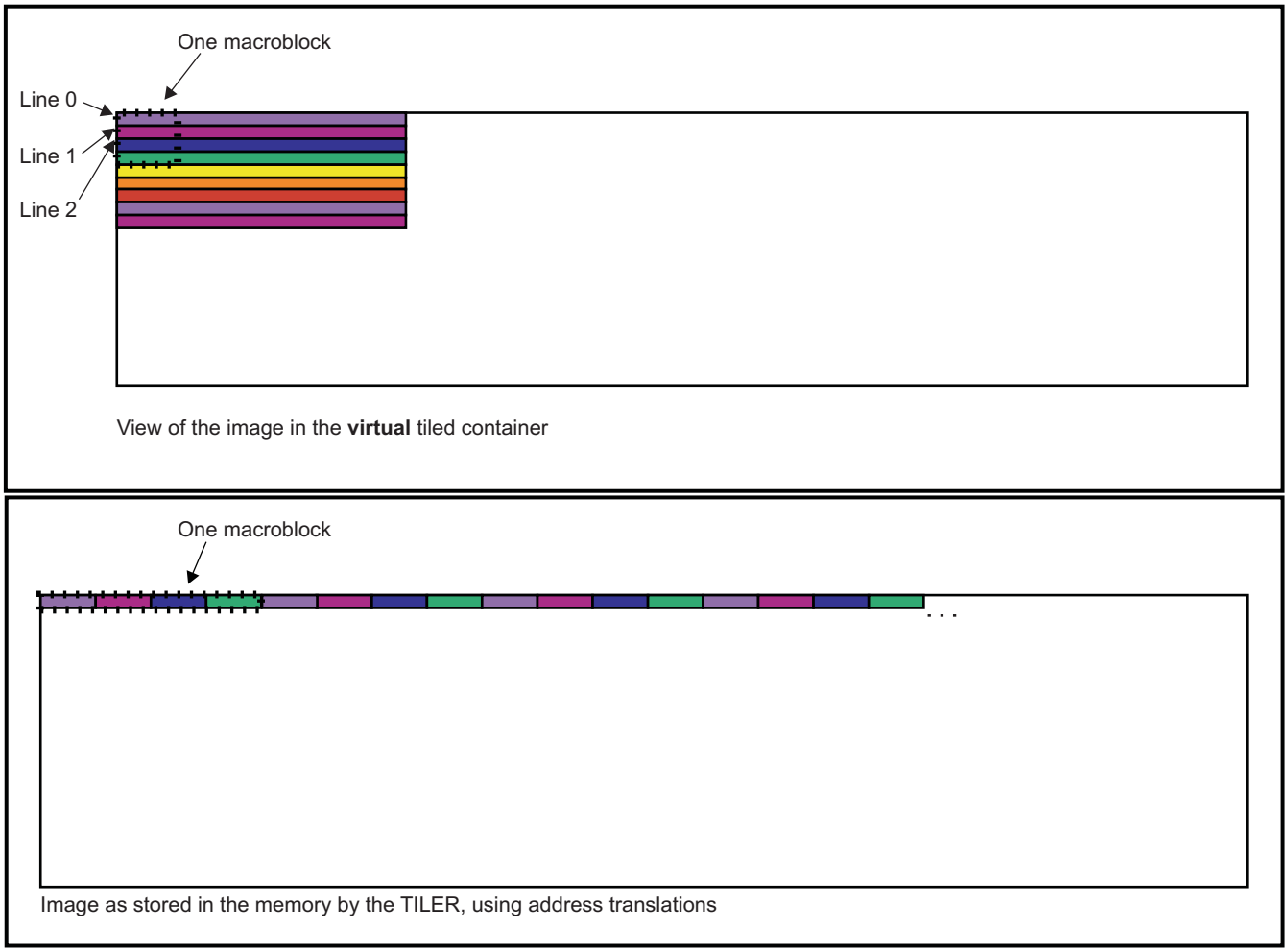

Instead, if the image is stored in bi-dimensional format, much fewer accesses are needed to read the entire macroblock of data, which directly transforms into several advantages:

1. Faster processing of algorithms.
2. Fewer SDRAM page opens.
3. Lesser wastage of memory bandwidth in the system.

This concept is the primary motive for storing the image data in the tiled format.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 6-13. Image Stored in the Memory by TILER**




NOTE : This is only a pictorial representation and actual arrangement of the data in the 128-bit subtile may be different.

#### *6.2.2.1.2 Consuming Tiled Data*

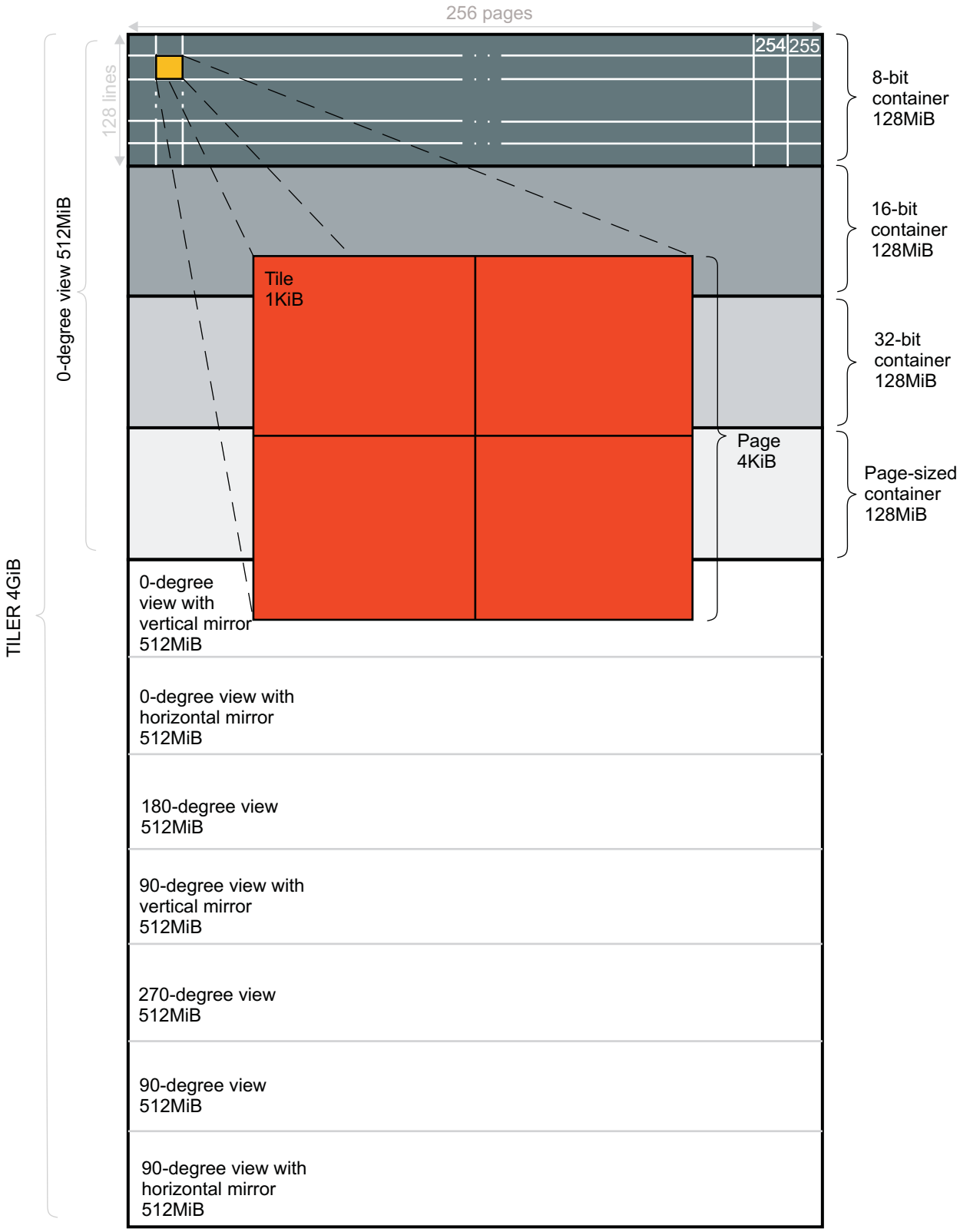
In the device, the DMA engine of HD_VPSS, also known as VPDMA, has internal line buffers of its capture and display paths (called clients). For every tiled access, it is ensured that rastering tiled data has zero overhead, when processing an entire frame, or $2^{2n}$ number of lines.

### 6.2.2.2 TILER Rationale - A Top-down Approach

This section is a synthesis of all TILER concepts, through a top-down approach starting from the main object container, giving one rule per TILER structure level. Figure 6-14 shows the TILER address space structure for tiled modes.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 6-14. Address Space Structure for Tiled Modes**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.2.1 The TILER is a 4-GB Virtual Address Space Composed of Eight Views

A TILER is addressed using a virtual 4 GB address space, containing 8 unique orientation views of 512 MB each. There is one view for each of the eight possible way of scanning a frame-buffer.

1. From left to right then from top to bottom (0 natural orientation)
2. From right to left then from top to bottom
3. From left to right then from bottom to top
4. From right to left then from bottom to top
5. From top to bottom then from left to right
6. From top to bottom then from right to left
7. From bottom to top then from left to right
8. From bottom to top then from right to left

Section 6.2.2.7 summarizes how the view can be visualized on a 128MB TILER container.

#### 6.2.2.2.2 A View is a 512-MB Virtual Address Space Composed of Four Containers

There is one container per element size to allow correct access patterns in any of the eight possible orientations. The container is the entity where all objects of a given element type are allocated. The element is the entity of maximum size - 8 bits, 16 bits, 32 bits or page-sized - which is invariant in any orientation.

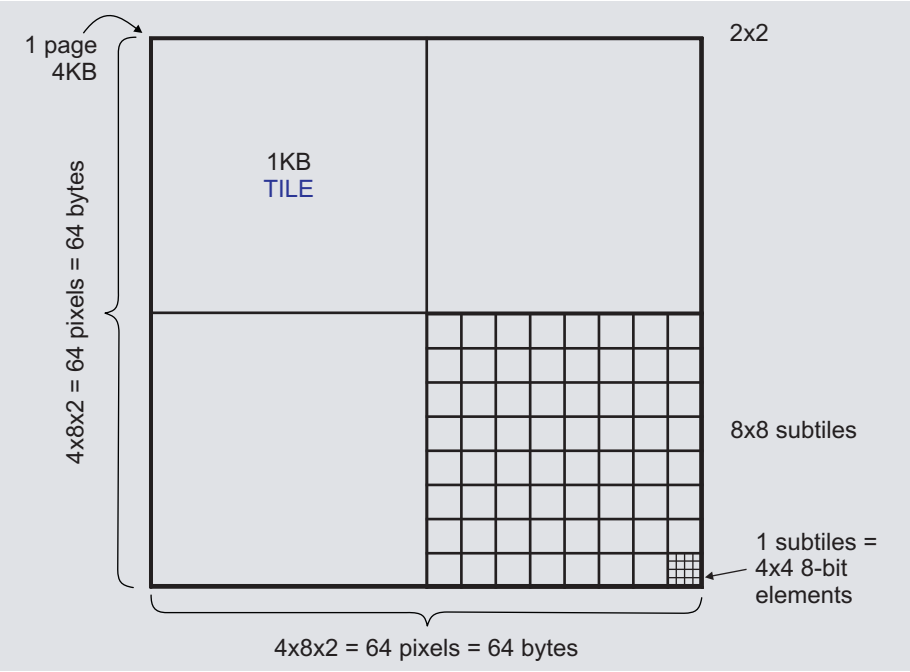
#### 6.2.2.2.3 A Container is a 128-MB Virtual Address Space

A 128 MB container can be visualized as an array of 128 x 256 pages of 4 KB each. The page defines the granularity of physical memory allocation through a physical address translation unit - MMU.

#### 6.2.2.2.4 A Page is a 4-KB Virtual Address Space

A page is composed of two lines. Each line consists of two tiles. Figure 6-15, Figure 6-16, and Figure 6-17 illustrate the Page geometry for 8-,16-, and 32-bit views, respectively.

**Figure 6-15. 4KB Page, 8-bit Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 6-16. 4KB Page, 16-bit Mode**

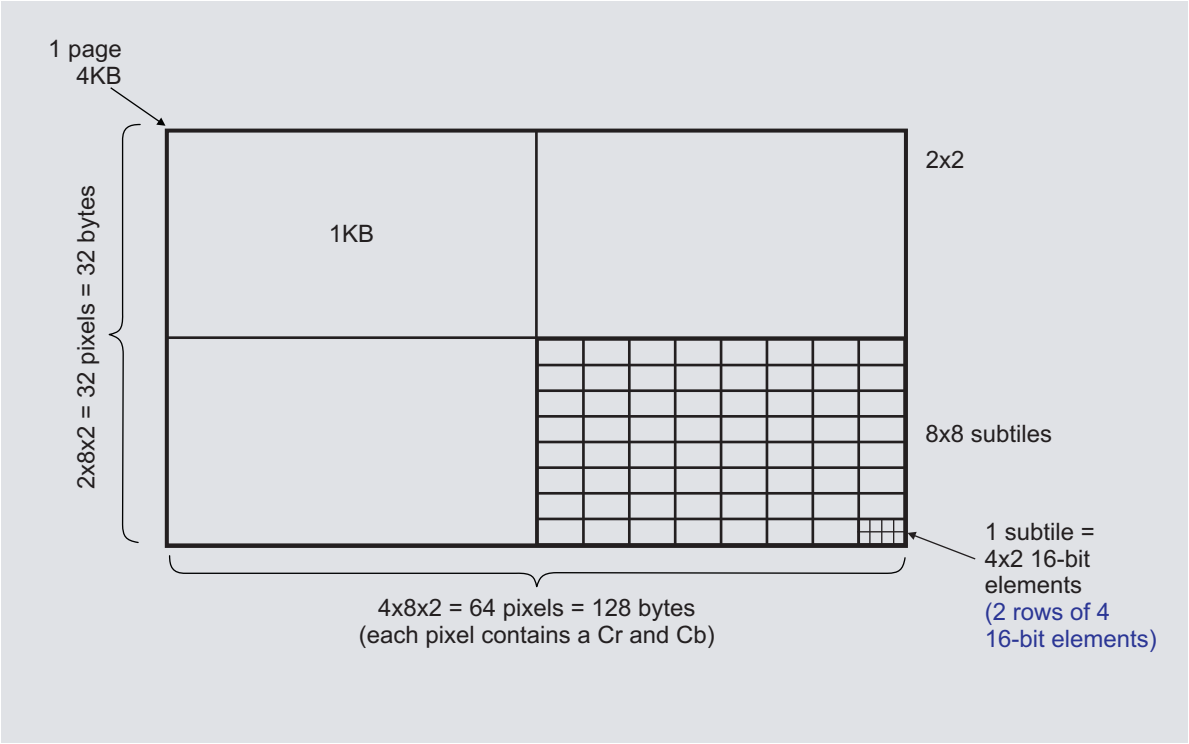
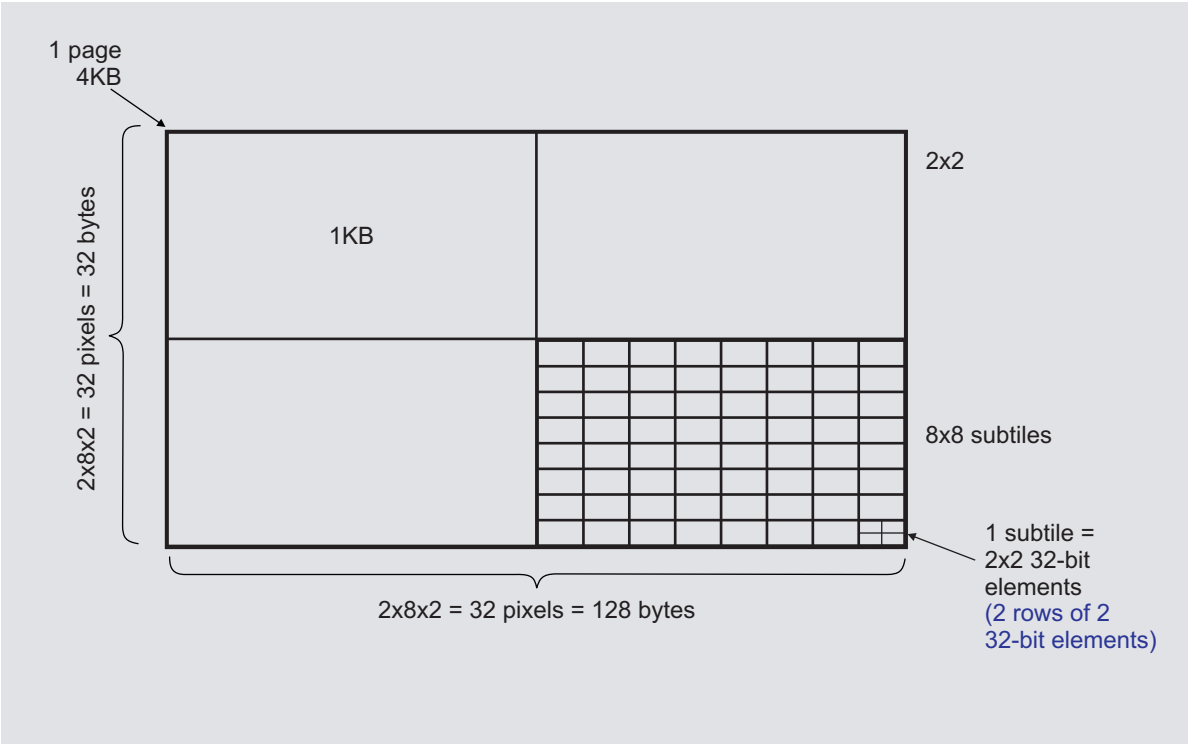



**Figure 6-17. 4KB Page, 32-bit Mode**





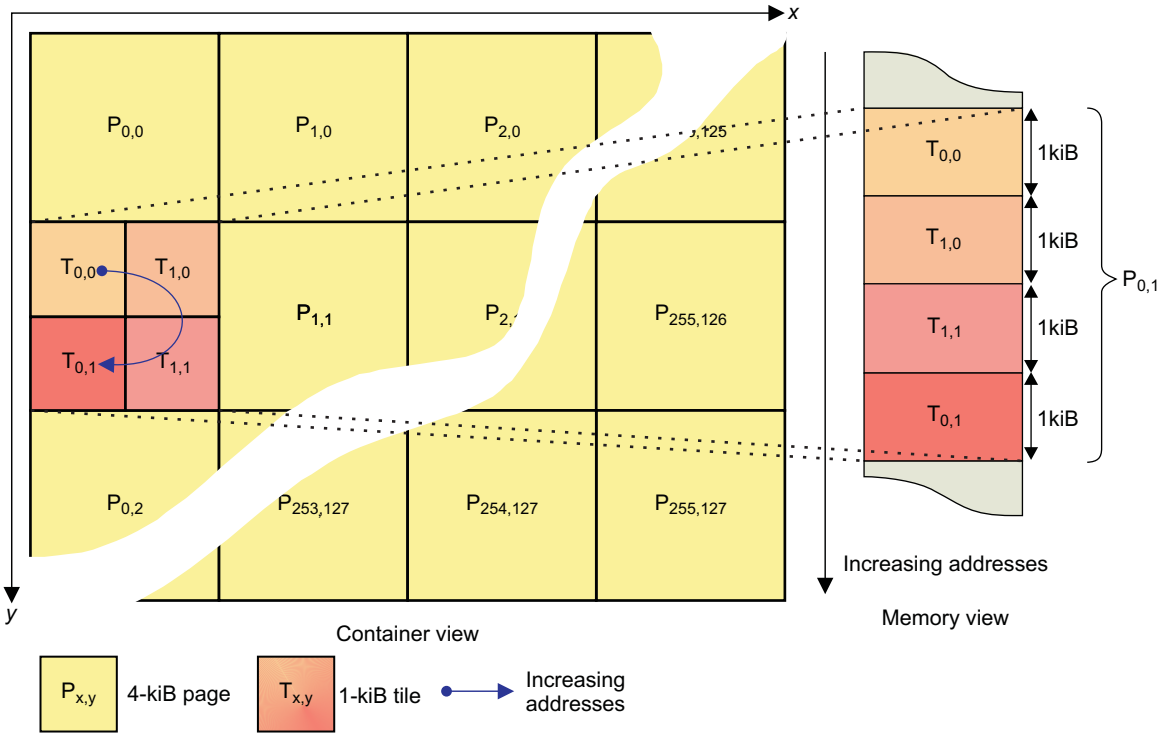
Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.2.5 A Tile is a 1-KB Address Space

The tile is designed to offer bi-dimensional data locality in a single SDRAM page. In this respect, it has been sized to 1 KB (the size of the smallest SDRAM page).

Figure 6-18 shows how 4 tiles are arranged in a page. This arrangement ensures that any two adjacent tiles in a page are not stored on the same EMIF(when interleaving is enabled). Thus a large tiled access will be efficient as it will be evenly spread across both SDRAM controllers.

**Figure 6-18. Four 1KB Tiles in One 4KB Page**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.2.6 A Sub-tile is a 128-bit Address Space

A 1KB tile is further decomposed into 64 sub-tiles, each sized 128 bits. The sub-tile structure is such that it balances accesses in two-dimensional modes, thus improving SDRAM access efficiency.

Furthermore, using Sub-tile pairing, the interlaced accesses to the tiled data can be improved. A Sub-tile is defined as follows:

1. An 8-bit sub-tile (Figure 6-19) is defined as an array of four horizontal lines of four 8-bit data. Thus, in 8-bit tiled mode, the TILER container is a 16384 × 8192 2D array of 8-bit elements.
2. A 16-bit sub-tile (Figure 6-20) is defined as an array of two horizontal lines of four 16-bit data. Thus, in 16-bit tiled mode, the TILER container is a 16384 × 4096 2D array of 16-bit elements.
3. A 32-bit tiled (Figure 6-21) mode as an array of two horizontal lines of two 32-bit data. Thus, in 32-bit tiled mode, the TILER container is an 8192 × 4096 2D array of 32-bit elements.

**Figure 6-19. 8-bit Sub-tile**

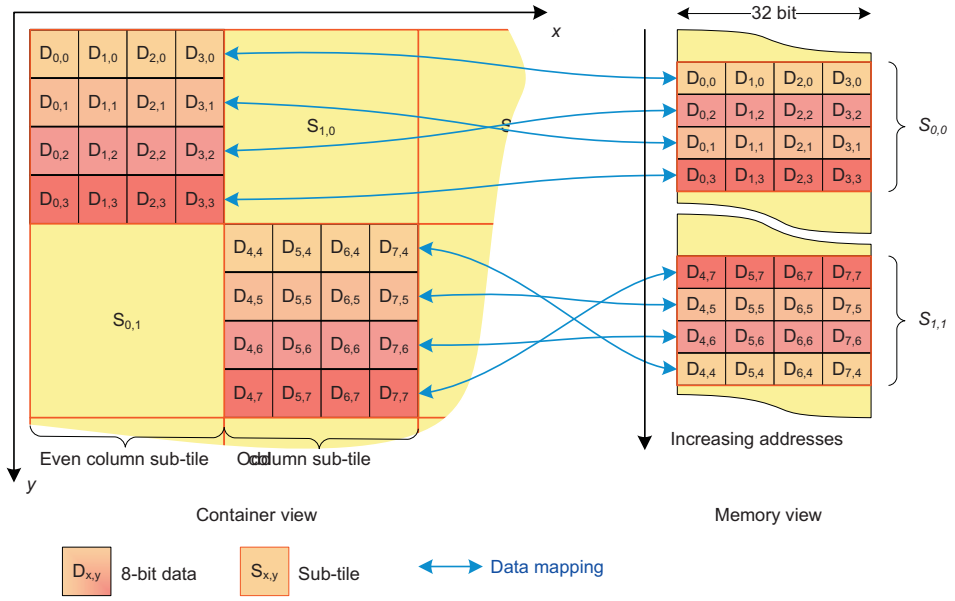
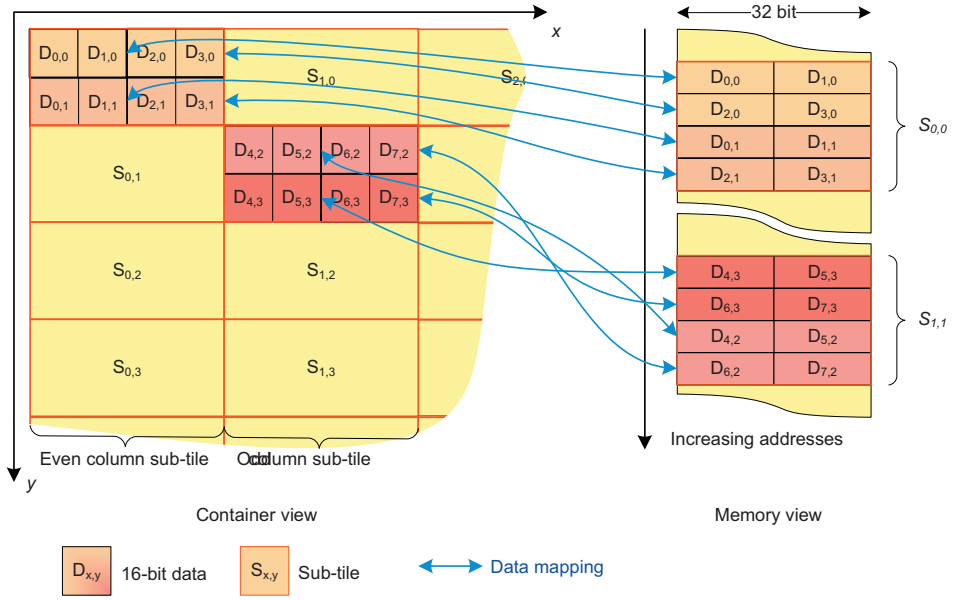


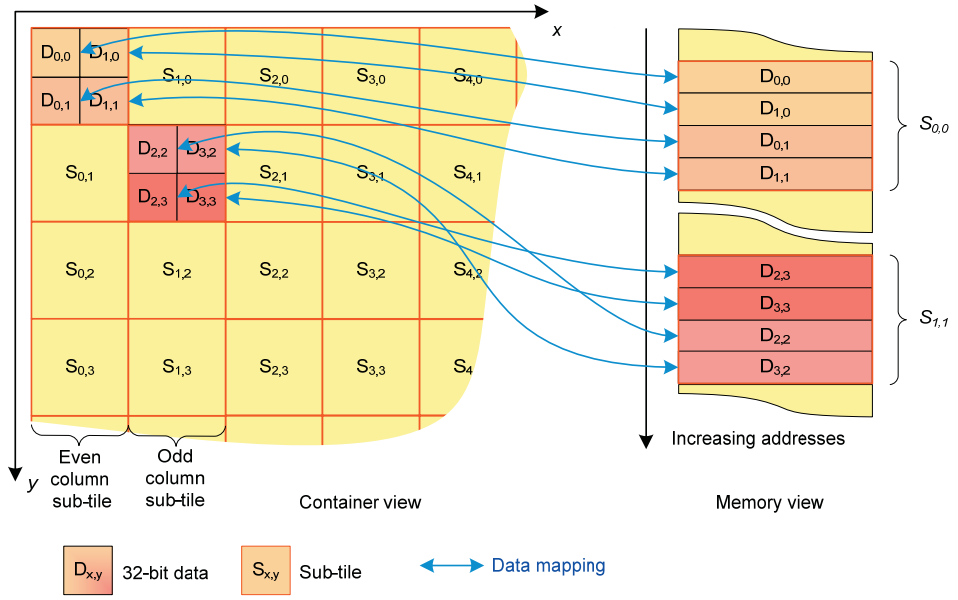

**Figure 6-20. 16-bit Sub-tile**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 6-21. 32-bit Sub-tile**




### 6.2.2.3 TILER Container - Summary

TILER is a 128-MB virtual container arranged in 2 dimensions as 256 × 128 pages, each of 4KB.

Each 4-KB page is has 4 tiles.

Each 1-KB tile has 64 subtiles, each of size 128 bits.

Based on the mode of access, 8/16/32 bit, the data stored can be aligned differently, as explained in the previous section.

#### *6.2.2.3.1 Where Will the Tiled Data Be Stored in Physical Memory?*

Using PAT Direct Access Translation:

By bypassing the LUT, the entire 128MB space has to be contiguous in the physical memory. Then each of the 8-bit 128 MB virtual container, 16-bit 128 MB virtual container, 32-bit 128 MB virtual container and paged container, can be mapped to either the same physical 128 MB space or at different locations. Refer to the PAT_VIEW_MAP register, which defines the base address of this 128 MB space. As can be seen, this base address is 256 MB aligned.

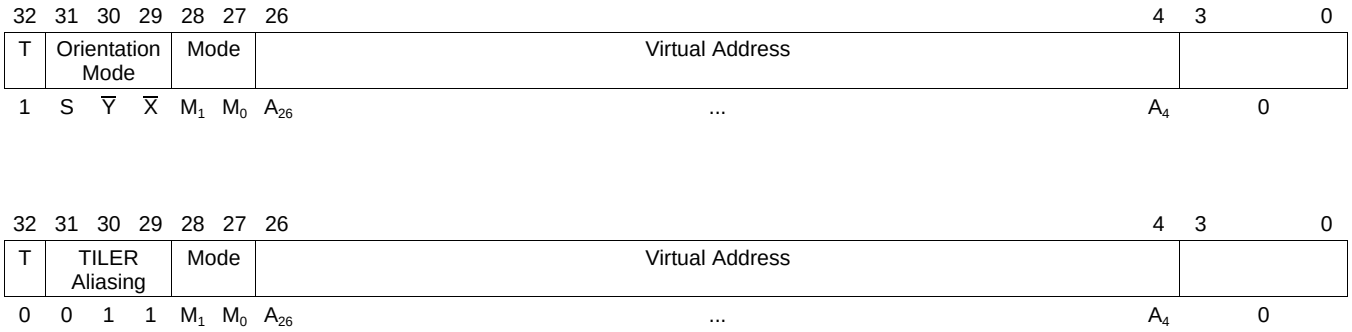
Using PAT In-Direct Access Translation:

A more common use case would be to use the single LUT in the DMM, to map the virtual data to physical address, at 4KB granularity. Then the physical 128MB need not be contiguous. However, with one LUT in the system, only 128 MB of data is possible; therefore the 8, 16, 32-bit and paged containers may need to be aliased to same pages. Thus use the LUT to map 8-bit, 16-bit, 32-bit containers, and paged mode containers, whereby they would all correspond to same physical memory pages, as shown in Figure 6-24.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.3.2 Addressing Formats

**Figure 6-22. Address Format**




1. The 33rd bit, noted T, is aimed at distinguishing the standard 4-GB system address map from the 4-GB TILER-specific address map In the Device, only HD_VPSS can generate accesses in the latter space.
2. The orientation bits, noted S, Y, and X, define the request orientation as specified in Section 6.2.2.7.
3. The mode bits, noted M1 and M0, encoded as in Section 6.2.2.8.1.
4. The remaining 27 bits, noted A0 to A26, define the mode and orientation specific virtual address.

### 6.2.2.4 TILER Modes

The TILER is supporting three major access modes - bypass, page and tiled - each having a specific output request generation.

#### 6.2.2.4.1 Bypass Mode

This mode is transparent at the TILER perspective. Bypass mode is for accesses generated by initiators with System addresses outside the virtual tiled address range, since TILER will bypass PAT. Still, at the DMM perspective:

- 2D block bursts are broken down on a line-basis in a set of incremental bursts
- Incremental bursts - including those issued by a 2D block burst breakdown - are split at the DMM atomic unit of the section hit by the burst at:
  - at the interleaving granularity of the section - 128 byte, 256 byte or 512 byte - in interleaved sections
  - at 1-KB boundary in non-interleaved sections

#### 6.2.2.4.2 Paged Mode

The purpose of this mode is to use the DMM PAT address translation mechanism for non-tiled accesses. In this respect it is similar to the bypass mode:

- 2D block bursts are broken down on a line-basis in a set of incremental bursts
- Incremental bursts - including those issued by a 2D block burst breakdown - are split at:
  - the interleaving granularity of the section - 128 byte, 256 byte or 512 byte - in interleaved sections
  - 1-KB boundary in non-interleaved sections


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.4.3 Tiled Mode

When the being accessed in tiled mode, it can be either of the following:

- A well formed 2D block requests - conforming to the orientation, mode and stride
- A 1D incremental requests and ill-formed 2D block requests

**Table 6-1. Stride for Well-formed Tiled Mode 2D Block Requests**

| Orientation | | | Mode | | Stride (bytes) | Description |
|---|---|---|---|---|---|---|
| S | Y | X | M1 | M0 | | |
| 0 | x | x | 0 | 0 | 16384 | Plain access to an 8-bit progressive frame in 0° or 180° |
| | | | | | 32768 | Field access to an 8-bit interlaced frame in 0° or 180° |
| | | | 0 | 1 | 32768 | Plain access to a 16-bit progressive frame in 0° or 180° |
| | | | | | 65536 | Field access to a 16-bit interlaced frame in 0° or 180° |
| | | | 1 | 0 | 32768 | Plain access to a 32-bit progressive frame in 0° or 180° |
| | | | | | 65536 | Field access to a 32-bit interlaced frame in 0° or 180° |
| 1 | x | x | 0 | 0 | 8192 | Plain access to an 8-bit progressive frame in 90° or 270° |
| | | | | | 16384 | Field access to an 8-bit interlaced frame in 90° or 270° |
| | | | 0 | 1 | 8192 | Plain access to a 16-bit progressive frame in 90° or 270° |
| | | | | | 16384 | Field access to a 16-bit interlaced frame in 90° or 270° |
| | | | 1 | 0 | 16384 | Plain access to a 32-bit progressive frame in 90° or 270° |
| | | | | | 32768 | Field access to a 32-bit interlaced frame in 90° or 270° |

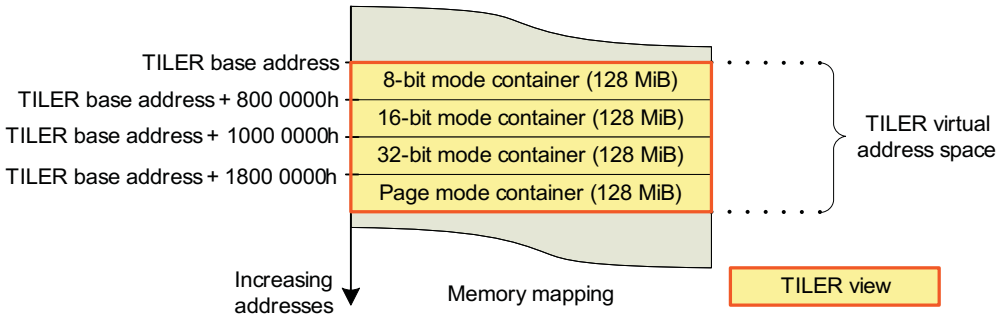
### 6.2.2.5 Object Container Definition

The object container is the unique addressable entry point of the TILER. It is a 128-MB virtual address space, where all objects of a same kind - and orientation - are allocated. Four main types of containers are present in the TILER, each one being referred by a mode:

- A 8-bit element mode, for efficiently accessing bi-dimensional arrays of 8-bit data, example 8-bits per pixel Luma buffers of image.
- A 16-bit element mode, for efficiently accessing bi-dimensional arrays of 16-bit data, example 16-bits per pixel interleaved-Chroma(CbCr) buffers of image.
- A 32-bit element mode, for efficiently accessing bi-dimensional arrays of 32-bit data, example 32-bits per pixel, ARGB graphics buffers.
- A page mode, for efficient 1D accesses

Figure 6-23 shows the TILER object containers and views.

**Figure 6-23. TILER Object Containers and Views**



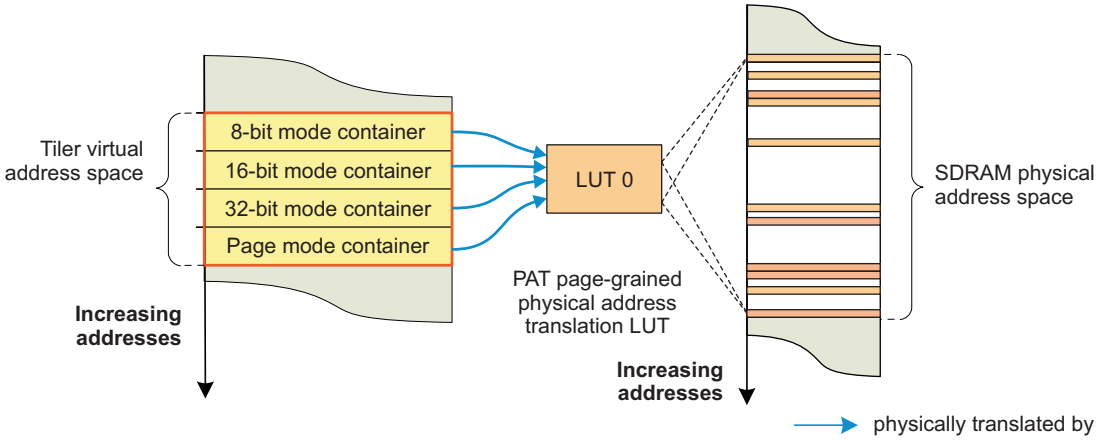


Copyright © 2011–2016, Texas Instruments Incorporated

In the device, there is a single LUT in the PAT. Each LUT can map up to 128MB of objects at a 4KB page granularity. The four modes share the single LUT.

As shown in Figure 6-24, 128MB of objects are available simultaneously in 8, 16, 32-bit modes and paged mode accesses.

**Figure 6-24. Using LUT to Translate Tiled Virtual Address to Physical SDRC Address**




**NOTE:** You need to ensure that an object allocated in a particular container mode does not physically overlap with any other object either in the same mode or other container modes.

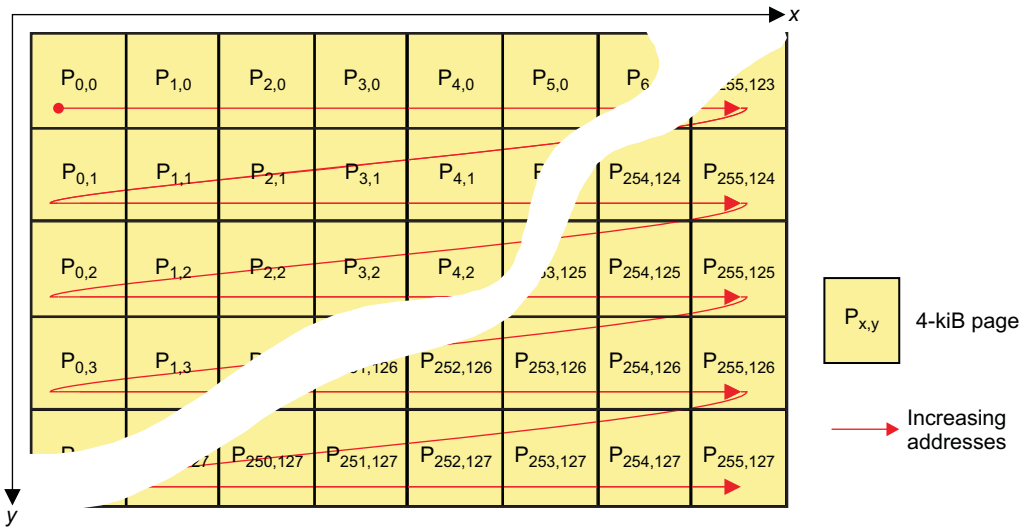
### 6.2.2.6 Page Definition

A TILER page defines the granularity of object allocation in virtual TILER containers. Given that the sub-page structure is mode specific, the 4KB page is the smallest granularity common to all modes, making it the granularity to consider in the TILER resource manager for object allocation.

#### *6.2.2.6.1 Container Geometry with 4-kiB Pages*

As pages size is 4KB, any 128MB object container is a set of 256 × 128 = 32768 pages, organized in an array of 256 columns and 128 rows, as shown in Figure 6-25.

**Figure 6-25. Object Container Geometry with 4KB Pages**

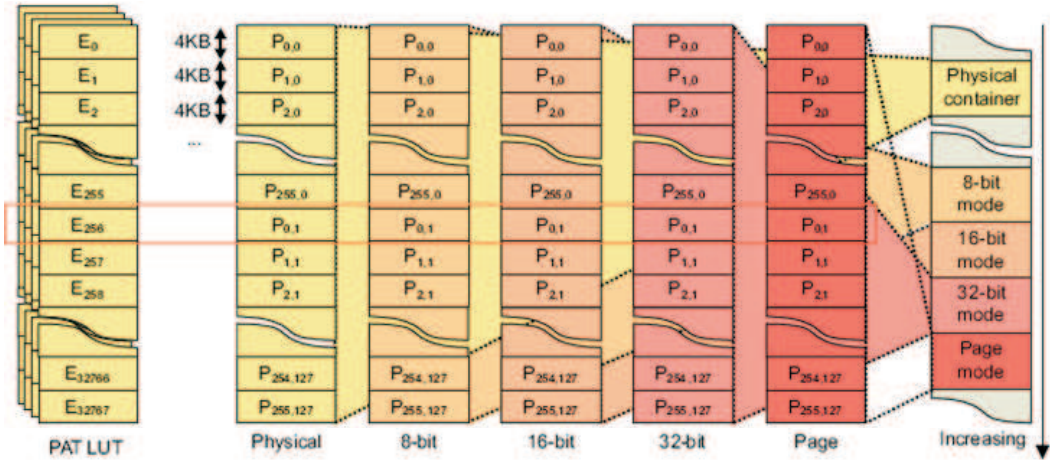




Copyright © 2011–2016, Texas Instruments Incorporated

This array of pages is mapped to the system address space as shown in Figure 6-26.

**Figure 6-26. TILER Page Mapping When Using 4KB Pages**




In any 128-MB object container, the 4KB page Px,y at column x (0 ≤ x < 256) and row y (0 ≤ y < 128), is found at an offset of 4096. (x + 256.y) bytes from the base address of the related object container.

Similarly, the page Px,y at column x (0 ≤ x < 256) and row y (0 ≤ y < 128), is translated by the LUT entry Ex + 256.y found at the index x + 256.y.

#### *6.2.2.6.2 Container Geometry and Page Mapping Summary*

The TILER has a page size of 4096 bytes. The page Px,y:

- has maxx = 256 and maxy = 128
- is found at an offset of: 4096.(x + maxx.y) bytes from the base address of the related object container
- is translated by the entry found at the index (x + maxx.y) of the LUT table

### 6.2.2.7 Orientation

This section describes the eight on-the-fly orientation-related isometric transforms, corresponding to all available changes of orthonormal basis in the TILER container bi-dimensional space.

Figure 6-27 shows isometric transforms in the TILER container.

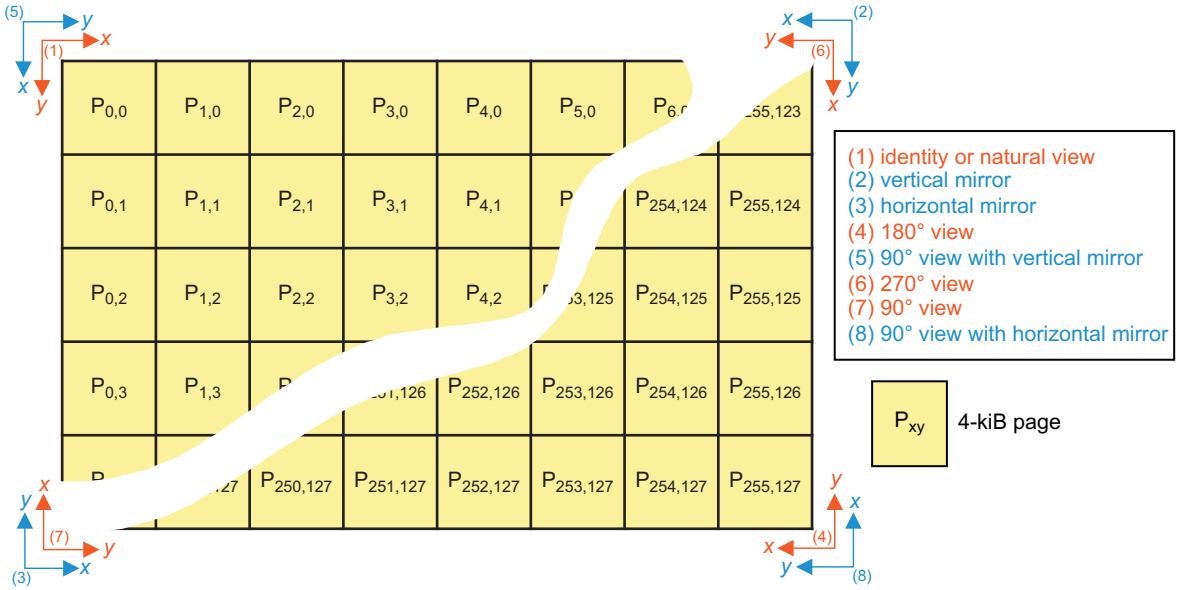
Mathematically speaking, all these transforms correspond to the composition of a 0º, 90º, 180º, or 270º rotation with an optional reflection. The nature of this orientation is based on the three following binary parameters:

- X to change the direction of the x axis of the TILER container
- Y to change the direction of the y axis of the TILER container
- S to swap the modified x and y axis

From now on, and in the rest of this document, the term orientation refers to any composition of a "quadrant" rotation with an optional horizontal - flip-flop - or vertical mirroring.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 6-27. Isometric Transforms in the TILER Container**




**Selection of orientation:**

1. For all initiators except, HD_VPSS, the tiled space is accessed in the 512 MB address range of (6000 0000h to 7FFF FFFFh). These initiators use the access the any of the 8 views by using the respective DMM_TILER_OR0 or DMM_TILER_OR1 registers.
2. For HD_VPSS, the TILER module supports its own 4-GB virtual addressing space, from address (1 0000 0000h to 1 FFFF FFFFh). These eight 512MB spaces correspond to the 8 availables views. With multiple display, capture and transcode path, using the unique 512 MB address space, these ports of HD_VPSS can simultaneously generate access in any of the 8 views.

### 6.2.2.8 Tile Definition

A tile is a subdivision of a page, aimed at:

- Representing a 2D block to better balance the accesses in both directions
- Ensuring that any "tiled" access that fits within a tile is made atomic in the SDRAM controller, and fits in a single SDRAM memory page
- Minimising the number of SDRAM page openings per 2D block transfer

The tile is defined as a 1KB 2D block, and a 4KB page as an array of 2 lines of 2 tiles each, as described in section detailing a 4KB page.

When the considered page is in an interleaved DMM section, it is necessary that:

- the DMM memory interleaving size of tiled accesses is set to 1 KB - a tile size - so that any tiled request that fits within a tile, fits in a single SDRAM memory page
- any request that spans over 2 or 4 tiles, is distributed on a maximum number of SDRAM memory controllers


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.1 TILER Virtual Addressing

The TILER can be virtually accessed in four different modes, namely 8-bit, 16-bit, 32-bit and page modes. Each mode defines the element granularity to apply isometric transforms, as summarized in Table 6-2.

**Table 6-2. TILER Modes**

| Mode | Name | Granularity (element size) |
|---|---|---|
| 0 | 8-bit tiled mode | 8 bits |
| 1 | 16-bit tiled mode | 16 bits |
| 2 | 32-bit tiled mode | 32 bits |
| 3 | Page mode | 4096 bytes |

For instance, making a vertical mirroring of a 16-byte horizontal line containing the word 0001 0203 0405 0607 0809 0A0B 0C0D 0E0Fh, leads to:

- 0F0E 0D0C 0B0A 0908 0706 0504 0302 0100h in 8-bit tiled mode
- 0E0F 0C0D 0A0B 0809 0607 0405 0203 0001h in 16-bit tiled mode
- 0C0D 0E0F 0809 0A0B 0405 0607 0001 0203h in 32-bit tiled mode
- 0001 0203 0405 0607 0809 0A0B 0C0D 0E0Fh in page mode - unchanged as the element granularity is 4KB

Besides, as each of the eight orientations is available for any of the four modes, this gives 32 TILER addressing possibilities.

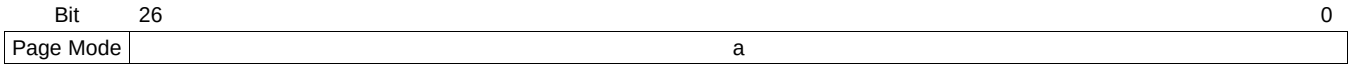
**NOTE:** The format that the TILER manages the tiled data is dictated by which 4 modes (8-,16-, 32-bit or paged) and which orientation (8 views) is used while accessing (read and write) the data.

As long as the data accessed is written and read in the same mode, it is assured that the response to the access will be correct. While accessing in the same mode, if written in one view and read back in the other view, then the read back data will be rotated and/or mirrored as per the view used.

#### 6.2.2.8.2 Page Mode Virtual Addressing and Characteristics

When used in page mode, the 128-MiB TILER space is seen as an orientation-specific sequence of 32768 pages of 4 kiB each. The access sequence inside a page is left unchanged. Therefore, in page mode, the TILER is accessed similarly to any 128-MiB memory, with a 27-bit byte-based address.

**Figure 6-28. Paged Mode Addressing**




#### 6.2.2.8.3 Tiled Mode Virtual Addressing and Characteristics

When used in tiled mode, the 128-MiB TILER space is seen as a giant frame-buffer - the container. The addressing and characteristics of this giant frame-buffer depends on:

- The tiled mode, which defines the considered atomic element size
- The orientation, which potentially swaps its x and y axis - and hence the container geometry

Table 6-3 summarizes tiled mode container characteristics.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 6-3. Tiled Mode Container Characteristics**

| Orientation | | | Element Size (bits) | Width (elements) | Height (elements) | Stride (bytes) | |
|---|---|---|---|---|---|---|---|
| S | Y | X | | | | Progressive | Interlaced |
| 0 | x | x | 8 | 16384 | 8192 | 16384 | 32768 |
| | | | 16 | 16384 | 4096 | 32768 | 65536 |
| | | | 32 | 8192 | 4096 | 32768 | 65536 |
| 1 | x | x | 8 | 8192 | 16384 | 8192 | 16384 |
| | | | 16 | 4096 | 16384 | 8192 | 16384 |
| | | | 32 | 4096 | 8192 | 16384 | 32768 |

As a result, the coordinate (x0, y0) of a pixel in an oriented view is translated in a virtual address as shown in Figure 6-29 and Figure 6-30, for all the four orientations.

**Figure 6-29. Tiled Mode Addressing in 0° or 180° Orientations, (S = 0)**

| Bit | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8-bit tiled mode | $y_0$ | | | | | | | | | | | | | $x_0$ | | | | | | | | | | | | | |
| 16-bit tiled mode | $y_0$ | | | | | | | | | | | | | $x_0$ | | | | | | | | | | | | | |
| 32-bit tiled mode | $y_0$ | | | | | | | | | | | | | $x_0$ | | | | | | | | | | | | | |

**Figure 6-30. Tiled Mode Addressing in 90° or 270° Orientations, (S = 1)**

| Bit | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8-bit tiled mode | $y_0$ | | | | | | | | | | | | | | $x_0$ | | | | | | | | | | | | |
| 16-bit tiled mode | $y_0$ $x_0$ | | | | | | | | | | | | | | $x_0$ | | | | | | | | | | | | |
| 32-bit tiled mode | $y_0$ | | | | | | | | | | | | | $x_0$ | | | | | | | | | | | | | |

Having understood the characteristic of the tiled container and the how the data is stored in bi-dimensional manner, it is easy to understand how stride is calculated. Stride is calculated as the number bytes between pixel (m,n) and pixel (m,(n+1)), where m = x co-ordinate of the image buffer, n = y co-ordinate of the image buffer.

**For S = 0, that is, all views with 0º and 180º:**

- 8-bit mode: stride_8bitmode = 4 bytes per sub-tile × 1 byte height of a sub-tile line × 8 sub-tiles per tile × 2 tiles per page × 256 pages = 4 × 1 × 8 × 2 × 256 = 16KB
- 16-bit mode: stride_16bitmode = 4 bytes per sub-tile × 2 bytes height of a sub-tile line × 8 sub-tiles per tile × 2 tiles per page × 256 pages = 4 × 2 × 8 × 2 × 256 = 32KB
- 32-bit mode: stride_32bitmode = 4 bytes per sub-tile × 2 bytes height of a sub-tile line × 8 sub-tiles per tile × 2 tiles per page × 256 pages = 4 × 2 × 8 × 2 × 256 = 32KB

For S = 1, that is, all views with 90º and 270º:

- 8-bit mode: stride_8bitmode = 4 bytes per sub-tile × 1 byte height of a sub-tile line × 8 sub-tiles per tile × 2 tiles per page × 128 pages = 4 × 1 × 8 × 2 × 128 = 8KB
- 16-bit mode: stride_16bitmode = 4 bytes per sub-tile × 1 byte height of a sub-tile line × 8 sub-tiles per tile × 2 tiles per page × 128 pages = 4 × 1 × 8 × 2 × 128 = 16KB
- 32-bit mode: stride_32bitmode = 4 bytes per sub-tile × 2 bytes height of a sub-tile line × 8 sub-tiles per tile × 2 tiles per page × 128 pages = 4 × 2 × 8 × 2 × 128 = 16KB


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.4 Element Ordering in the TILER Container

This section highlights how elements - 8-bit data, 16-bit data, 32-bit data or 4-kiB page - are ordered in the container. In other words, this section highlights how the path of incrementing virtual addresses is mapped in the container.

It is interesting to note that whatever the mode, and hence the element size, the sequence for ordering the elements in their related container is strictly similar and only depends on the related orientation. In other words:

- Mode is about element granularity
- Orientation is about change of orthornormal basis for ordering the elements in the mode-specific container

A corollary to the previous statements is that in a given mode the internal structure of an element is unchanged whatever the orientation. In page mode for instance, the offset of a word inside a page is invariant by orientation; the content of a page is always accessed in the same manner.

In the next sections, the natural container orthonormal basis is referenced as: (Xn , Yn) and the oriented orthonormal basis as: (Xo , Yo).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.4.1 Natural View or 0º View (Orientation 0)

This orientation defined by S = 0, Y = 0, and X = 0 and means that the operated change of basis is:

X0 = Xn

Y0 = Yn

In any TILER mode, the elements are then ordered from left-to-right and then from top-to-bottom in their container, as shown in Figure 6-31 and Figure 6-32.

**Figure 6-31. Tiled Mode Ordering of Elements in Natural View**

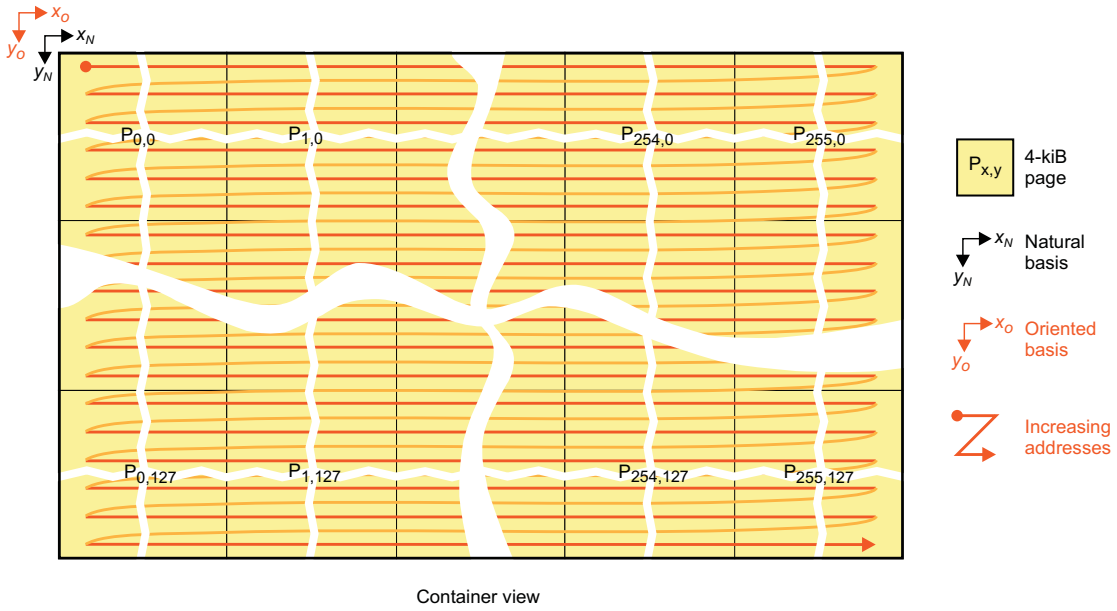
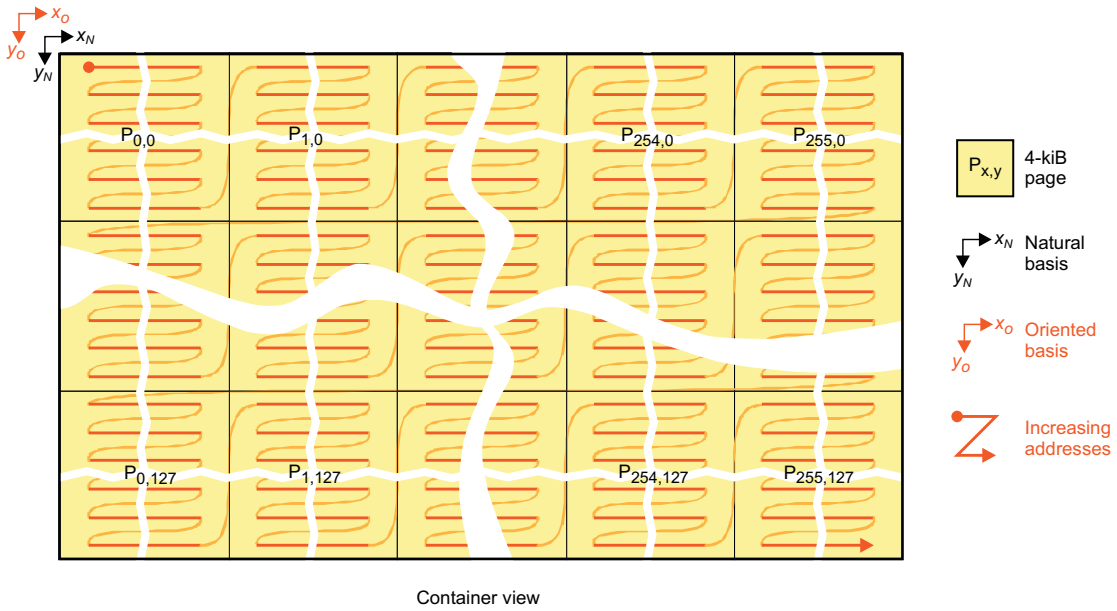



**Figure 6-32. Page Mode Ordering of Elements in Natural View**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.4.2 0º View with Vertical Mirror or 180º View with Horizontal Mirror (Orientation 1)

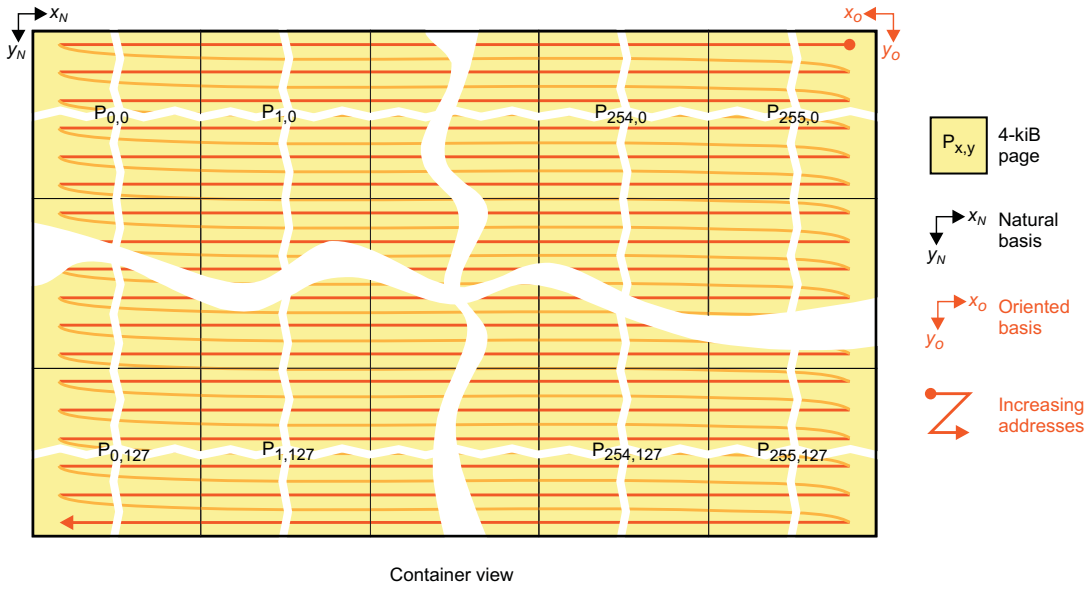
This orientation defined by S = 0, Y = 0, and X = 1 and means that the operated change of basis is:

X0 = -Xn

Y0 = Yn

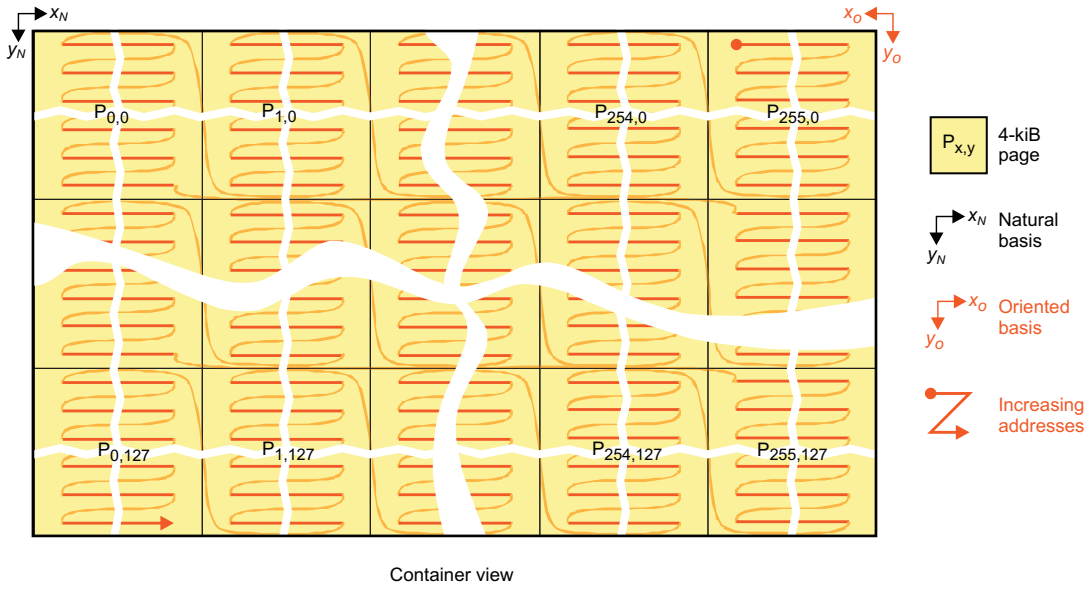
In any TILER mode, the elements are then ordered from right-to-left and then from top-to-bottom in their container, as shown in Figure 6-33 and Figure 6-34.

**Figure 6-33. Tiled Mode Ordering of Elements in 0° View with Vertical Mirror**




Container view

**Figure 6-34. Page Mode Ordering of Elements in 0° View with Vertical Mirror**




Container view



Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.4.3 0º View with Horizontal Mirror or 180º View with Vertical Mirror (Orientation 2)

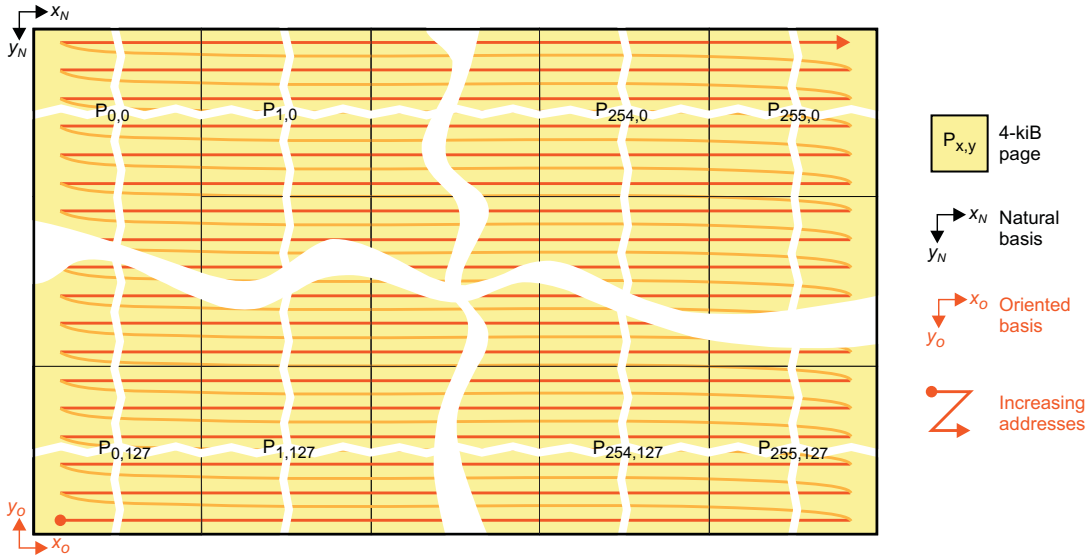
This orientation defined by S = 0, Y = 1, and X = 0 and means that the operated change of basis is:

X0 = Xn

Y0 = -Yn

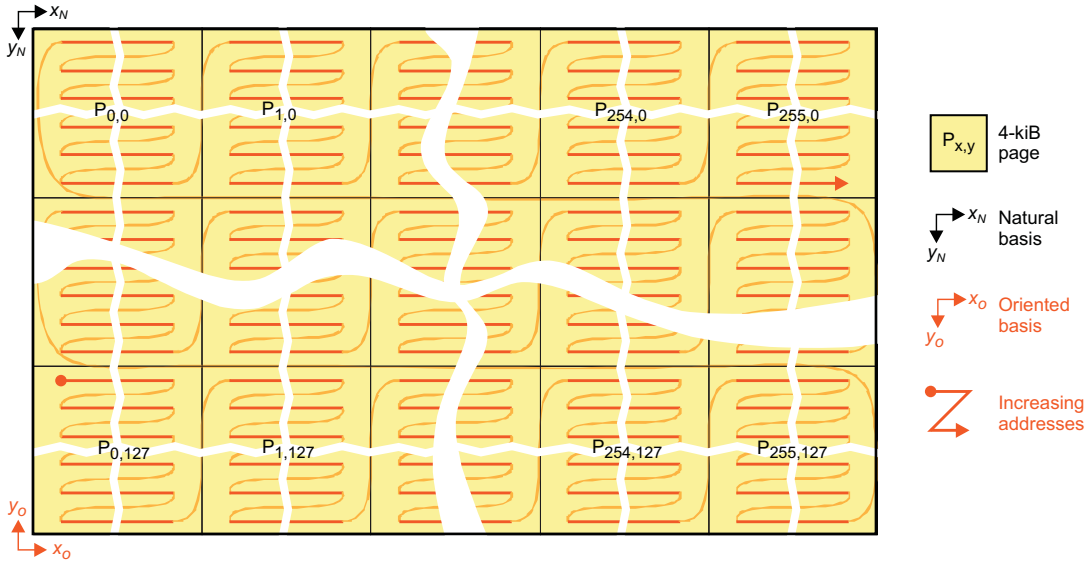
In any TILER mode, the elements are then ordered from left-to-right and then from bottom-to-top in their container, as shown in Figure 6-35 and Figure 6-36.

**Figure 6-35. Tiled Mode Ordering of Elements in 0° View with Horizontal Mirror**




Container view

**Figure 6-36. Page Mode Ordering of Elements in 0° View with Horizontal Mirror**




Container view


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.4.4 180º View (Orientation 3)

This orientation defined by S = 0, Y = 1, and X = 1 and means that the operated change of basis is:

X0 = -Xn

Y0 = -Yn

In any TILER mode, the elements are then ordered from right-to-left and then from bottom-to-top in their container, as shown in Figure 6-37 and Figure 6-38.

**Figure 6-37. Tiled Mode Ordering of Elements in 180° View**

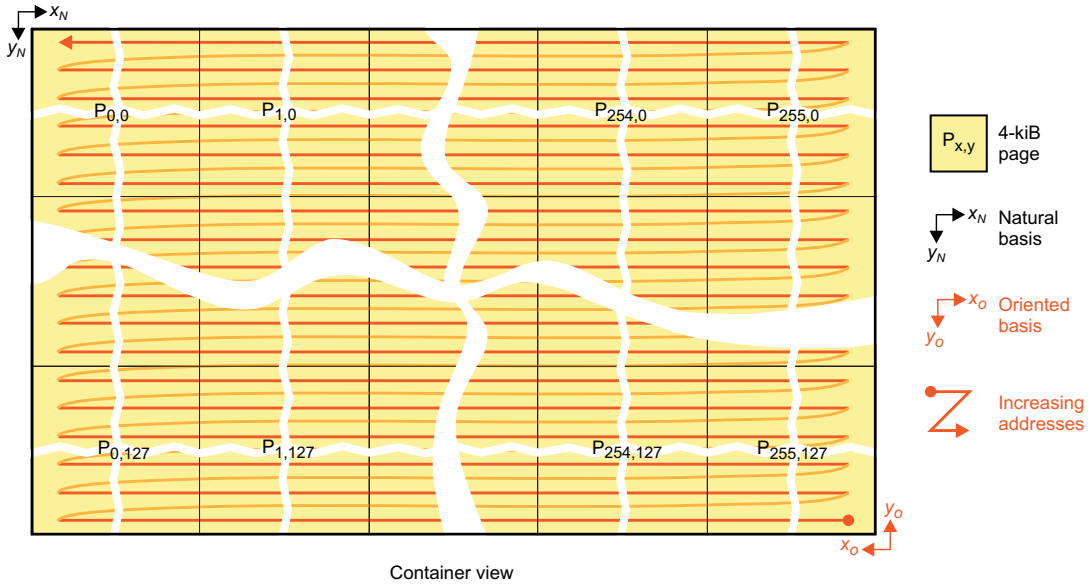
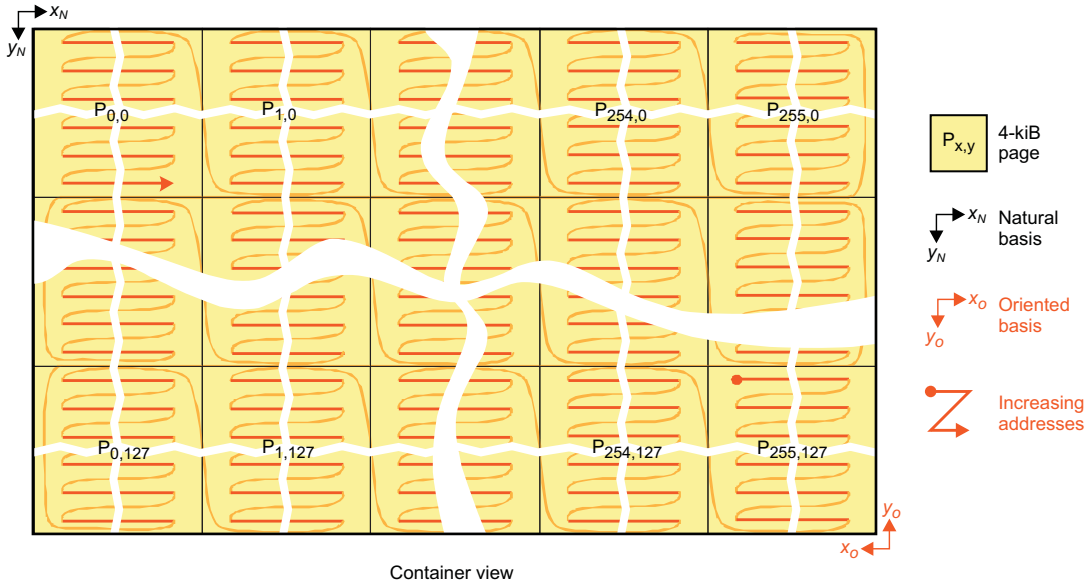



**Figure 6-38. Page Mode Ordering of Elements in 180° View**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.4.5 90º View with Vertical Mirror or 270º View with Horizontal Mirror (Orientation 4)

This orientation defined by S = 1, Y = 0, and X = 0 and means that the operated change of basis is:

X0 = Yn

Y0 = Xn

In any TILER mode, the elements are then ordered from top-to-bottom and then from left-to-right in their container, as shown in Figure 6-39 and Figure 6-40.

**Figure 6-39. Tiled Mode Ordering of Elements in 90º View with Vertical Mirror**

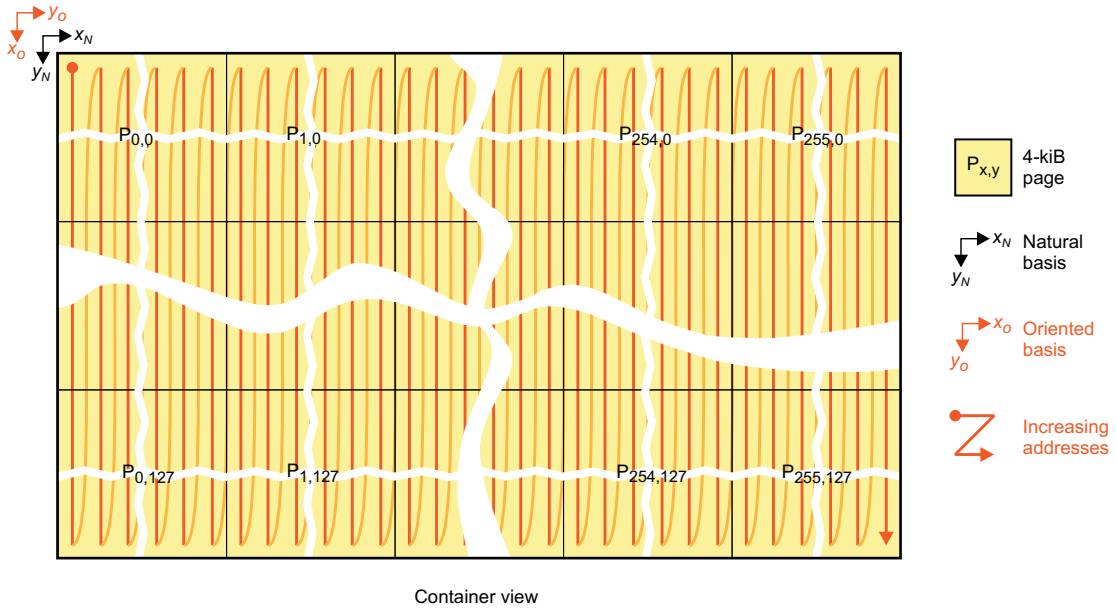
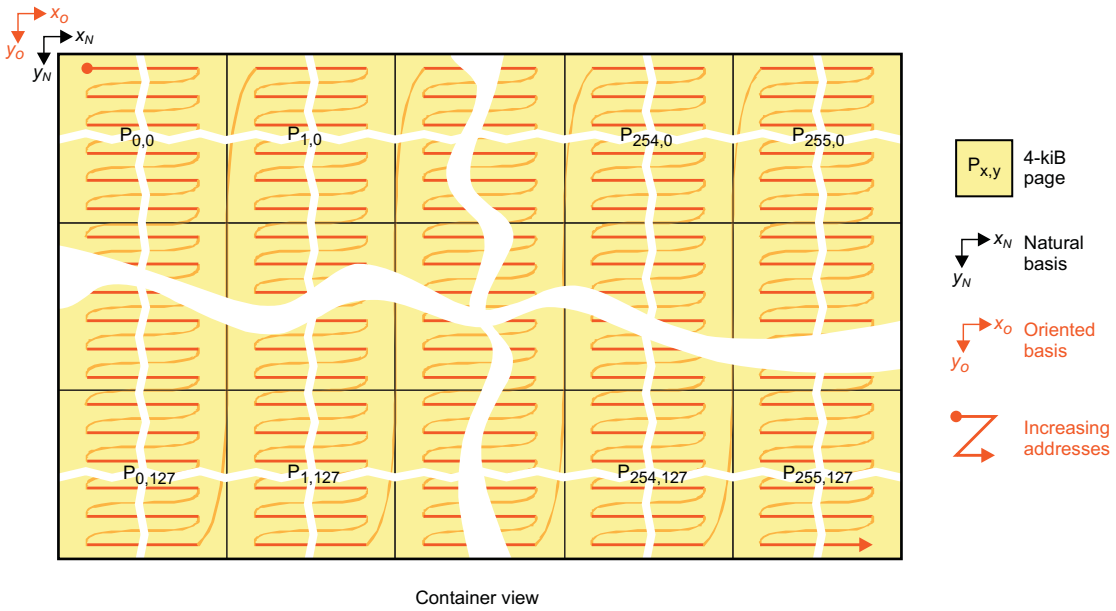



**Figure 6-40. Page Mode Ordering of Elements in 90º View with Vertical Mirror**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.4.6 270º View (Orientation 5)

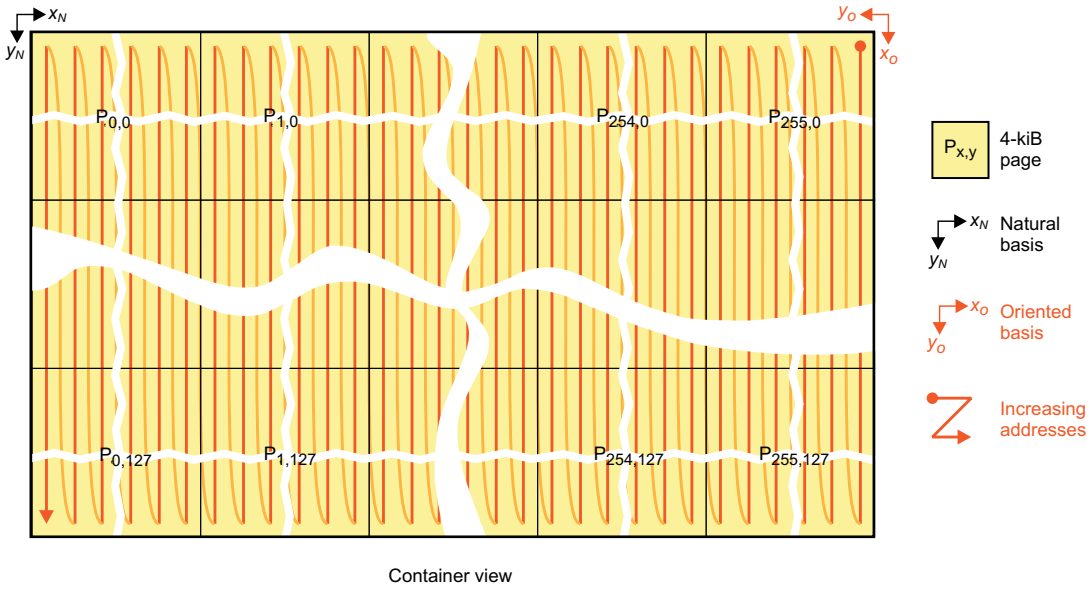
This orientation defined by S = 1, Y = 0, and X = 1 and means that the operated change of basis is:

X0 = -Yn

Y0 = -Xn

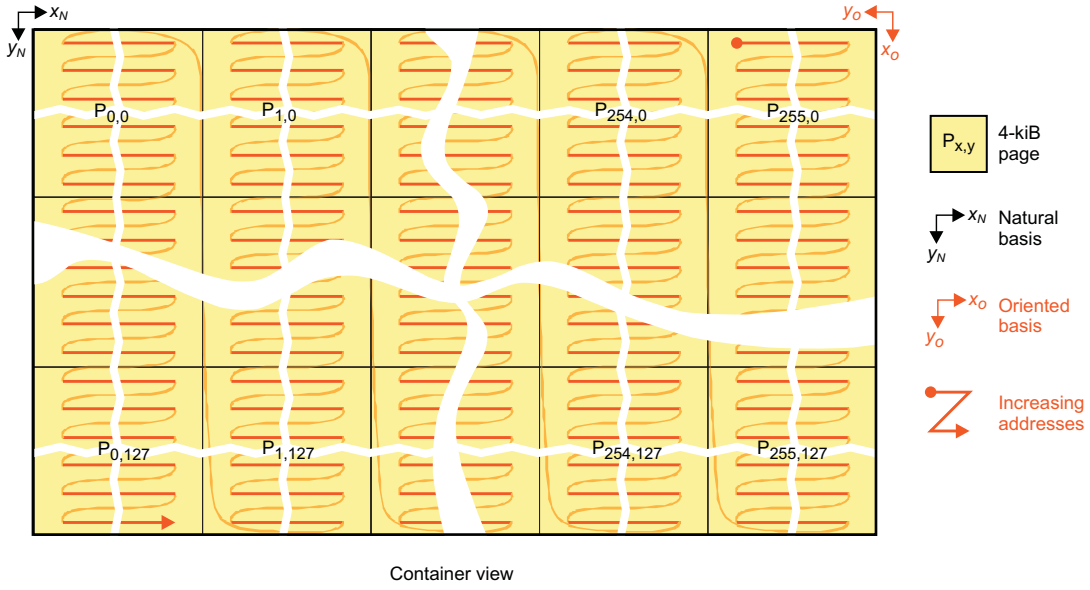
In any TILER mode, the elements are then ordered from top-to-bottom and then from right-to-left in their container, as shown in Figure 6-41 and Figure 6-42.

**Figure 6-41. Tiled Mode Ordering of Elements in 270° View**




Container view

**Figure 6-42. Page Mode Ordering of Elements in 270° View**




Container view



Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.4.7 90º View (Orientation 6)

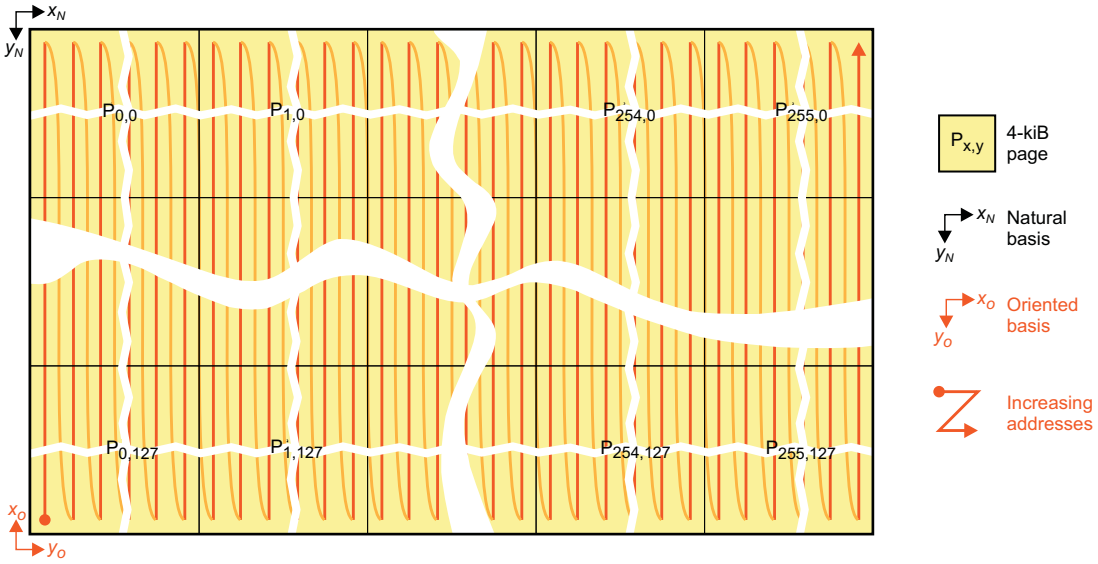
This orientation defined by S = 1, Y = 1, and X = 0 and means that the operated change of basis is:

X0 = -Yn

Y0 = Xn

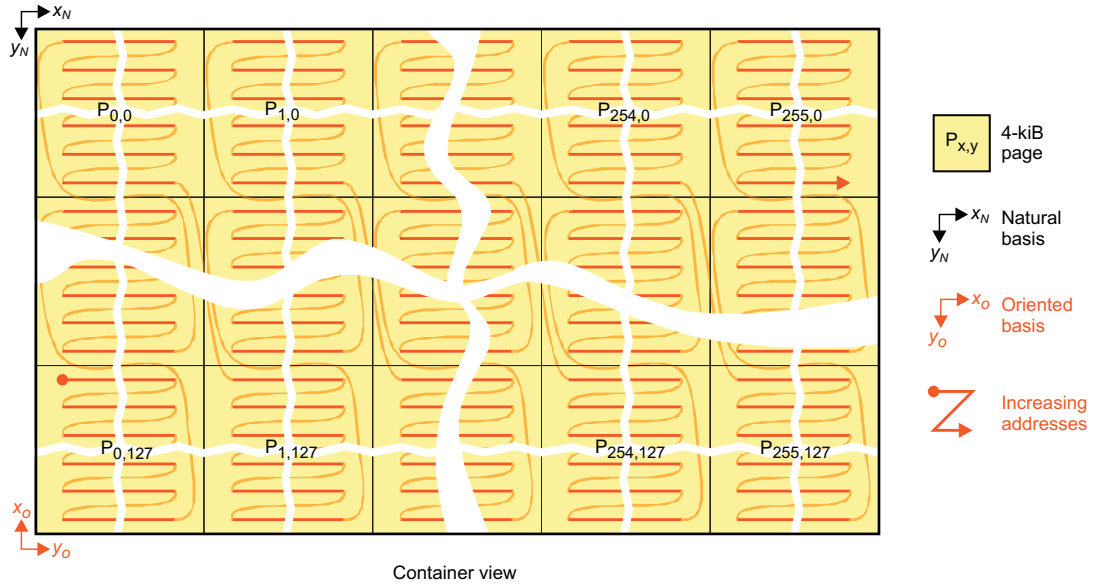
In any TILER mode, the elements are then ordered from bottom-to-top and then from left-to-right in their container, as shown in Figure 6-43 and Figure 6-44.

**Figure 6-43. Tiled Mode Ordering of Elements in 90° View**




Container view

**Figure 6-44. Page Mode Ordering of Elements in 90° View**




Container view


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.4.8 90º View with Horizontal Mirror or 270º View with Vertical Mirror (Orientation 7)

This orientation defined by S = 1, Y = 1, and X = 1 and means that the operated change of basis is:

X0 = -Yn

Y0 = -Xn

In any TILER mode, the elements are then ordered from bottom-to-top and then from right-to-left in their container, as shown in Figure 6-45 and Figure 6-46.

**Figure 6-45. Tiled Mode Ordering of Elements in 90º View with Horizontal Mirror**

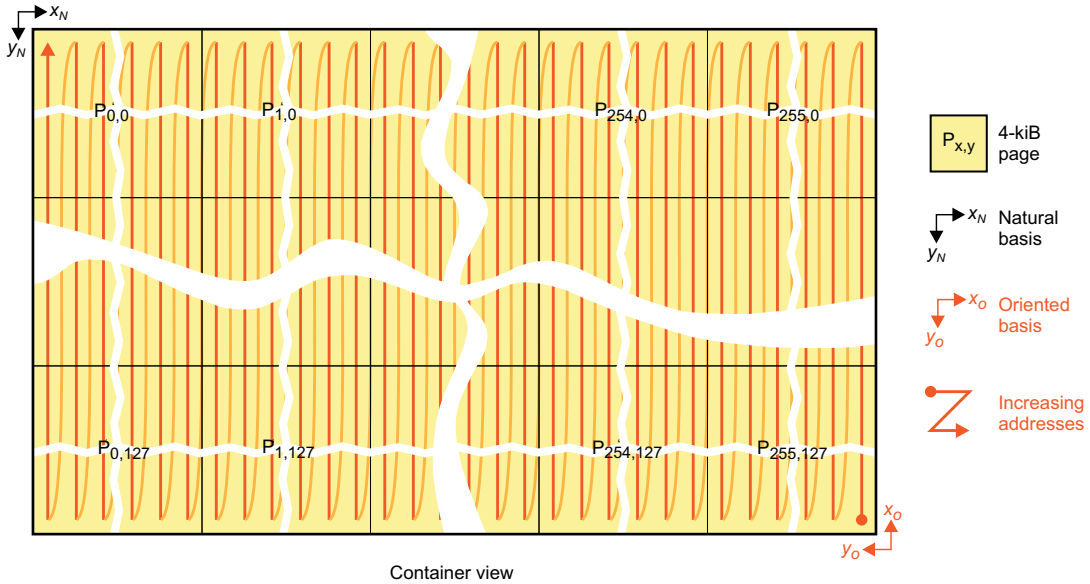
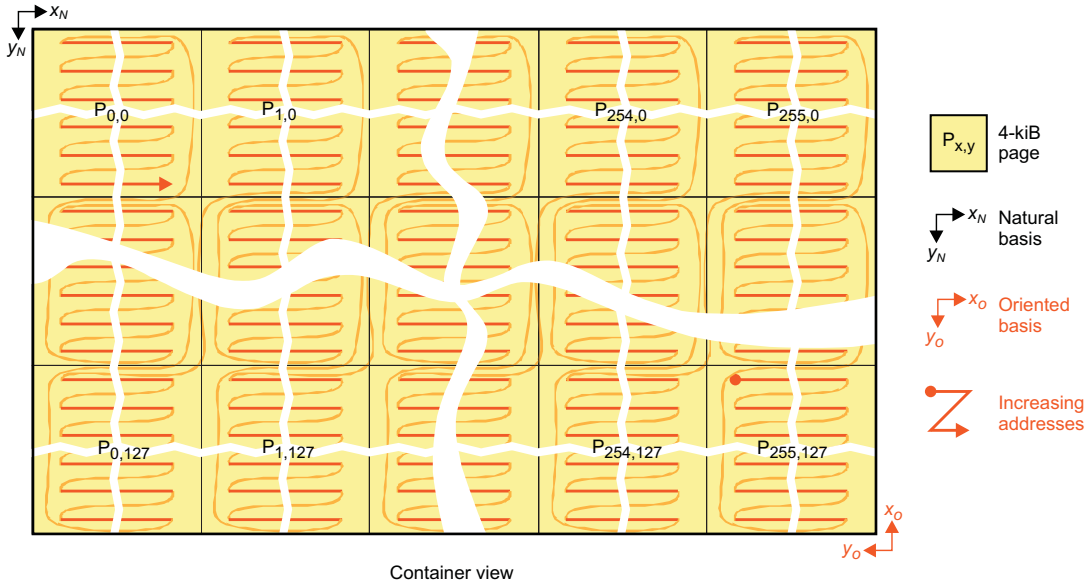



**Figure 6-46. Page Mode Ordering of Elements in 90º View with Horizontal Mirror**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.2.2.8.5 Tiled and Rotational Efficiency

Because of the distribution of the image data in both direction and co-locating them, using sub-tiling, the Image processing algorithms running on the HDVICPs are more efficient by making fewer accesses to the SDRAM, to fetch Macro-blocks of data.

In the Device, the DMA engine of HD_VPSS, also known as VPDMA, has internal line buffers of its clients. For every tiled access, it is ensured that rastering tiled data has zero overhead, when processing an entire frame, or $2^{2n}$ number of lines.

## 6.3 Use Case

### 6.3.1 DMM Basic Register Setup

This section describes basic steps to initialize the DMM, in the device.

1. Set base address, DMM_PAT_VIEW_MAP_BASE to value 8000 0000h, which corresponds to base SDRAM address in the device memory map.
2. Program the four PAT views, by writing DMM_PAT_VIEW_MAP__0..3 registers. By default, all the four PAT views are having value of 0, which implies Direct access translation for all the four tiled modes of (8,16,32-bit and paged mode) and all the four containers will overlap with their base address mapped to 8000 0000h.
3. Program PAT views for all initiators, indexed by ConnID, by programming the DMM_PAT_VIEW__0..1 registers. There are 4 possible views available in device. Default is for all initiators to use PAT view 0.
4. Program PEG priority for all the initiators. By default - all are having priority = 0.
5. Program the TILER orientation for all the initiators, by writing to DMM_TILER_OR__0..1 registers. By default - all will access tiled data in Orientation = 0, which is the normal view.
6. Program Section mapping, based on the SDRAM configuration of the system, which depends on the size of the SDRAM on one or both the banks.
   - Example 1: Symmetrical DDRs on both EMIF banks. Each size = 512MB to add up to 1GB of total SDRAM memory. In all DMM_LISA_MAP__0..3 registers, program value 8064 0300h, which implies 1GB section starting from System address : 8000 0000h, mapped to 0000 0000h of both EMIF0 and EMIF1, with interleaving of 128 bytes enabled. Note that the above example uses only one DMM section and the three other are identical. There are other ways to define up-to 4 sections and configure each differently

#### 6.3.1.1 PAT Direct Access Translation with Distinct 128 MB Containers for Each of the 4 Modes

Following is a simplistic use case, bypassing the LUTs for address translation and setting aside a chunk of 128MB contiguous memory, with base address 128MB aligned, for each of the 4 modes (8, 16,32-bit and paged mode). The same addresses will be used by all initiators in the system.

Configure PAT view 0:

Program DMM_PAT_VIEW_MAP__0 to a value of 0302 0100h. Here : 8-bit mode container will be mapped to 8000 0000h. 16-bit mode container will be mapped to 8800 0000h. 32-bit mode container will be mapped to 9000 0000h. Paged mode container will be mapped to 9800 0000h.

Configure all initiators to use PAT view 0:

Program DMM_PAT_VIEW__0..1 to a value of 0.


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.3.2 Simple LUT Bypass Use Case: Arrangement of Video Buffers

This section describes one possible arrangement of H.264 video buffers which are in YUV 420 format. Setup is as follows:

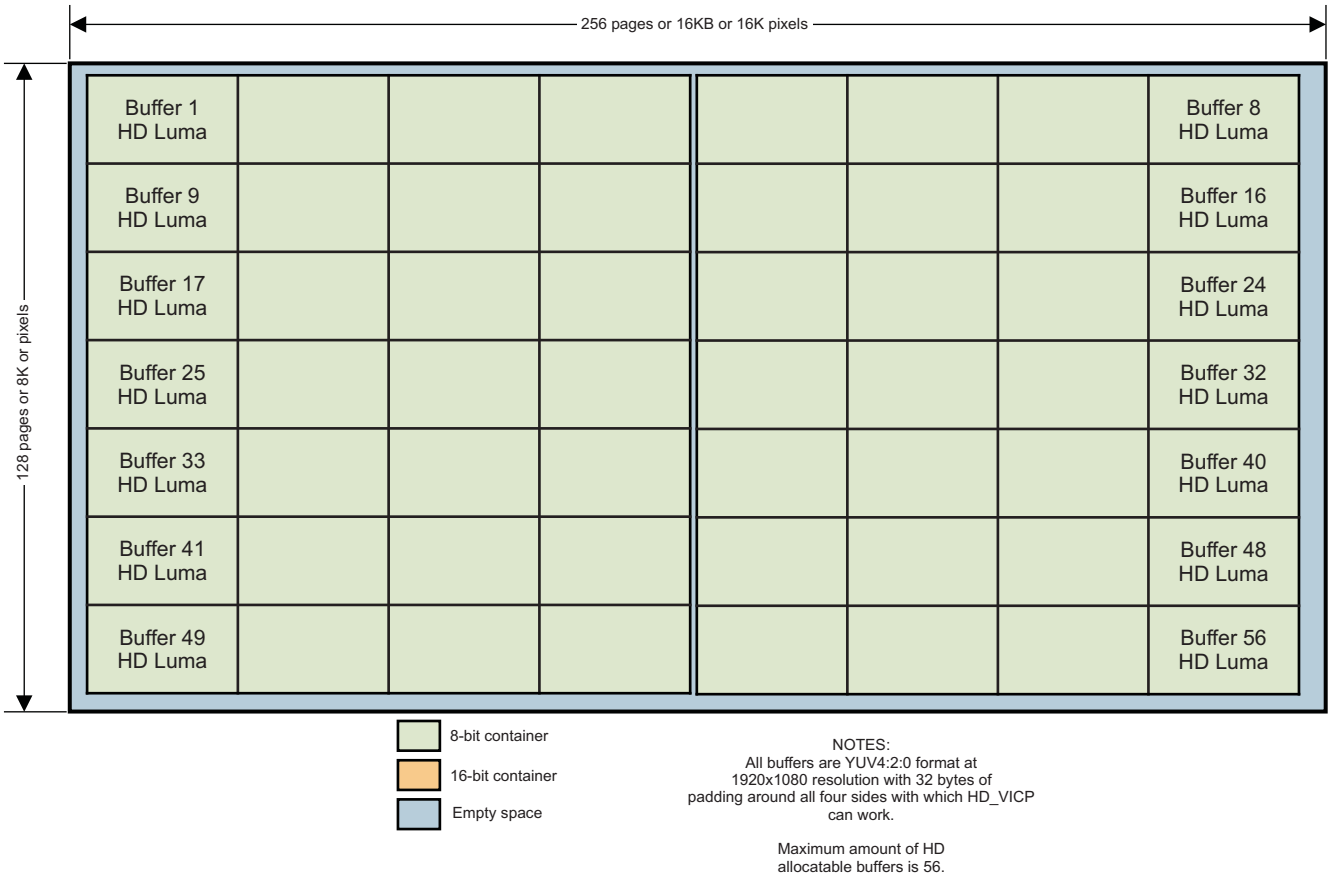
1. The buffers size if 1920 × 1080
2. The Luma buffers are allocated in the 8-bit mode container
3. The Chroma buffers are allocated in the 16-bit mode container
4. The H.264 algorithm needs 32 bytes of padding on each side for Luma buffer and 16 bytes padding in each side for a chroma buffer
5. All buffers are allocated on a 4K aligned address

#### 6.3.2.1 Luma Buffers

In 8-bit tiled mode (Figure 6-47), each 4K page is arranged as 64-pixels wide and 64-pixels high, where each pixel is of size 8-bits. In YUV 420 format, the Luma buffer of the image is 1920 × 1080 and each pixel is 8-bits. Thus, each buffer is 1920 bytes wide. With a pad of 32 bytes on each side, the buffer is (1920 + 64 =) 1984-bytes wide. So one video buffer needs (1984/64 =) 31 pages, along the width of the tiler container. So the tiler container can fit (256/31 ~) 8 buffers, along its width.

Similarly, with a pad of 32 bytes on top and bottom, the buffer is (1080 + 64 =) 1144. Rounding it to (1144 + 8 =) 1152, so that buffer is allocated on page boundary. So one video buffer needs (1152/64 =) 18 pages, along the height of the tiler container. So the tiler container can fit (128/18 ~) 7 buffers, along its height.

**Figure 6-47. Buffer Arrangement for HD Luma Buffers in 128MB 8-bit Mode Container**





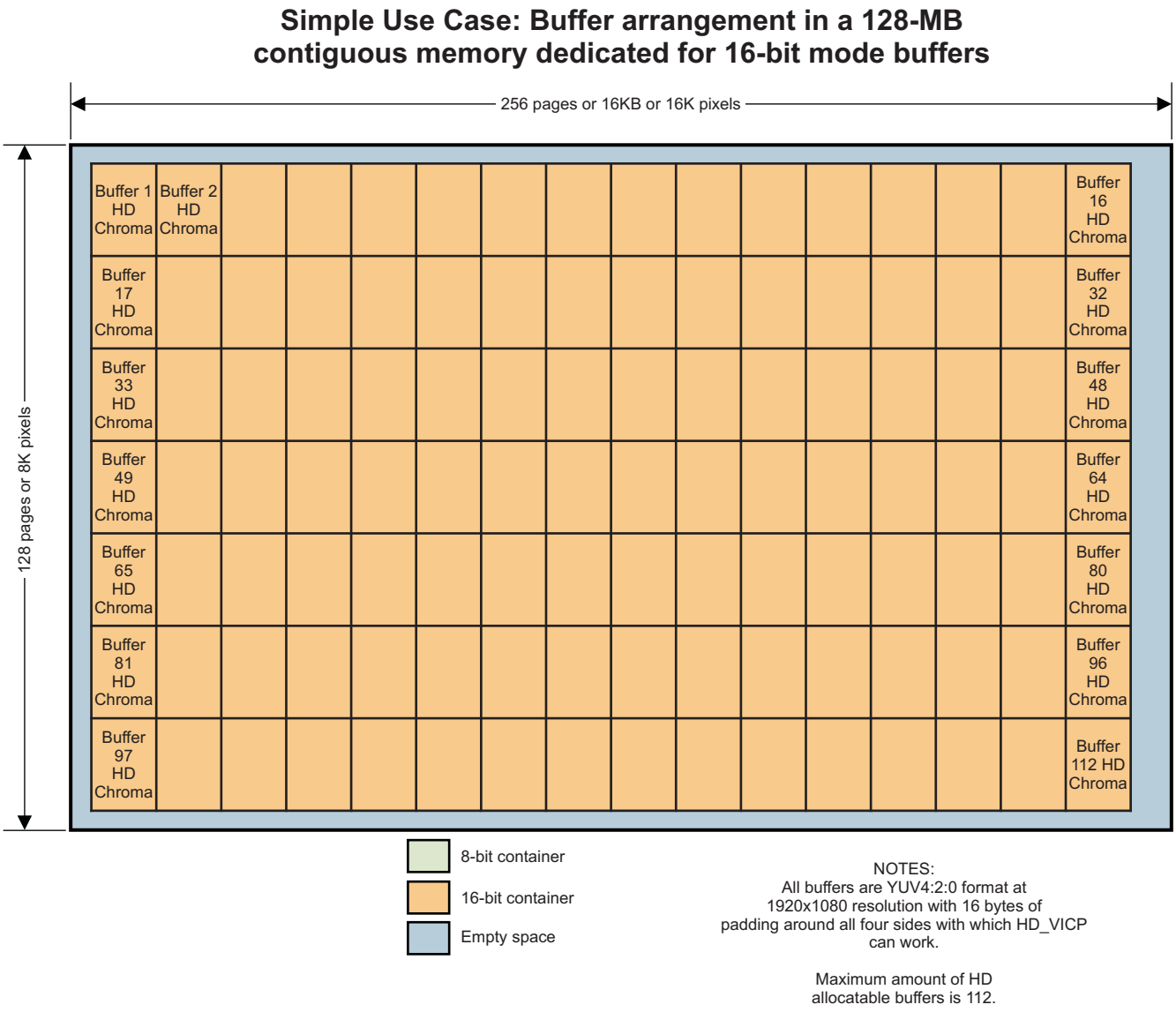
Copyright © 2011–2016, Texas Instruments Incorporated

### 6.3.2.2 Chroma Buffers

In 16-bit tiled mode (Figure 6-48), each 4K page is arranged as 64-pixels wide and only 32-pixels high, where each pixel is of size 16-bits. In YUV 420 format, the Chroma buffer of the image is 960 × 540 and each pixel is 16-bits. Thus, each buffer is 960-bytes wide. With a pad of 16 bytes on each side, the buffer is (960 + 32 =) 992-bytes wide. Rounding it to (992 + 32 =) 1024, so that buffer is allocated on page boundary. So one video buffer needs (1024/64 =) 16 pages, along the width of the tiler container. So the tiler container can fit (256/16 =) 16 buffers, along its width.

Similarly, with a pad of 16 bytes on top and bottom, the buffer is (540 + 32 =) 572. Rounding it to (572 + 4 =) 576, so that buffer is allocated on page boundary. So one video buffer needs (576/32 =) 18 pages, along the height of the tiler container. So the tiler container can fit (128/18 ~) 7 buffers, along its height.

**Figure 6-48. Buffer Arrangement for HD Chroma Buffers in 128MB 16-bit Mode Container**




**NOTE:** The above scheme of Luma and Chroma buffer arrangement can be optimized further, by allocating buffers on sub-tile boundary, instead of page boundary (which is required when using LUT for address translation).


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.3.3 LUT Refill Using the PAT Refill Engines

There is a single LUT of size 256 × 128 elements, in the device.

There four refill engines, along with their own set of registers, to program the LUTs.

Also in this section are described in five different ways to program the LUT.

1. Simple manual area refill
2. Single auto-configured area refill
3. Chained auto-configured area refill
4. Synchronised auto-configured area refill
5. Cyclic synchronised auto-configured area refill

Some of the guidelines, in programming the LUTs are mentioned below:

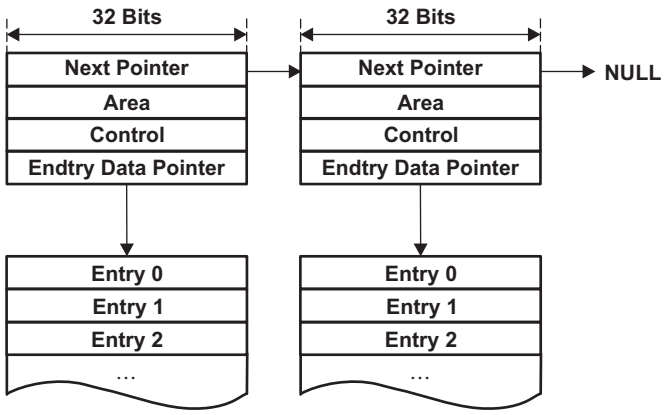
1. Entry table must be in DDR and cannot be split between 2 EMIF banks, when data interleaving enabled.
2. The data table should start from address 16-byte aligned.
3. With one set of descriptor, up to 64 LUT entries can be programmed, in any 2D area dimension.

#### 6.3.3.1 PAT Memory Mapped Refill Descriptors

A PAT descriptor is a singly-linked list node as shown in Figure 6-49, containing:

- a description of the LUT area to reload
- a description of how to reload the defined LUT area
- a pointer to the location where the corresponding LUT entries are stored

**Figure 6-49. PAT Descriptor Node**




The PAT descriptor node and the entry data pointer must both be 16-byte aligned. A typical 'C' language description of this node is as follows:

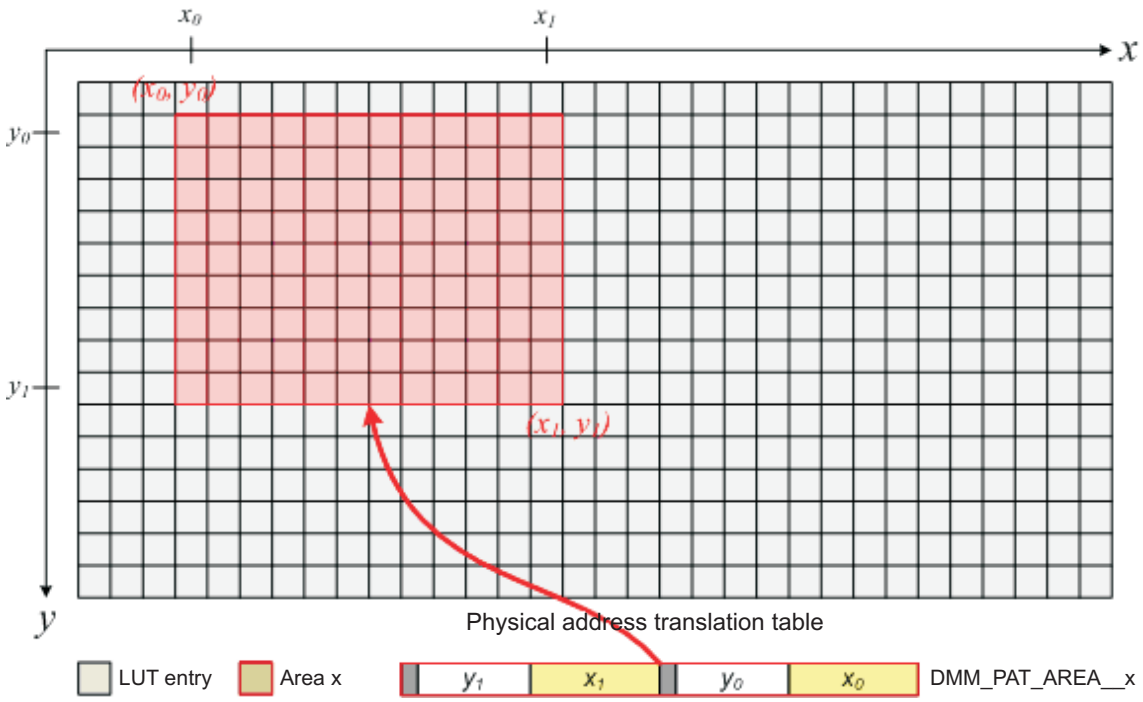
```
struct PAT_desc {
        struct PAT_desc *next;
   u32_t            area;
   u32_t            ctrl;
   PAT_data        *data;
} __attribute__ ((aligned(16)));
```


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.3.3.1.1 PAT Area Refill Description

The area refill description is actually describing the top-left and bottom right corner of the PAT vector table as shown in Figure 6-50.

**Figure 6-50. PAT Area Description**




#### 6.3.3.1.2 PAT Data Description

The pointer to the address in SDRAM, containing the memory based Data table, should be 16 byte aligned. Each field in this table is a 32 bit value, corresponding to the value to be programmed in one LUT entry. Though it is a 32 bit value, only 19 bits [30:12] are relevant. Bit 31 and bits [0:11] are ignored by the PAT refill engine.

#### 6.3.3.1.3 PAT Memory Dump

This section presents how to dump the PAT LUT. This has to be done using the direct access mode of PAT, following the below steps:

1. Set at least one of the refill engine to direct LUT access mode. This has to be done using DMM_PAT_CONFIG register. Only bit 0 to 3 are used for mode of refill engine 0 to 3. (Value 0 is normal mode, value 1 is direct LUT access mode).
2. Set system coordinates we would like to read/write in the LUT. This has to be done using DMM_PAT_AREA__x. x is the refill engine index that has to be coherent with the refill engine used in direct mode (DMM_PAT_CONFIG). Only (x0,y0) is used in direct mode.
3. Select the LUT_ID = 0 using DMM_PAT_CTRL__x register.
4. Read (or write) the data in DMM_PAT_DATA__x register. x is the refill engine index.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.3.3.1.4 PAT Refill Direction

The DMM_PAT_CTRL__x.DIRECTION field gives the “order” of the area refill with the same “SYX” encoding as in the TILER orientations:

- 0: from left to right then from top to bottom
- 1: from right to left then from top to bottom
- 2: from left to right then from bottom to top
- 3: from right to left then from bottom to top
- 4: from top to bottom then from left to right
- 5: from top to bottom then from right to left
- 6: from bottom to top then from left to right
- 7: from bottom to top then from right to left

This feature allows the initiators HD_VPSS and HDVICPs to start an access to an on going refill area. In this case, the direction field is set in accordance with the orientation accesses made by the initiators. The DMM_PAT_STATUS__x.ERROR is asserted to 1 if an access is done to a not yet refilled page.

### 6.3.3.2 Simple Manual Area Refill

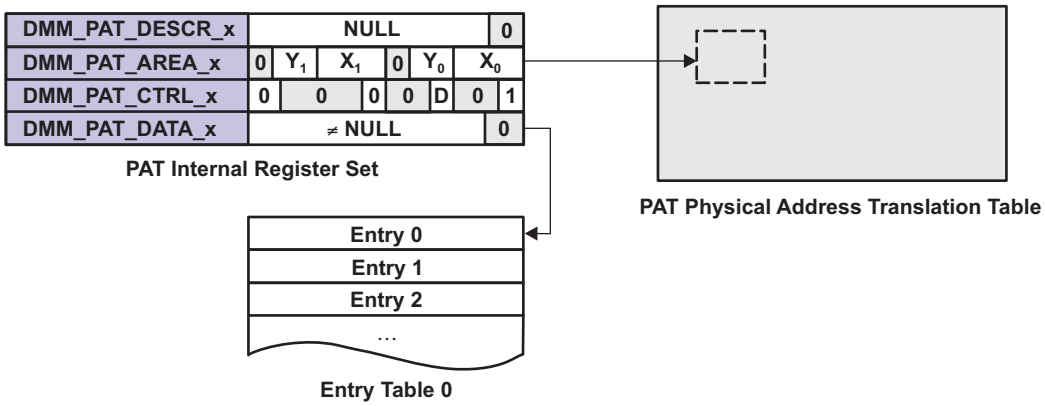
These steps have to be performed to create the 16-byte aligned memory-mapped entry table containing all entries of the defined area, as seen in Figure 6-51.

- Write the DMM_PAT_AREA_i register with the relevant (x0, y0) (x1, y1) area definition
- Write the DMM_PAT_DATA_i register with the physical address of the created entry table
- Write the DMM_PAT_CTRL_i register with the requested refill direction and assert the

DMM_PAT_CTRL_i[0] START bit:

- The refill is done when the DMM_PAT_STATUS_i[3] DONE bit is set.
- A new refill can be initiated when the DMM_PAT_STATUS_i[0] READY bit is set.

**Figure 6-51. DMM Simple Manual Area Refill**





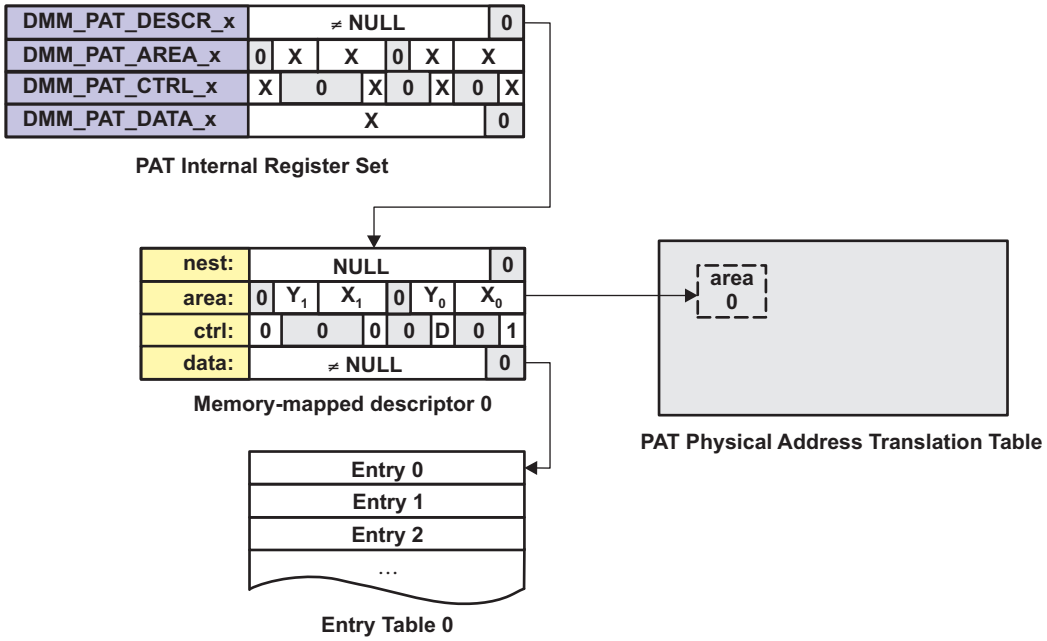
Copyright © 2011–2016, Texas Instruments Incorporated

### 6.3.3.3 Single Auto-Configured Area Refill

These steps have to be performed to create the 16-byte aligned memory-mapped entry table containing all entries of the defined area, as seen in Figure 6-52.

1. Create a 16-byte aligned memory-mapped descriptor structure where:
   (a) the next field is set to NULL
   (b) the area field is set with the relevant (x0, y0) (x1, y1) area definition
   (c) the ctrl field is set with the requested direction D and the start bit asserted to start refilling as soon as this descriptor enters the PAT refill engine
   (d) the data field is set to the physical address of the created entry table
2. The DMM_PAT_DESCR_i register with the physical address of the created descriptor.
3. The refill is done when the DMM_PAT_STATUS_i[3] DONE bit is set.
4. A new refill can be initiated when the DMM_PAT_STATUS_i[0] READY bit is set.

**Figure 6-52. DMM Single Auto-configured Area Refill**





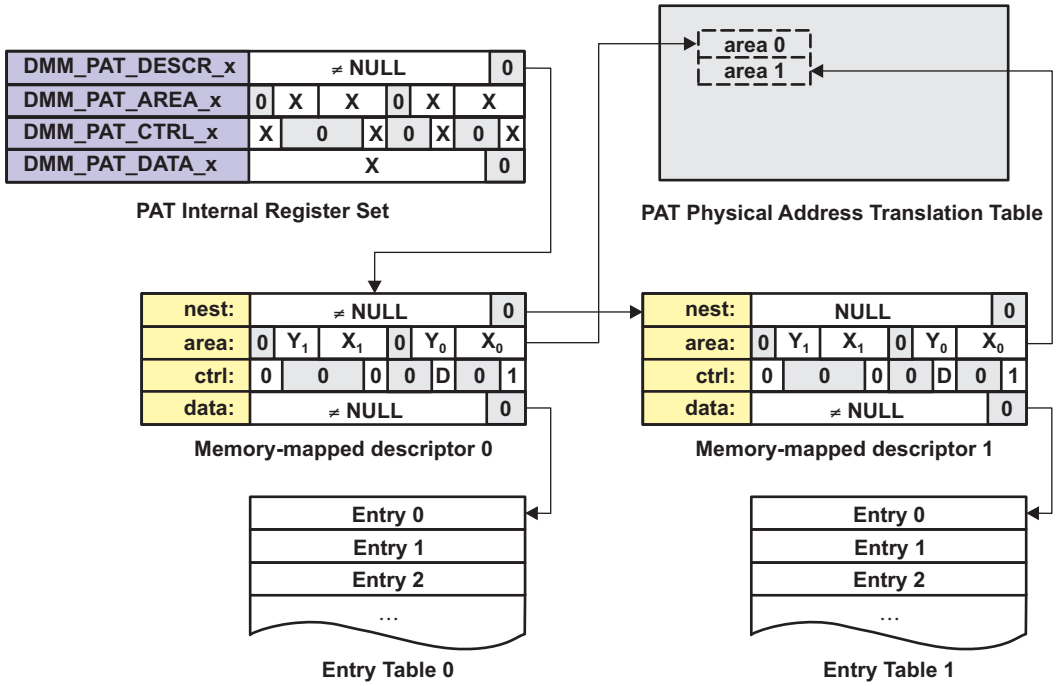
Copyright © 2011–2016, Texas Instruments Incorporated

### 6.3.3.4 Chained Auto-Configured Area Refill

These steps have to be performed to create one 16-byte aligned memory-mapped entry table per area containing the entries for the corresponding area, as seen in Figure 6-53.

1. Create one 16-byte aligned memory-mapped descriptor structures per area where:
   (a) the next field is set to the physical address of the next descriptor or NULL for the last one
   (b) the area field is set with the relevant (x0, y0) (x1, y1) area definition
   (c) the ctrl field is set with the requested direction D and the START bit asserted to start refilling as soon as the previous area refill is done
   (d) the data field is set to the physical address of the corresponding entry table
2. Write the DMM_PAT_DESCR_i register with the physical address of the first created descriptor.
3. Each area refill is done when the DMM_PAT_STATUS_i[3] DONE bit is set.
4. All area refills are done when the DMM_PAT_STATUS_i[0] READY bit is set.
5. A new refill can be initiated when the DMM_PAT_STATUS_i[0] READY bit is set.

**Figure 6-53. DMM Chained Auto-configured Area Refill**





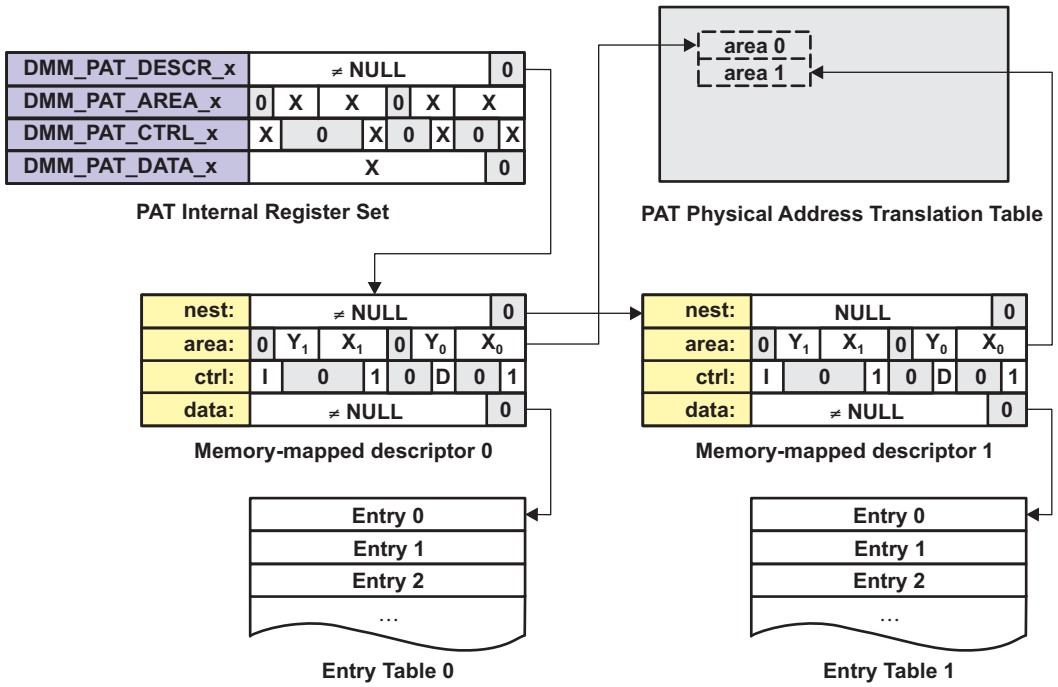
Copyright © 2011–2016, Texas Instruments Incorporated

### 6.3.3.5 Synchronised Auto-configured Area Refill

These steps have to be performed to create one 16-byte aligned memory-mapped entry table per area containing the entries for the corresponding area, as seen in Figure 6-54.

1. Create one 16-byte aligned memory-mapped descriptor structures per area where:
   (a) the next field is set to the physical address of the next descriptor or NULL for the last one
   (b) the area field is set with the relevant (x0, y0) (x1, y1) area definition
   (c) the ctrl field is set with the synchronising initiator identifier I, the SYNC bit asserted, the requested direction D, and the START bit asserted to start refilling as soon as the previous area refill is done and initiator I has made one access in the previous area
   (d) the data field is set to the physical address of the corresponding entry table
2. Write the DMM_PAT_DESCR_i register with the physical address of the first created descriptor.
3. Each area refill is done when the DMM_PAT_STATUS_i[3] DONE bit is set.
4. All area refills are done when the DMM_PAT_STATUS_i[0] READY bit is set.
5. A new refill can be initiated when the DMM_PAT_STATUS_i[0] READY bit is set.

**Figure 6-54. DMM Synchronised Auto-configured Area Refill**





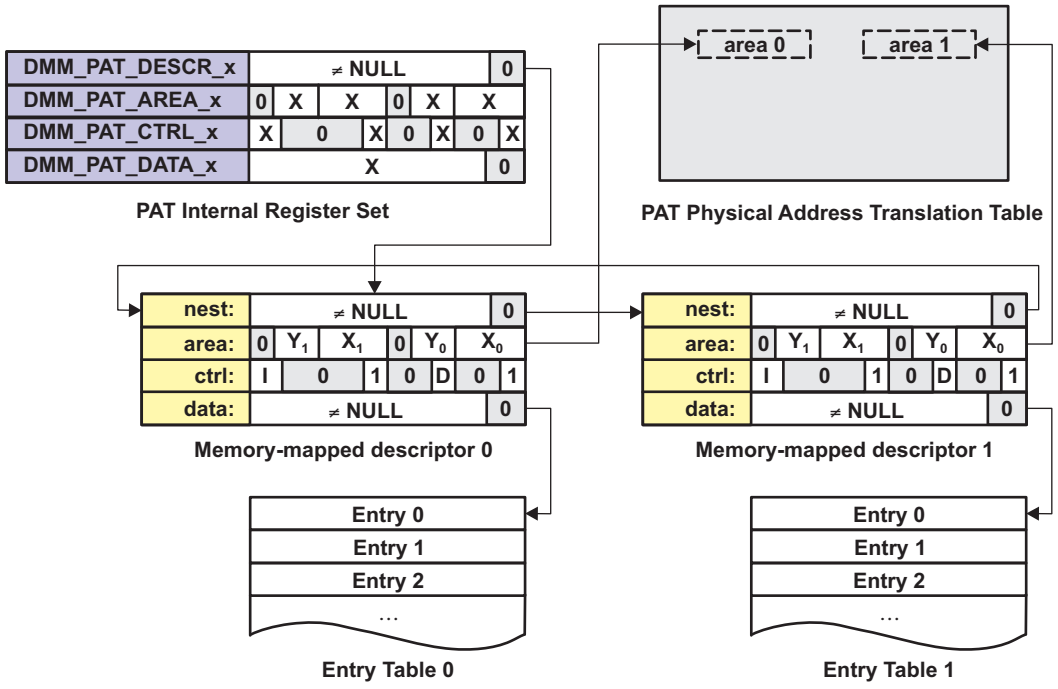
Copyright © 2011–2016, Texas Instruments Incorporated

### 6.3.3.6 Cyclic Snchronised Auto-Configured Area Refill

These steps have to be performed to create one 16-byte aligned memory-mapped entry table per area containing the entries for the corresponding area, as seen in Figure 6-55.

1. Create one 16-byte aligned memory-mapped descriptor structures per area where:
   (a) the next field is set to the physical address of the next descriptor in the circular list
   (b) the area field is set with the relevant (x0, y0) (x1, y1) area definition
   (c) the ctrl field is set with the synchronising initiator identifier I, the SYNC bit asserted, the requested direction D, and the START bit asserted to start refilling as soon as the previous area refill is done and initiator I has made one access in the previous area
   (d) the data field is set to the physical address of the corresponding entry table
2. Write the DMM_PAT_DESCR_i register with the physical address of the initial descriptor.
3. Each area refill is done when the DMM_PAT_STATUS_i[3] DONE bit is set.
4. A new refill can be initiated by writing any value to the DMM_PAT_DESCR_i to abort the current one.

**Figure 6-55. DMM Cyclic Synchronised Auto-configured Area Refill**





Copyright © 2011–2016, Texas Instruments Incorporated

### 6.3.4 Address Management Using LISA Sections

In the device, there are either one ot two memory controllers (EMIFs).

#### 6.3.4.1 DMM Sections

The DMM section definition is certainly the very first step in the DMM software configuration, as it is about defining the system address map to be used when accessing the SDRAM controllers.

A DMM section description fits in a single 32-bit register (DMM_LISA_MAP_x), where:

- SYS_ADDR defines the base address of the decoding range for the section
- SYS_SIZE defines the size of the section. Note that the encoding of this parameter is the number of bits actually used in the upper 8-bits of the incoming system address.
- SDRC_ADDR defines the physical base address of the section in the external memory controller
- SDRC_ADDRSPC defines the address space used on the SDRAM controller when hitting this section. The address space feature is not supported.
- SDRC_MAP defines the target memory controllers for this section. A section may hit a one or two controllers. The section is not used if this parameter is set to zero.
- SDRC_INTL defines the granularity of the interleaving if the section is mapped on more than one memory controllers

A couple of examples are given in the next subsections.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.3.4.1.1 Case 1: Two Memory Controllers, 2GB DDR Symmetrical Distribution

Use Case: In this example we assume 2GB of external memory, spread evenly between 2 SDRAM controllers.

**Table 6-4. Case 1 Memory Controllers**

| System Address Range | Memory Controller | EMIF Address Range |
|---|---|---|
| 8000 0000-FFFF FFFFh | 1 and 2, interleaved at 256 Bytes | 0000 0000h-7FFF FFFFh |

**Configuration:**

**Table 6-5. Controller Configuration**

| Bit | Field | Section 0 |
|---|---|---|
| 31-24 | SYS_ADDR | 80h (incoming System Address MSB) |
| 22-20 | SYS_SIZE | 7h (2 GB) |
| 19-18 | SDRC_INTL | 2h (256 bytes interleaving) |
| 17-16 | SDRC_ADDRSPC | 0 (Unused Reserved field) |
| 8 | SDRC_MAP | 3h (Map to both EMIF0 and EMIF1 |
| 7-0 | SDRC_ADDR | 0 (SDRC address MSB) |

**NOTE:** Section 1, Section 2 and Section 3 - identical to Section 0.

**How DMM decodes and translates:**

1. To check if an address hits a section, use the eight upper address bits of the address and mask them with the hit mask (2 8 - 2SYS_SIZE). If the result is equal to SYS_ADDR, the section is hit.
2. To define the physical address to be issued to the memory controller, use the eight upper address of the system address, mask them with the address mask ($2^{SYS_SIZE} – 1$ ), and then OR them with SDRC_ADDR. This will give the resulting eight upper physical address bits. All lower address bits are forwarded unchanged.
3. If interleaving enabled, the physical address generation is modified. In case of a 256 bytes interleaving, the first chunk of 256 bytes is mapped to the first controller, the second chunk to the second controller, and so on. This results in system address bit 8 to be decoded as the SDRAM controller ID, 0 for the first controller and 1 for the second one (system address bit 7 would be used for interleaving at 128 bytes boundary, and bit 9 for 512 bytes). This bit is not included in the computed physical address, meaning the upper system address bits 9 to 31 are shifted to bits 8 to 30 when generating the physical address.

**Example with above configuration:**

1. Incoming access's System address 99AE 37F0h:
2. Hit mask: $2^8 - 2^7 = 80h$
3. Masked upper address bits: 80h AND 80h = 80h (the SYS_ADDR) , thus hits section 0
4. Address mask: $2^7 - 1 = 7Fh$
5. Masked upper address bits: 99h AND 7Fh = 19h
6. Full masked address: 19AE 37F0h
7. Bit 8 is 1 -> targets the second memory controller
8. Full masked shifted address (suppressing bit 8): 0CD7 1BF0h
9. OR upper physical address bits with SDRC_ADDR: 0CD7 1BF0h
10. Physical address: 0CD7 1BF0h


Copyright © 2011–2016, Texas Instruments Incorporated

This request will be forwarded to address 0CD7 1BF0h, of the second memory controller, that is, EMIF1.

#### 6.3.4.1.2 Case 2: Two Memory Controllers, 1152 MB (1 GB + 128 MB) DDR Symmetrical Distribution

In this example we assume 2GB of external memory, spread evenly between two EMIF controllers.

**Table 6-6. Case 2 Memory Controllers**

| System Address Range | SDRAM Controller | EMIF Address Range |
|---|---|---|
| 8000 0000h-C7FF FFFFh | 1 and 2, interleaved at 256 Bytes | 0000 0000h-47FF FFFFh |

**Configuration:**

**Table 6-7. Controller Configuration**

| Bit | Field | Section 0 | Section 1 |
|---|---|---|---|
| 31-24 | SYS_ADDR | 80h (incoming System Address MSB) | C0h (incoming System Address MSB) |
| 22-20 | SYS_SIZE | 6h (1 GB) | 3h (128 MB) |
| 19-18 | SDRC_INTL | 2h (256 bytes interleaving) | 1 (128 bytes interleaving) |
| 17-16 | SDRC_ADDRSPC | 0 (Unused Reserved field) | 0 (Unused Reserved field) |
| 8 | SDRC_MAP | 3h (Map to both EMIF0 and EMIF1) | 3h (Map to both EMIF0 and EMIF1) |
| 7-0 | SDRC_ADDR | 0 (SDRC address MSB) | 20h (SDRC address MSB) |

**NOTE:** Section 2 and Section 3 are identical to Section 1.

**How DMM decodes and translates:**

Identical to Section 6.3.4.1.1.

See Figure 6-56 for System Address Memory Range and its mapping to EMIFs.

**Example with above configuration:**

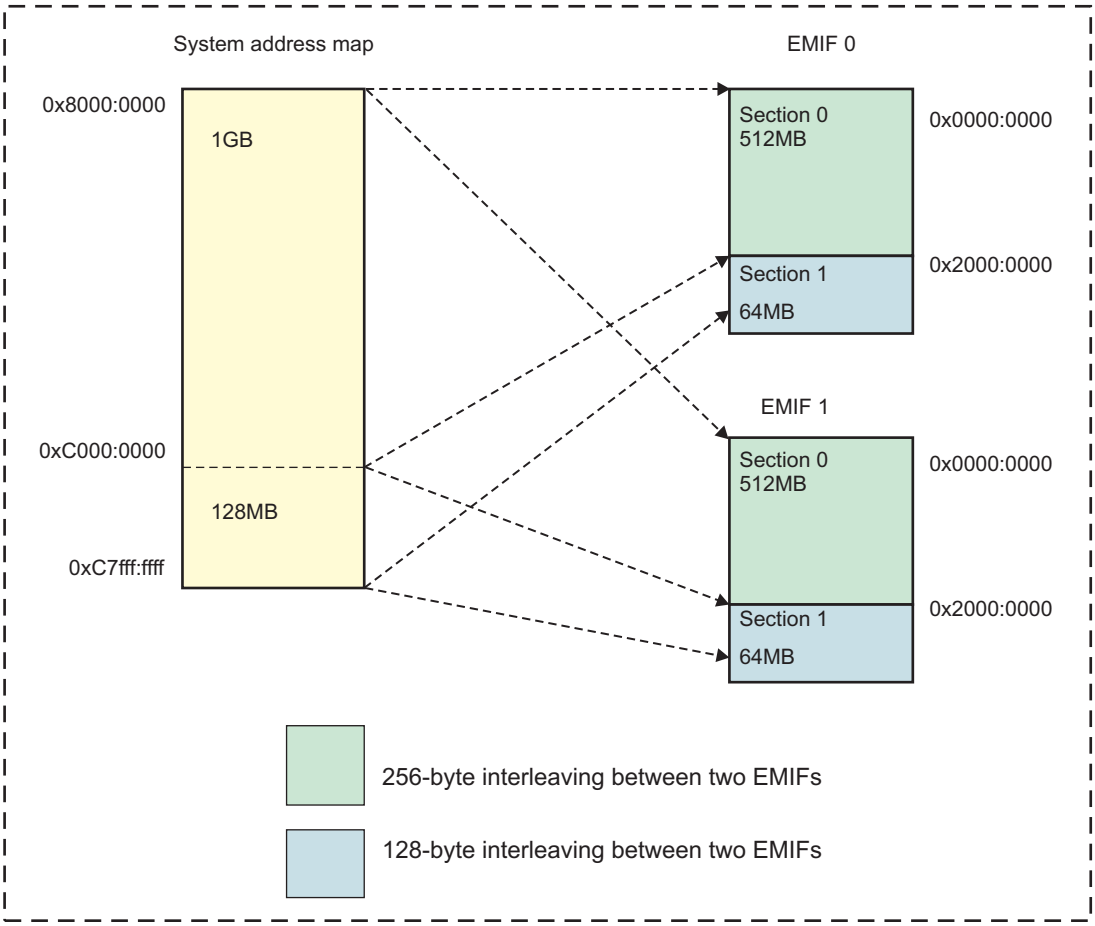
Incoming accesses System address C022 34C0h:

1. Upper address bits: C0h
2. Hit mask: $2^8 - 2^3$ = F8h
3. Masked upper address bits: C0h AND F8h = C0h (the SYS_ADDR) , thus hits section 1
4. Address mask: $2^3 - 1$ = 7h
5. Masked upper address bits: C0h AND 7h = 0
6. Full masked address: 0022 34C0h
7. Bit 7 is 1 -> targets the second memory controller
8. Full masked shifted address (suppressing bit 7): 0011 1A60h
9. OR upper physical address bits with SDRC_ADDR: 0011 1A60h
10. Physical address: 0011 1A60h

This request will be forwarded to address 0011 1A60h, of the second memory controller; that is, EMIF1.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 6-56. DMM Section Use-Case 2**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 6.3.4.1.3 Case 3: Two SDRAM Controllers, 1152 MB (1 GB + 128 MB) DDR Asymmetrical Distribution

**NOTE:** For optimal system performance, symmetric configuration is HIGHLY recommended. Unless dictated by system cost and other constraints, Asymmetrical distribution of memory should not be considered.

There are many ways to configure the section mapping in this use case.

Option 1: Asymmetrical section of EMIF0 is mapped to system address C000 0000h-C7FF FFFFh, higher portion of the SDRAM in system memory map.

**Table 6-8. Section Mapping Option 1**

| Bit | Field | Section 0 | Section 1 |
|---|---|---|---|
| 31-24 | SYS_ADDR | 80h (incoming System Address MSB) | 88h (incoming System Address MSB) |
| 22-20 | SYS_SIZE | 3h (128 MB) | 6h (1 GB) |
| 19-18 | SDRC_INTL | 0 (No interleaving) | 2h (256 bytes interleaving) |
| 17-16 | SDRC_ADDRSPC | 0 (Unused Reserved field) | 0 (Unused Reserved field) |
| 8 | SDRC_MAP | 1 (Map to EMIF0) | 3h (Map to both EMIF0 and EMIF1) |
| 7-0 | SDRC_ADDR | 20h (SDRC address MSB) | 0 (SDRC address MSB) |

Option 2: Asymmetrical section of EMIF0 is mapped to system address 8000 0000h-87FF FFFFh, Lower portion of the SDRAM in system memory map.

**Table 6-9. Section Mapping Option 2**

| Bit | Field | Section 0 | Section 1 |
|---|---|---|---|
| 31-24 | SYS_ADDR | 80h (incoming System Address MSB) | C0h (incoming System Address MSB) |
| 22-20 | SYS_SIZE | 6h (1 GB) | 3h (128 MB) |
| 19-18 | SDRC_INTL | 2h (256 bytes interleaving) | 0 (No interleaving) |
| 17-16 | SDRC_ADDRSPC | 0 (Unused Reserved field) | 0 (Unused Reserved field) |
| 8 | SDRC_MAP | 3h (Map to both EMIF0 and EMIF1) | 1 (Map to EMIF0) |
| 7-0 | SDRC_ADDR | 0 (SDRC address MSB) | 20h (SDRC address MSB) |


Copyright © 2011–2016, Texas Instruments Incorporated

## 6.4 DMM/TILER Registers

Table 6-10 lists the DMM/TILER registers.

**Table 6-10. DMM/TILER Registers**

| Address Offset | Register Name | Section |
|---|---|---|
| 0 | DMM_REVISION | Section 6.4.1 |
| 10h | DMM_SYSCONFIG | Section 6.4.2 |
| 1Ch | DMM_LISA_LOCK | Section 6.4.3 |
| 40h-4Ch | DMM_LISA_MAP_0-3 | Section 6.4.4 |
| 220h-224h | DMM_TILER_OR_0-1 | Section 6.4.5 |
| 410h | DMM_PAT_CONFIG | Section 6.4.6 |
| 420h-424h | DMM_PAT_VIEW_0-3 | Section 6.4.7 |
| 440h-44Ch | DMM_PAT_VIEW_MAP_0-3 | Section 6.4.8 |
| 460h | DMM_PAT_VIEW_MAP_BASE | Section 6.4.9 |
| 478h | DMM_PAT_IRQ_EOI | Section 6.4.10 |
| 480h | DMM_PAT_IRQSTATUS_RAW | Section 6.4.11 |
| 490h | DMM_PAT_IRQSTATUS | Section 6.4.12 |
| 4A0h | DMM_PAT_IRQENABLE_SET | Section 6.4.13 |
| 4B0h | DMM_PAT_IRQENABLE_CLR | Section 6.4.14 |
| 4C0h-4CCh | DMM_PAT_STATUS_0-3 | Section 6.4.15 |
| 500h-530h | DMM_PAT_DESCR_0-3 | Section 6.4.16 |
| 504h-534h | DMM_PAT_AREA_0-3 | Section 6.4.17 |
| 508h-538h | DMM_PAT_CTRL_0-3 | Section 6.4.18 |
| 50Ch-53Ch | DMM_PAT_DATA_0-3 | Section 6.4.19 |
| 620h-63Ch | DMM_PEG_PRIO_0-7 | Section 6.4.20 |
| 640h | DMM_PEG_PRIO_PAT | Section 6.4.21 |

### 6.4.1 DMM Revision Register: DMM_REVISION

The DMM Revision register is shown in Figure 6-57 and described in Table 6-11.

**Figure 6-57. DMM_REVISION Register**

| 31 | 30 | 29 | 28 | 27 | 16 |
|---|---|---|---|---|---|
| SCHEME | | Reserved | | FUNC | |
| R-1 | | R-0 | | R-0 | |

| 15 | 11 | 10 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| RRTL | | XMAJOR | | CUSTOM | | YMINOR | |
| R-0 | | R-0 | | R-0 | | R-8h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-11. DMM_REVISION Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31-30 | SCHEME | 1 | Used to distinguish between old scheme and current. | R |
| 29-28 | Reserved | 0 | Reserved | R |
| 27-16 | FUNC | 0 | Software compatability level | R |
| 15-11 | RRTL | 0 | RTL Version (R) | R |
| 10-8 | XMAJOR | 0 | Major Revision (X) | R |
| 7-6 | CUSTOM | 0 | Special DMM version | R |
| 5-0 | YMINOR | 8h | Minor Revision (Y) | R |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.2 DMM Clock Management Configuration: DMM_SYSCONFIG

The DMM Clock Management Configuration register is shown in Figure 6-58 and described in Table 6-12.

**Figure 6-58. DMM_SYSCONFIG Register**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | IDLEMODE | | Rsvd | |
| R-0 | | R/W-2h | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-12. DMM_SYSCONFIG Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved | R |
| 3-2 | IDLEMODE | | Configuration of the local target state management mode. | R/W |
| | | 0 | Force-idle mode. Backup mode, for debug only. | |
| | | 1h | No idle mode. Backup mode, for debug only. | |
| | | 2h | Smart-idle mode: local target's idle state eventually follows systems IDLE request. | |
| | | 3h | Reserved | |
| 1-0 | Reserved | 0 | Reserved | R |

### 6.4.3 LISA Configuration Locking Register: DMM_LISA_LOCK

The LISA Configuration Locking register is shown in Figure 6-59 and described in Table 6-13.

**Figure 6-59. DMM_LISA_LOCK Register**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | LOCK |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-13. DMM_LISA_LOCK Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved | R |
| 0 | LOCK | | DMM lock map | R/W |
| | | R0 | DMM_LISA_MAP__x un-locked | |
| | | W0 | No effect (clear on reset only) | |
| | | R1 | DMM_LISA_MAP__xlocked | |
| | | W1 | Locking DMM_LISA_MAP__x registers | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.4 DMM LISA MAP Registers: DMM_LISA_MAP_0-DMM_LISA_MAP_3

The DMM LISA MAP register is shown in Figure 6-60 and described in Table 6-14.

**Figure 6-60. DMM_LISA_MAP Registers**

| 31 | 24 | 23 | 22 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| SYS_ADDR | | Rsvd | SYS_SIZE | | SDRC_INTL | | SDRC_ADDRSPC | |
| R/W-0 | | R-0 | R/W-0 | | R/W-0 | | R/W-0 | |

| 15 | 10 | 9 | 8 | 7 | 0 |
|---|---|---|---|---|---|
| Reserved | | SDRC_MAP | | SDRC_ADDR | |
| R-0 | | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-14. DMM_LISA_MAP Registers Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31-24 | SYS_ADDR | 0 | DMM system section address MSB | R/W |
| 23 | Reserved | 0 | Reserved | R |
| 22-20 | SYS_SIZE | | DMM system section size | R/W |
| | | 0h | 16-MB section | |
| | | 1h | 32-MB section | |
| | | 2h | 64-MB section | |
| | | 3h | 128-MB section | |
| | | 4h | 256-MB section | |
| | | 5h | 512-MB section | |
| | | 6h | 1-GB section | |
| | | 7h | 2-GB section | |
| 19-18 | SDRC_INTL | | SDRAM controller interleaving mode | R/W |
| | | 0h | No interleaving | |
| | | 1h | 128-byte interleaving | |
| | | 2h | 256-byte interleaving | |
| | | 3h | 512-byte interleaving | |
| 17-16 | SDRC_ADDRSPC | 0h | Reserved. Should be 0. | R/W |
| 15-10 | Reserved | 0 | Reserved | R |
| 9-8 | SDRC_MAP | | SDRAM controller mapping | R/W |
| | | 0h | Un-mapped | |
| | | 1h | Mapped on SDRC 0 only (not interleaved) | |
| | | 2h | Mapped on SDRC 1 only (not interleaved) | |
| | | 3h | Mapped on SDRC 0 and SDRC 1 (interleaved) | |
| 7-0 | SDRC_ADDR | 0h | SDRAM controller address MSB | R/W |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.5 DMM TILER Orientation Registers: DMM_TILER_OR0-DMM_TILER_OR1

The DMM TILER Orientation register is shown in Figure 6-61 and described in Table 6-15.

**Figure 6-61. DMM_TILER_OR Registers**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| W7 | OR7 | | | W6 | OR6 | | | W5 | OR5 | | | W4 | OR4 | | |
| R/W-0 | R/W-0 | | | R/W-0 | R/W-0 | | | R/W-0 | R/W-0 | | | R/W-0 | R/W-0 | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| W3 | OR3 | | | W2 | OR2 | | | W1 | OR1 | | | W0 | OR0 | | |
| R/W-0 | R/W-0 | | | R/W-0 | R/W-0 | | | R/W-0 | R/W-0 | | | R/W-0 | R/W-0 | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-15. DMM_TILER_OR Registers Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | W7 | 0 | Write-enable for OR7 bit-field; OR7 field is unchanged | R/W |
| 30-28 | OR7 | 0 | Orientation for initiator 8.n+7 | R/W |
| 27 | W6 | 0 | Write-enable for OR6 bit-field; OR6 field is unchanged | R/W |
| 26-24 | OR6 | 0 | Orientation for initiator 8.n+6 | R/W |
| 23 | W5 | 0 | Write-enable for OR5 bit-field; OR5 field is unchanged | R/W |
| 22-20 | OR5 | 0 | Orientation for initiator 8.n+5 | R/W |
| 19 | W4 | 0 | Write-enable for OR4 bit-field; OR4 field is unchanged | R/W |
| 18-16 | OR4 | 0 | Orientation for initiator 8.n+4 | R/W |
| 15 | W3 | 0 | Write-enable for OR3 bit-field; OR3 field is unchanged | R/W |
| 14-12 | OR3 | 0 | Orientation for initiator 8.n+3 | R/W |
| 11 | W2 | 0 | Write-enable for OR2 bit-field; OR2 field is unchanged | R/W |
| 10-8 | OR2 | 0 | Orientation for initiator 8.n+2 | R/W |
| 7 | W1 | 0 | Write-enable for OR1 bit-field; OR1 field is unchanged | R/W |
| 6-4 | OR1 | 0 | Orientation for initiator 8.n+1 | R/W |
| 3 | W0 | 0 | Write-enable for OR0 bit-field; OR0 field is unchanged | R/W |
| 2-0 | OR0 | 0 | Orientation for initiator 8.n+0 | R/W |

### 6.4.6 DMM PAT Configuration Register: DMM_PAT_CONFIG

The DMM PAT Configuration register is shown in Figure 6-62 and described in Table 6-16.

**Figure 6-62. DMM_PAT_CONFIG Register**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | MODE3 | MODE2 | MODE1 | MODE0 |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-16. DMM_PAT_CONFIG Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved | R |
| 3 | MODE3 | 0 | Mode of Refill Engine 3 0 : Normal Mode 1h : Direct LUT access | R/W |
| 2 | MODE2 | 0 | Mode of Refill Engine 1 0 : Normal Mode 1h : Direct LUT access | R/W |
| 1 | MODE1 | 0 | Mode of Refill Engine 2 0 : Normal Mode 1h : Direct LUT access | R/W |
| 0 | MODE0 | 0 | Mode of Refill Engine 0 0 : Normal Mode 1h : Direct LUT access | R/W |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.7 DMM PAT View Registers: DMM_PAT_VIEW_0-DMM_PAT_VIEW_2

The DMM PAT View register is shown in Figure 6-63 and described in Table 6-17.

**Figure 6-63. DMM_PAT_VIEW Registers**

| 31 | 30 | 29 28 | 27 | 26 | 25 24 | 23 | 22 | 21 20 | 19 | 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| W7 | Rsvd | V7 | W6 | Rsvd | V6 | W5 | Rsvd | V5 | W4 | Rsvd | V4 |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 |

| 15 | 14 | 13 12 | 11 | 10 | 9 8 | 7 | 6 | 5 4 | 3 | 2 | 1 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| W3 | Rsvd | V3 | W2 | Rsvd | V2 | W1 | Rsvd | V1 | W0 | Rsvd | V0 |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-17. DMM_PAT_VIEW Registers Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | W7 | 0 | Write-enable for V7 bit-field; V7 field is unchanged | R/W |
| 30 | Reserved | 0 | Reserved | R |
| 29-28 | V7 | 0 | PAT view for initiator 8.n+7 | R/W |
| 27 | W6 | 0 | Write-enable for V6 bit-field; V6 field is unchanged | R/W |
| 26 | Reserved | 0 | Reserved | R |
| 25-24 | V6 | 0 | PAT view for initiator 8.n+6 | R/W |
| 23 | W5 | 0 | Write-enable for V5 bit-field; V5 field is unchanged | R/W |
| 22 | Reserved | 0 | Reserved | R |
| 21-20 | V5 | 0 | PAT view for initiator 8.n+5 | R/W |
| 19 | W4 | 0 | Write-enable for V4 bit-field; V4 field is unchanged | R/W |
| 18 | Reserved | 0 | Reserved | R |
| 17-16 | V4 | 0 | PAT view for initiator 8.n+4 | R/W |
| 15 | W3 | 0 | Write-enable for V3 bit-field; V3 field is unchanged | R/W |
| 14 | Reserved | 0 | Reserved | R |
| 13-12 | V3 | 0 | PAT view for initiator 8.n+3 | R/W |
| 11 | W2 | 0 | Write-enable for V2 bit-field; V2 field is unchanged | R/W |
| 10 | Reserved | 0 | Reserved | R |
| 9-8 | V2 | 0 | PAT view for initiator 8.n+2 | R/W |
| 7 | W1 | 0 | Write-enable for V1 bit-field; V1 field is unchanged | R/W |
| 6 | Reserved | 0 | Reserved | R |
| 5-4 | V1 | 0 | PAT view for initiator 8.n+1 | R/W |
| 3 | W0 | 0 | Write-enable for V0 bit-field; V0 field is unchanged | R/W |
| 2 | Reserved | 0 | Reserved | R |
| 1-0 | V0 | 0 | PAT view for initiator 8.n+0 | R/W |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.8 DMM View Map Registers: DMM_PAT_VIEW_MAP_0-DMM_PAT_VIEW_MAP_4

The DMM View Map register is shown in Figure 6-64 and described in Table 6-18.

**Figure 6-64. DMM_PAT_VIEW_MAP Registers**

| 31 | 30 28 | 27 24 | 23 | 22 20 | 19 16 |
|---|---|---|---|---|---|
| ACCESS_PAGE | Reserved | CONT_PAGE | ACCESS_32 | Reserved | CONT_32 |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 |

| 15 | 14 12 | 11 8 | 7 | 6 4 | 3 0 |
|---|---|---|---|---|---|
| ACCESS_16 | Reserved | CONT_16 | ACCESS_8 | Reserved | CONT_8 |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 6-18. DMM_PAT_VIEW_MAP Registers Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | ACCESS_PAGE | | Kind of access for this page mode container | R/W |
| | | 0 | Direct access, container base address given in CONT_PAGE | |
| | | 1 | Indirect access translation, through the LUT indexed by CONT_PAGE, which should be 0, pointing to the single LUT in the device | |
| 30-28 | Reserved | 0 | Reserved | R |
| 27-24 | CONT_PAGE | 0 | Base address of Container for page mode -or - LUT index | R/W |
| 23 | ACCESS_32 | | Kind of access for this 32-bit mode container | R/W |
| | | 0 | Direct access, container base address given in CONT_32 | |
| | | 1 | Indirect access translation, through the LUT indexed by CONT_32, which should be 0, pointing to the single LUT in the device | |
| 22-20 | Reserved | 0 | Reserved | R |
| 19-16 | CONT_32 | 0 | Base address of Container for 32-bit mode -or - LUT index | R/W |
| 15 | ACCESS_16 | 0 | Kind of access for this 16-bit mode container | R/W |
| | | 0 | Direct access, container base address given in CONT_16 | |
| | | 1 | Indirect access translation, through the LUT indexed by CONT_16, which should be 0, pointing to the single LUT in the device | |
| 14-12 | Reserved | 0 | Reserved | R |
| 11-8 | CONT_16 | 0 | Base address of Container for 16-bit mode -or - LUT index | R/W |
| 7 | ACCESS_8 | | Kind of access for this 8-bit mode container | R/W |
| | | 0 | Direct access, container base address given in CONT_8 | |
| | | 1h | Indirect access translation, through the LUT indexed by CONT_8, which should be 0, pointing to the single LUT in the device | |
| 6-4 | Reserved | 0 | Reserved | R |
| 3-0 | CONT_8 | 0 | Base address of Container for 8-bit mode -or - LUT index | R/W |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.9 DMM PAT View Mapping Base Address Register: DMM_PAT_VIEW_MAP_BASE

The DMM PAT View Mapping Base Address register is shown in Figure 6-65 and described in Table 6-19.

**Figure 6-65. DMM_PAT_VIEW_MAP_BASE Register**

| 31 | 30 | 0 |
|---|---|---|
| BASEADDR | Reserved | |
| R/W-0 | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-19. DMM_PAT_VIEW_MAP_BASE Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | BASEADDR | 0 | MSB of the PAT view mapping base address | R/W |
| 30-0 | Reserved | 0 | Reserved | R |

### 6.4.10 DMM PAT End of Interrupt Register: DMM_PAT_IRQ_EOI

The DMM PAT End of Interrupt register is shown in Figure 6-66 and described in Table 6-20. This register is per-event raw interrupt status vector. The purpose is mostly for debug. Raw status is set even if the related event is not enabled. Write 1 to set the (raw) status.

**Figure 6-66. DMM_PAT_IRQ_EOI Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ERR_LUT_MISS3 | ERR_UPD_DATA3 | ERR_UPD_CTRL3 | ERR_UPD_AREA3 | ERR_INV_DATA3 | ERR_INV_DSC3 | FILL_LST3 | FILL_DSC3 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ERR_LUT_MISS2 | ERR_UPD_DATA2 | ERR_UPD_CTRL2 | ERR_UPD_AREA2 | ERR_INV_DATA2 | ERR_INV_DSC2 | FILL_LST2 | FILL_DSC2 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ERR_LUT_MISS1 | ERR_UPD_DATA1 | ERR_UPD_CTRL1 | ERR_UPD_AREA1 | ERR_INV_DATA1 | ERR_INV_DSC1 | FILL_LST1 | FILL_DSC1 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERR_LUT_MISS0 | ERR_UPD_DATA0 | ERR_UPD_CTRL0 | ERR_UPD_AREA0 | ERR_INV_DATA0 | ERR_INV_DSC0 | FILL_LST0 | FILL_DSC0 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-20. DMM_PAT_IRQ_EOI Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | ERR_LUT_MISS3 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 3 | R/W1S |
| 30 | ERR_UPD_DATA3 | 0 | Data register update whilst refilling error event in area 3 | R/W1S |
| 29 | ERR_UPD_CTRL3 | 0 | Control register update whilst refilling error event in area 3 | R/W1S |
| 28 | ERR_UPD_AREA3 | 0 | Area register update whilst refilling error event in area 3 | R/W1S |
| 27 | ERR_INV_DATA3 | 0 | Invalid entry-table pointer error event in area 3 | R/W1S |
| 26 | ERR_INV_DSC3 | 0 | Invalid descriptor pointer error event in area 3 | R/W1S |
| 25 | FILL_LST3 | 0 | End of refill event for the last descriptor in area 3 | R/W1S |
| 24 | FILL_DSC3 | 0 | End of refill event for any descriptor in area 3 | R/W1S |
| 23 | ERR_LUT_MISS2 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 2 | R/W1S |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 6-20. DMM_PAT_IRQ_EOI Register Field Descriptions (continued)**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 22 | ERR_UPD_DATA2 | 0 | Data register update whilst refilling error event in area 2 | R/W1S |
| 21 | ERR_UPD_CTRL2 | 0 | Control register update whilst refilling error event in area 2 | R/W1S |
| 20 | ERR_UPD_AREA2 | 0 | Area register update whilst refilling error event in area 2 | R/W1S |
| 19 | ERR_INV_DATA2 | 0 | Invalid entry-table pointer error event in area 2 | R/W1S |
| 18 | ERR_INV_DSC2 | 0 | Invalid descriptor pointer error event in area 2 | R/W1S |
| 17 | FILL_LST2 | 0 | End of refill event for the last descriptor in area 2 | R/W1S |
| 16 | FILL_DSC2 | 0 | End of refill event for any descriptor in area 2 | R/W1S |
| 15 | ERR_LUT_MISS1 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 1 | R/W1S |
| 14 | ERR_UPD_DATA1 | 0 | Data register update whilst refilling error event in area 1 | R/W1S |
| 13 | ERR_UPD_CTRL1 | 0 | Control register update whilst refilling error event in area 1 | R/W1S |
| 12 | ERR_UPD_AREA1 | 0 | Area register update whilst refilling error event in area 1 | R/W1S |
| 11 | ERR_INV_DATA1 | 0 | Invalid entry-table pointer error event in area 1 | R/W1S |
| 10 | ERR_INV_DSC1 | 0 | Invalid descriptor pointer error event in area 1 | R/W1S |
| 9 | FILL_LST1 | 0 | End of refill event for the last descriptor in area 1 | R/W1S |
| 8 | FILL_DSC1 | 0 | End of refill event for any descriptor in area 1 | R/W1S |
| 7 | ERR_LUT_MISS0 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 0 | R/W1S |
| 6 | ERR_UPD_DATA0 | 0 | Data register update whilst refilling error event in area 0 | R/W1S |
| 5 | ERR_UPD_CTRL0 | 0 | Control register update whilst refilling error event in area 0 | R/W1S |
| 4 | ERR_UPD_AREA0 | 0 | Area register update whilst refilling error event in area 0 | R/W1S |
| 3 | ERR_INV_DATA0 | 0 | Invalid entry-table pointer error event in area 0 | R/W1S |
| 2 | ERR_INV_DSC0 | 0 | Invalid descriptor pointer error event in area 0 | R/W1S |
| 1 | FILL_LST0 | 0 | End of refill event for the last descriptor in area 0 | R/W1S |
| 0 | FILL_DSC0 | 0 | End of refill event for any descriptor in area 0 | R/W1S |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.11 DMM PAT Raw Interrupt Status Register: DMM_PAT_IRQSTATUS_RAW

The DMM PAT Raw Interrupt Status Register is shown in Figure 6-67 and described in Table 6-21. Per-event raw interrupt status vector. Raw status is set even if the related event is not enabled. Write 1 to set the (raw) status, mostly for debug.

**Figure 6-67. DMM_PAT_IRQSTATUS_RAW Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ERR_LUT_MISS3 | ERR_UPD_DATA3 | ERR_UPD_CTRL3 | ERR_UPD_AREA3 | ERR_INV_DATA3 | ERR_INV_DSC3 | FILL_LST3 | FILL_DSC3 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ERR_LUT_MISS2 | ERR_UPD_DATA2 | ERR_UPD_CTRL2 | ERR_UPD_AREA2 | ERR_INV_DATA2 | ERR_INV_DSC2 | FILL_LST2 | FILL_DSC2 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ERR_LUT_MISS1 | ERR_UPD_DATA1 | ERR_UPD_CTRL1 | ERR_UPD_AREA1 | ERR_INV_DATA1 | ERR_INV_DSC1 | FILL_LST1 | FILL_DSC1 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERR_LUT_MISS0 | ERR_UPD_DATA0 | ERR_UPD_CTRL0 | ERR_UPD_AREA0 | ERR_INV_DATA0 | ERR_INV_DSC0 | FILL_LST0 | FILL_DSC0 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-21. DMM_PAT_IRQSTATUS_RAW Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | ERR_LUT_MISS3 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 3 | R/W1S |
| 30 | ERR_UPD_DATA3 | 0 | Data register update whilst refilling error event in area 3 | R/W1S |
| 29 | ERR_UPD_CTRL3 | 0 | Control register update whilst refilling error event in area 3 | R/W1S |
| 28 | ERR_UPD_AREA3 | 0 | Area register update whilst refilling error event in area 3 | R/W1S |
| 27 | ERR_INV_DATA3 | 0 | Invalid entry-table pointer error event in area 3 | R/W1S |
| 26 | ERR_INV_DSC3 | 0 | Invalid descriptor pointer error event in area 3 | R/W1S |
| 25 | FILL_LST3 | 0 | End of refill event for the last descriptor in area 3 | R/W1S |
| 24 | FILL_DSC3 | 0 | End of refill event for any descriptor in area 3 | R/W1S |
| 23 | ERR_LUT_MISS2 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 2 | R/W1S |
| 22 | ERR_UPD_DATA2 | 0 | Data register update whilst refilling error event in area 2 | R/W1S |
| 21 | ERR_UPD_CTRL2 | 0 | Control register update whilst refilling error event in area 2 | R/W1S |
| 20 | ERR_UPD_AREA2 | 0 | Area register update whilst refilling error event in area 2 | R/W1S |
| 19 | ERR_INV_DATA2 | 0 | Invalid entry-table pointer error event in area 2 | R/W1S |
| 18 | ERR_INV_DSC2 | 0 | Invalid descriptor pointer error event in area 2 | R/W1S |
| 17 | FILL_LST2 | 0 | End of refill event for the last descriptor in area 2 | R/W1S |
| 16 | FILL_DSC2 | 0 | End of refill event for any descriptor in area 2 | R/W1S |
| 15 | ERR_LUT_MISS1 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 1 | R/W1S |
| 14 | ERR_UPD_DATA1 | 0 | Data register update whilst refilling error event in area 1 | R/W1S |
| 13 | ERR_UPD_CTRL1 | 0 | Control register update whilst refilling error event in area 1 | R/W1S |
| 12 | ERR_UPD_AREA1 | 0 | Area register update whilst refilling error event in area 1 | R/W1S |
| 11 | ERR_INV_DATA1 | 0 | Invalid entry-table pointer error event in area 1 | R/W1S |
| 10 | ERR_INV_DSC1 | 0 | Invalid descriptor pointer error event in area 1 | R/W1S |
| 9 | FILL_LST1 | 0 | End of refill event for the last descriptor in area 1 | R/W1S |
| 8 | FILL_DSC1 | 0 | End of refill event for any descriptor in area 1 | R/W1S |
| 7 | ERR_LUT_MISS0 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 0 | R/W1S |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 6-21. DMM_PAT_IRQSTATUS_RAW Register Field Descriptions (continued)**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 6 | ERR_UPD_DATA0 | 0 | Data register update whilst refilling error event in area 0 | R/W1S |
| 5 | ERR_UPD_CTRL0 | 0 | Control register update whilst refilling error event in area 0 | R/W1S |
| 4 | ERR_UPD_AREA0 | 0 | Area register update whilst refilling error event in area 0 | R/W1S |
| 3 | ERR_INV_DATA0 | 0 | Invalid entry-table pointer error event in area 0 | R/W1S |
| 2 | ERR_INV_DSC0 | 0 | Invalid descriptor pointer error event in area 0 | R/W1S |
| 1 | FILL_LST0 | 0 | End of refill event for the last descriptor in area 0 | R/W1S |
| 0 | FILL_DSC0 | 0 | End of refill event for any descriptor in area 0 | R/W1S |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.12 DMM PAT Interrupt Status Register: DMM_PAT_IRQSTATUS

The DMM PAT Interrupt Status register is shown in Figure 6-68 and described in Table 6-22. Per-event "enabled" interrupt status vector. Error status is not set unless the event is enabled.

**Figure 6-68. DMM_PAT_IRQSTATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ERR_LUT_MISS3 | ERR_UPD_DATA3 | ERR_UPD_CTRL3 | ERR_UPD_AREA3 | ERR_INV_DATA3 | ERR_INV_DSC3 | FILL_LST3 | FILL_DSC3 |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ERR_LUT_MISS2 | ERR_UPD_DATA2 | ERR_UPD_CTRL2 | ERR_UPD_AREA2 | ERR_INV_DATA2 | ERR_INV_DSC2 | FILL_LST2 | FILL_DSC2 |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ERR_LUT_MISS1 | ERR_UPD_DATA1 | ERR_UPD_CTRL1 | ERR_UPD_AREA1 | ERR_INV_DATA1 | ERR_INV_DSC1 | FILL_LST1 | FILL_DSC1 |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERR_LUT_MISS0 | ERR_UPD_DATA0 | ERR_UPD_CTRL0 | ERR_UPD_AREA0 | ERR_INV_DATA0 | ERR_INV_DSC0 | FILL_LST0 | FILL_DSC0 |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 6-22. DMM_PAT_IRQSTATUS Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | ERR_LUT_MISS3 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 3 | R/W1C |
| 30 | ERR_UPD_DATA3 | 0 | Data register update whilst refilling error event in area 3 | R/W1C |
| 29 | ERR_UPD_CTRL3 | 0 | Control register update whilst refilling error event in area 3 | R/W1C |
| 28 | ERR_UPD_AREA3 | 0 | Area register update whilst refilling error event in area 3 | R/W1C |
| 27 | ERR_INV_DATA3 | 0 | Invalid entry-table pointer error event in area 3 | R/W1C |
| 26 | ERR_INV_DSC3 | 0 | Invalid descriptor pointer error event in area 3 | R/W1C |
| 25 | FILL_LST3 | 0 | End of refill event for the last descriptor in area 3 | R/W1C |
| 24 | FILL_DSC3 | 0 | End of refill event for any descriptor in area 3 | R/W1C |
| 23 | ERR_LUT_MISS2 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 2 | R/W1C |
| 22 | ERR_UPD_DATA2 | 0 | Data register update whilst refilling error event in area 2 | R/W1C |
| 21 | ERR_UPD_CTRL2 | 0 | Control register update whilst refilling error event in area 2 | R/W1C |
| 20 | ERR_UPD_AREA2 | 0 | Area register update whilst refilling error event in area 2 | R/W1C |
| 19 | ERR_INV_DATA2 | 0 | Invalid entry-table pointer error event in area 2 | R/W1C |
| 18 | ERR_INV_DSC2 | 0 | Invalid descriptor pointer error event in area 2 | R/W1C |
| 17 | FILL_LST2 | 0 | End of refill event for the last descriptor in area 2 | R/W1C |
| 16 | FILL_DSC2 | 0 | End of refill event for any descriptor in area 2 | R/W1C |
| 15 | ERR_LUT_MISS1 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 1 | R/W1C |
| 14 | ERR_UPD_DATA1 | 0 | Data register update whilst refilling error event in area 1 | R/W1C |
| 13 | ERR_UPD_CTRL1 | 0 | Control register update whilst refilling error event in area 1 | R/W1C |
| 12 | ERR_UPD_AREA1 | 0 | Area register update whilst refilling error event in area 1 | R/W1C |
| 11 | ERR_INV_DATA1 | 0 | Invalid entry-table pointer error event in area 1 | R/W1C |
| 10 | ERR_INV_DSC1 | 0 | Invalid descriptor pointer error event in area 1 | R/W1C |
| 9 | FILL_LST1 | 0 | End of refill event for the last descriptor in area 1 | R/W1C |
| 8 | FILL_DSC1 | 0 | End of refill event for any descriptor in area 1 | R/W1C |
| 7 | ERR_LUT_MISS0 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 0 | R/W1C |
| 6 | ERR_UPD_DATA0 | 0 | Data register update whilst refilling error event in area 0 | R/W1C |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 6-22. DMM_PAT_IRQSTATUS Register Field Descriptions (continued)**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 5 | ERR_UPD_CTRL0 | 0 | Control register update whilst refilling error event in area 0 | R/W1C |
| 4 | ERR_UPD_AREA0 | 0 | Area register update whilst refilling error event in area 0 | R/W1C |
| 3 | ERR_INV_DATA0 | 0 | Invalid entry-table pointer error event in area 0 | R/W1C |
| 2 | ERR_INV_DSC0 | 0 | Invalid descriptor pointer error event in area 0 | R/W1C |
| 1 | FILL_LST0 | 0 | End of refill event for the last descriptor in area 0 | R/W1C |
| 0 | FILL_DSC0 | 0 | End of refill event for any descriptor in area 0 | R/W1C |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.13 DMM PAT Interrupt Enable Register: DMM_PAT_IRQENABLE_SET

The DMM PAT Interrupt Enable register is shown in Figure 6-69 and described in Table 6-23. Per-event interrupt enable bit vector. Write 1 to set (enable interrupt).

**Figure 6-69. DMM_PAT_IRQENABLE_SET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ERR_LUT_MISS3 | ERR_UPD_DATA3 | ERR_UPD_CTRL3 | ERR_UPD_AREA3 | ERR_INV_DATA3 | ERR_INV_DSC3 | FILL_LST3 | FILL_DSC3 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ERR_LUT_MISS2 | ERR_UPD_DATA2 | ERR_UPD_CTRL2 | ERR_UPD_AREA2 | ERR_INV_DATA2 | ERR_INV_DSC2 | FILL_LST2 | FILL_DSC2 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ERR_LUT_MISS1 | ERR_UPD_DATA1 | ERR_UPD_CTRL1 | ERR_UPD_AREA1 | ERR_INV_DATA1 | ERR_INV DSC1 | FILL_LST1 | FILL_DSC1 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERR_LUT_MISS0 | ERR_UPD_DATA0 | ERR_UPD_CTRL0 | ERR_UPD_AREA0 | ERR_INV_DATA0 | ERR_INV_DSC0 | FILL_LST0 | FILL_DSC0 |
| R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-23. DMM_PAT_IRQENABLE_SET Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | ERR_LUT_MISS3 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 3 | R/W1S |
| 30 | ERR_UPD_DATA3 | 0 | Data register update whilst refilling error event in area 3 | R/W1S |
| 29 | ERR_UPD_CTRL3 | 0 | Control register update whilst refilling error event in area 3 | R/W1S |
| 28 | ERR_UPD_AREA3 | 0 | Area register update whilst refilling error event in area 3 | R/W1S |
| 27 | ERR_INV_DATA3 | 0 | Invalid entry-table pointer error event in area 3 | R/W1S |
| 26 | ERR_INV_DSC3 | 0 | Invalid descriptor pointer error event in area 3 | R/W1S |
| 25 | FILL_LST3 | 0 | End of refill event for the last descriptor in area 3 | R/W1S |
| 24 | FILL_DSC3 | 0 | End of refill event for any descriptor in area 3 | R/W1S |
| 23 | ERR_LUT_MISS2 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 2 | R/W1S |
| 22 | ERR_UPD_DATA2 | 0 | Data register update whilst refilling error event in area 2 | R/W1S |
| 21 | ERR_UPD_CTRL2 | 0 | Control register update whilst refilling error event in area 2 | R/W1S |
| 20 | ERR_UPD_AREA2 | 0 | Area register update whilst refilling error event in area 2 | R/W1S |
| 19 | ERR_INV_DATA2 | 0 | Invalid entry-table pointer error event in area 2 | R/W1S |
| 18 | ERR_INV_DSC2 | 0 | Invalid descriptor pointer error event in area 2 | R/W1S |
| 17 | FILL_LST2 | 0 | End of refill event for the last descriptor in area 2 | R/W1S |
| 16 | FILL_DSC2 | 0 | End of refill event for any descriptor in area 2 | R/W1S |
| 15 | ERR_LUT_MISS1 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 1 | R/W1S |
| 14 | ERR_UPD_DATA1 | 0 | Data register update whilst refilling error event in area 1 | R/W1S |
| 13 | ERR_UPD_CTRL1 | 0 | Control register update whilst refilling error event in area 1 | R/W1S |
| 12 | ERR_UPD_AREA1 | 0 | Area register update whilst refilling error event in area 1 | R/W1S |
| 11 | ERR_INV_DATA1 | 0 | Invalid entry-table pointer error event in area 1 | R/W1S |
| 10 | ERR_INV_DSC1 | 0 | Invalid descriptor pointer error event in area 1 | R/W1S |
| 9 | FILL_LST1 | 0 | End of refill event for the last descriptor in area 1 | R/W1S |
| 8 | FILL_DSC1 | 0 | End of refill event for any descriptor in area 1 | R/W1S |
| 7 | ERR_LUT_MISS0 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 0 | R/W1S |
| 6 | ERR_UPD_DATA0 | 0 | Data register update whilst refilling error event in area 0 | R/W1S |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 6-23. DMM_PAT_IRQENABLE_SET Register Field Descriptions (continued)**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 5 | ERR_UPD_CTRL0 | 0 | Control register update whilst refilling error event in area 0 | R/W1S |
| 4 | ERR_UPD_AREA0 | 0 | Area register update whilst refilling error event in area 0 | R/W1S |
| 3 | ERR_INV_DATA0 | 0 | Invalid entry-table pointer error event in area 0 | R/W1S |
| 2 | ERR_INV_DSC0 | 0 | Invalid descriptor pointer error event in area 0 | R/W1S |
| 1 | FILL_LST0 | 0 | End of refill event for the last descriptor in area 0 | R/W1S |
| 0 | FILL_DSC0 | 0 | End of refill event for any descriptor in area 0 | R/W1S |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.14 DMM PAT Interrupt Disable Register: DMM_PAT_IRQENABLE_CLR

The DMM PAT Interrupt Disable register is shown in Figure 6-70 and described in Table 6-24. Per-event interrupt enable bit vector. Write 1 to clear (disable interrupt).

**Figure 6-70. DMM_PAT_IRQENABLE_CLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ERR_LUT_MISS3 | ERR_UPD_DATA3 | ERR_UPD_CTRL3 | ERR_UPD_AREA3 | ERR_INV_DATA3 | ERR_INV_DSC3 | FILL_LST3 | FILL_DSC3 |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ERR_LUT_MISS2 | ERR_UPD_DATA2 | ERR_UPD_CTRL2 | ERR_UPD_AREA2 | ERR_INV_DATA2 | ERR_INV_DSC2 | FILL_LST2 | FILL_DSC2 |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ERR_LUT_MISS1 | ERR_UPD_DATA1 | ERR_UPD_CTRL1 | ERR_UPD_AREA1 | ERR_INV_DATA1 | ERR_INV_DSC1 | FILL_LST1 | FILL_DSC1 |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERR_LUT_MISS0 | ERR_UPD_DATA0 | ERR_UPD_CTRL0 | ERR_UPD_AREA0 | ERR_INV_DATA0 | ERR_INV_DSC0 | FILL_LST0 | FILL_DSC0 |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 6-24. DMM_PAT_IRQENABLE_CLR Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | ERR_LUT_MISS3 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 3 | R/W1C |
| 30 | ERR_UPD_DATA3 | 0 | Data register update whilst refilling error event in area 3 | R/W1C |
| 29 | ERR_UPD_CTRL3 | 0 | Control register update whilst refilling error event in area 3 | R/W1C |
| 28 | ERR_UPD_AREA3 | 0 | Area register update whilst refilling error event in area 3 | R/W1C |
| 27 | ERR_INV_DATA3 | 0 | Invalid entry-table pointer error event in area 3 | R/W1C |
| 26 | ERR_INV_DSC3 | 0 | Invalid descriptor pointer error event in area 3 | R/W1C |
| 25 | FILL_LST3 | 0 | End of refill event for the last descriptor in area 3 | R/W1C |
| 24 | FILL_DSC3 | 0 | End of refill event for any descriptor in area 3 | R/W1C |
| 23 | ERR_LUT_MISS2 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 2 | R/W1C |
| 22 | ERR_UPD_DATA2 | 0 | Data register update whilst refilling error event in area 2 | R/W1C |
| 21 | ERR_UPD_CTRL2 | 0 | Control register update whilst refilling error event in area 2 | R/W1C |
| 20 | ERR_UPD_AREA2 | 0 | Area register update whilst refilling error event in area 2 | R/W1C |
| 19 | ERR_INV_DATA2 | 0 | Invalid entry-table pointer error event in area 2 | R/W1C |
| 18 | ERR_INV_DSC2 | 0 | Invalid descriptor pointer error event in area 2 | R/W1C |
| 17 | FILL_LST2 | 0 | End of refill event for the last descriptor in area 2 | R/W1C |
| 16 | FILL_DSC2 | 0 | End of refill event for any descriptor in area 2 | R/W1C |
| 15 | ERR_LUT_MISS1 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 1 | R/W1C |
| 14 | ERR_UPD_DATA1 | 0 | Data register update whilst refilling error event in area 1 | R/W1C |
| 13 | ERR_UPD_CTRL1 | 0 | Control register update whilst refilling error event in area 1 | R/W1C |
| 12 | ERR_UPD_AREA1 | 0 | Area register update whilst refilling error event in area 1 | R/W1C |
| 11 | ERR_INV_DATA1 | 0 | Invalid entry-table pointer error event in area 1 | R/W1C |
| 10 | ERR_INV_DSC1 | 0 | Invalid descriptor pointer error event in area 1 | R/W1C |
| 9 | FILL_LST1 | 0 | End of refill event for the last descriptor in area 1 | R/W1C |
| 8 | FILL_DSC1 | 0 | End of refill event for any descriptor in area 1 | R/W1C |
| 7 | ERR_LUT_MISS0 | 0 | Unexpected Access to a yet-to-be-refilled area event in area 0 | R/W1C |
| 6 | ERR_UPD_DATA0 | 0 | Data register update whilst refilling error event in area 0 | R/W1C |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 6-24. DMM_PAT_IRQENABLE_CLR Register Field Descriptions (continued)**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 5 | ERR_UPD_CTRL0 | 0 | Control register update whilst refilling error event in area 0 | R/W1C |
| 4 | ERR_UPD_AREA0 | 0 | Area register update whilst refilling error event in area 0 | R/W1C |
| 3 | ERR_INV_DATA0 | 0 | Invalid entry-table pointer error event in area 0 | R/W1C |
| 2 | ERR_INV_DSC0 | 0 | Invalid descriptor pointer error event in area 0 | R/W1C |
| 1 | FILL_LST0 | 0 | End of refill event for the last descriptor in area 0 | R/W1C |
| 0 | FILL_DSC0 | 0 | End of refill event for any descriptor in area 0 | R/W1C |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.15 DMM PAT Status Registers: DMM_PAT_STATUS_0-DMM_PAT_STATUS_3

The DMM PAT Status register is shown in Figure 6-71 and described in Table 6-25.

**Figure 6-71. DMM_PAT_STATUS Registers**

| 31 | 25 | 24 | 16 |
|---|---|---|---|
| Reserved | | CNT | |
| R-0 | | R-0 | |

| 15 | 10 | 9 | 8 |
|---|---|---|---|
| ERROR | | Reserved | |
| R-0 | | R-0 | |

| 7 | 6 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| BYPASSED | Reserved | | DONE | RUN | VALID | READY |
| R-0 | R-0 | | R-0 | R-0 | R-0 | R-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-25. DMM_PAT_STATUS Registers Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31-25 | Reserved | 0 | Reserved | R |
| 24-16 | CNT | 0 | Counter of remaining lines to re-load for engine n | R |
| 15-10 | ERROR | | Error occurred in engine n | R |
| | | 0 | No error | |
| | | 1h | Invalid descriptor provided | |
| | | 2h | Invalid data pointer provided | |
| | | 4h | Unexpected area register update whilst refilling | |
| | | 8h | Unexpected control register update whilst refilling | |
| | | 10h | Unexpected data register update whilst refilling | |
| | | 20h | Unexpected access to a yet-to-be-refilled location | |
| 9-8 | Reserved | 0 | Reserved | R |
| 7 | BYPASSED | 0 | Engine n is bypassed. Direct access to the LUT is provided. | R |
| 6-4 | Reserved | 0 | Reserved | R |
| 3 | DONE | 0 | Area reloading finished for engine n | R |
| 2 | RUN | 0 | Area currently reloading for engine n | R |
| 1 | VALID | 0 | Valid area description for engine n | R |
| 0 | READY | 1 | Area registers ready for engine n | R |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.16 DMM PAT Descriptor Registers: DMM_PAT_DESCR_0-DMM_PAT_DESCR_3

The DMM PAT Descriptor register is shown in Figure 6-72 and described in Table 6-26.

- Physical address of the next table refill descriptor
- Writing to this register aborts the current ongoing area reload
- Write also resets the corresponding DMM_PAT_AREA_x, DMM_PAT_CTRL_x and DMM_PAT_DATA_x registers
- The descriptor address must be in the DDR and not any other internal memory

**Figure 6-72. DMM_PAT_DESCR Registers**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| ADDR | | Reserved | |
| R/W-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-26. DMM_PAT_DESCR Registers Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31-4 | ADDR | 0 | Physical address of the next table refill descriptor. | R/W |
| 3-0 | Reserved | 0 | Reserved | R |

### 6.4.17 DMM PAT Area Geometry Registers: DMM_PAT_AREA_0-DMM_PAT_AREA_3

The DMM PAT Area Geometry register is shown in Figure 6-73 and described in Table 6-27.

**Figure 6-73. DMM_PAT_AREA Registers**

| 31 | 30 | 24 | 23 | 16 | 15 | 14 | 8 | 7 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Rsvd | Y1 | | X1 | | Rsvd | Y0 | | X0 | |
| R-0 | R/W-0 | | R/W-0 | | R-0 | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-27. DMM_PAT_AREA Registers Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | Reserved | 0 | Reserved | R |
| 30-24 | Y1 | 0 | Y-coordinate of the bottom right corner of the PAT area | R/W |
| 23-16 | X1 | 0 | X-coordinate of the bottom right corner of the area | R/W |
| 15 | Reserved | 0 | Reserved | R |
| 14-8 | Y0 | 0 | Y-coordinate of the top left corner of the PAT area | R/W |
| 7-0 | X0 | 0 | X-coordinate of the top left corner of the PAT area | R/W |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.18 DMM PAT Control Registers: DMM_PAT_CTRL_0-DMM_PAT_CTRL_3

The DMM PAT Control register is shown in Figure 6-74 and described in Table 6-28.

**Figure 6-74. DMM_PAT_CTRL Registers**

| 31 28 | 27 17 | 16 |
|---|---|---|
| INITIATOR | Reserved | SYNC |
| R/W-0 | R-0 | R/W-0 |

| 15 10 | 9 8 | 7 | 6 4 | 3 1 | 0 |
|---|---|---|---|---|---|
| Reserved | LUT_ID | Rsvd | DIRECTION | Reserved | START |
| R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-28. DMM_PAT_CTRL Registers Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31-28 | INITIATOR | 0 | CONT_ID of the initiator for synchronisation | R/W |
| 27-17 | Reserved | 0 | Reserved | R |
| 16 | SYNC | | DMM PAT table reload synchronisation | R/W |
| | | 0 | Not synchronised | |
| | | 1 | Synchronised | |
| 15-10 | Reserved | 0 | Reserved | R |
| 9-8 | LUT_ID | 0 | PAT LUT index, which is 0, pointing to the single LUT in the device | R/W |
| 7 | Reserved | 0 | Reserved | R |
| 6-4 | DIRECTION | 0 | Direction of this PAT table refill | R/W |
| 3-1 | Reserved | 0 | Reserved | R |
| 0 | START | 0 | Starting a PAT table refill | R/W |

### 6.4.19 DMM PAT Area Entry Data Registers: DMM_PAT_DATA_0-DMM_PAT_DATA_3

The DMM PAT Area Entry Data register is shown in Figure 6-75 and described in Table 6-29.

- Physical address of the current table refill entry data
- The entry data table must in the DDR and the address should be 32 bytes aligned

**Figure 6-75. DMM_PAT_DATA Registers**

| 31 4 | 3 0 |
|---|---|
| ADDR | Reserved |
| R/W-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-29. DMM_PAT_DATA Registers Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31-4 | ADDR | 0 | Physical address of the current table refill entry data or single actual entry data when in manual mode | R/W |
| 3-0 | Reserved | 0 | Reserved | R |


Copyright © 2011–2016, Texas Instruments Incorporated

### 6.4.20 DMM PEG Priority Registers: DMM_PEG_PRIO_0-DMM_PEG_PRIO_7

The DMM PEG Priority register is shown in Figure 6-76 and described in Table 6-30.

**Figure 6-76. DMM_PEG_PRIO Registers**

| 31 | 30 28 | 27 | 26 24 | 23 | 22 20 | 19 | 18 16 |
|---|---|---|---|---|---|---|---|
| W7 | P7 | W6 | P6 | W5 | P5 | W4 | P4 |
| R/W-0 | R/W-4h | R/W-0 | R/W-4h | R/W-0 | R/W-4h | R/W-0 | R/W-4h |

| 15 | 14 13 | 12 11 | 10 9 | 8 7 | 6 4 | 3 | 2 0 |
|---|---|---|---|---|---|---|---|
| W3 | P3 | W2 | P2 | W1 | P1 | W0 | P0 |
| R/W-0 | R/W-4h | R/W-0 | R/W-4h | R/W-0 | R/W-4h | R/W-0 | R/W-4h |

LEGEND: R/W = Read/Write; R = Read only; n = 0 for the first priority register; n = 1 for the second priority register

**Table 6-30. DMM_PEG_PRIO Registers Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31 | W7 | 0 | Write-enable for P7 bit-field; P7 field is updated | R/W |
| 30-28 | P7 | 4h | Priority for initiator 8.n+7 | R/W |
| 27 | W6 | 0 | Write-enable for P6 bit-field; P6 field is updated | R/W |
| 26-24 | P6 | 4h | Priority for initiator 8.n+6 | R/W |
| 23 | W5 | 0 | Write-enable for P5 bit-field; P5 field is updated | R/W |
| 22-20 | P5 | 4h | Priority for initiator 8.n+5 | R/W |
| 19 | W4 | 0 | Write-enable for P4 bit-field; P4 field is updated | R/W |
| 18-16 | P4 | 4h | Priority for initiator 8.n+4 | R/W |
| 15 | W3 | 0 | Write-enable for P3 bit-field; P3 field is updated | R/W |
| 14-12 | P3 | 4h | Priority for initiator 8.n+3 | R/W |
| 11 | W2 | 0 | Write-enable for P2 bit-field; P2 field is updated | R/W |
| 10-8 | P2 | 4h | Priority for initiator 8.n+2 | R/W |
| 7 | W1 | 0 | Write-enable for P1 bit-field; P1 field is updated | R/W |
| 6-4 | P1 | 4h | Priority for initiator 8.n+1 | R/W |
| 3 | W0 | 0 | Write-enable for P0 bit-field; P0 field is updated | R/W |
| 2-0 | P0 | 4h | Priority for initiator 8.n | R/W |

### 6.4.21 DMM PEG Priority Registers for PAT: DMM_PEG_PRIO_PAT

The DMM PEG Priority register for PAT is shown in Figure 6-77 and described in Table 6-31. The DMM PEG Priority register is for the internal PAT engine.

**Figure 6-77. DMM_PEG_PRIO_PAT Register**

| 31 4 | 3 | 2 0 |
|---|---|---|
| Reserved | W_PAT | P_PAT |
| R-0 | R/W-0 | R/W-4h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 6-31. DMM_PEG_PRIO_PAT Register Field Descriptions**

| Bit | Field | Value | Description | Type |
|---|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved | R |
| 3 | W_PAT | 0 | Write-enable for P_PAT bit-field; P_PAT field is updated | R/W |
| 2-0 | P_PAT | 4h | Priority for PAT engine. | R/W |


Copyright © 2011–2016, Texas Instruments Incorporated

# DDR2 and DDR3 Memory Controller

This document describes the DDR2 and DDR3 memory controller specifications and general information.

**Topic** **Page**


| | | |
|---|---|---|
| 7.1 | Introduction | 1227 |
| 7.2 | Architecture | 1227 |
| 7.3 | DDR PHY | 1244 |
| 7.4 | Electrical Characteristics Control | 1246 |
| 7.5 | DDR2/DDR3 SDRAM Memory Initialization | 1248 |
| 7.6 | Interfacing the DDR2/DDR3 Memory Controller to DDR Memory Devices | 1250 |
| 7.7 | DDR2/DDR3 SDRAM Device Configuration Steps | 1260 |
| 7.8 | DDR2/DDR3 Configuration Registers | 1261 |



Copyright © 2011–2016, Texas Instruments Incorporated

## 7.1 Introduction

This document describes the DDR2 and DDR3 memory controller specifications and general information.

### 7.1.1 Purpose of the Peripheral

The DDR2 and DDR3 memory controller is used to interface with JESD79-2E and JESD79-3C standard-compliant DDR2 and DDR3 SDRAM devices, respectively. Memory types such as DDR1 SDRAM, SDR SDRAM, SBSRAM, and asynchronous memories are not supported.

### 7.1.2 Features Supported

The DDR2 and DDR3 memory controller supports the following features:

- JESD79-2E standard compliant DDR2 SDRAM
- JESD79-3C standard compliant DDR3 SDRAM
- 1024 Mbyte memory space
- Data bus width of 32 or 16 bits
- CAS latencies(DDR2): 3,4,5,6 and 7
- CAS latencies(DDR3): 5,6,7,8,9,10 and 11
- Internal banks DDR2 and DDR3: 1,2,4,8
- Capable of supporting two chip select signals (refer to the device data manual for the exact number of chip selects supported)
- Burst length:8
- Burst type: Sequential
- Page sizes: 256, 512, 1024 and 2048 word
- SDRAM auto-initialization on configuration change
- Self-refresh and power-down modes for low power
- Periodic ZQ calibration (DDR3 only)
- Self-refresh mode
- Prioritized refresh
- Programmable refresh rate and backlog counter
- Programmable timing parameters
- Little endian
- Software write/read leveling (calibration)
- Support for fly-by topology board routing for DDR3 interfaces via leveling
- ODT on DDR2 and DDR3
- Class of service support

### 7.1.3 Features Not Supported

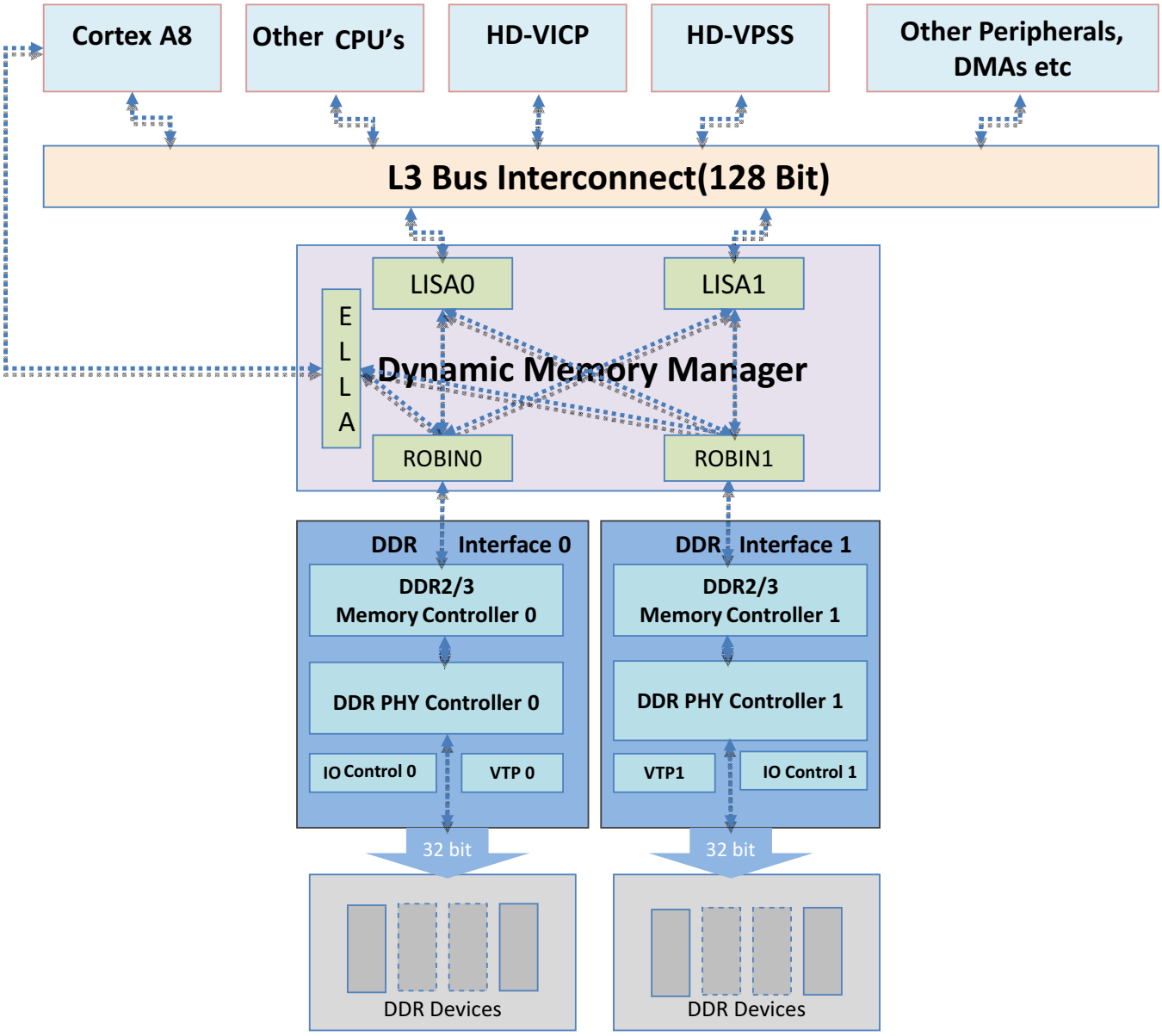
- DDR1, LPDDR1, and LPDDR2 memory types
- Burst Length other than 8
- Interleave burst type
- OCD calibration for DDR2
- DDR3 burst chop support

## 7.2 Architecture

Figure 7-1 shows the DDR subsystem interconnection to the rest of the system and the data paths. Please refer to the *DMM/TILER* chapter in the TRM for a detailed description of the dynamic memory controller. This chapter focuses only on the DDR subsystem and various sub-blocks inside the DDR subsystem.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 7-1. DDR Block Diagram**




The DDR2/DDR3 memory controllers and their associated PHYs provide the physical interface to external DDR2 or DDR3 memory devices. Data, which is read from or written to external memories by internal initiators (for example, Cortex A8, HD-VPSS, etc), is managed internally by the Dynamic Memory Manager (DMM).

In the most common usage case, the DMM acts to transparently interleave memory accesses across the two DDR interfaces on a 128B, 256B or 512B boundary. For example, the EDMA could write an 8KB block of consecutive addresses to external memory space and if 512B interleaving is enabled, the DMM would cause the write to be interleaved across both DDR interfaces on 256B boundaries. Likewise, when the same 8KB block of addresses is read back from external memory, the DMM would interleave the reads from appropriate sub-blocks so that the original data set is read back. This interleaving process substantially improves memory efficiency and is transparent to the memory requestors.

It is also possible to operate with interleaving function deactivated in which case the two external DDR interfaces can be mapped to different address spaces.


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.2.1 Signal Descriptions

The DDR2 and DDR3 memory controller signals are shown in Figure 7-2 and described in Table 7-1. The following features are included:

- The maximum width for the data bus (DDR[x]_D[31:0]) is 32-bits
- The address bus (DDR[x]_A[14:0]) is 15-bits wide with an additional 3-bank address pins (DDR[x]_BA[2:0])
- Two differential output clocks (DDR[x]_CLK and $\overline{\text{DDR[x]_CLK}}$ driven by internal clock sources
- Command signals: Row and column address strobe ($\overline{\text{DDR[x]_RAS}}$ and $\overline{\text{DDR[x]_CAS}}$), write enable strobe

($\overline{\text{DDR[x]_WE}}$), data strobe (DDR[x]_DQS[3:0] and $\overline{\text{DDR[x]_DQS[3:0]}}$), and data mask (DDR[x]_DQM[3:0]).

- Two chip-select signals ($\overline{\text{DDR[x]_CS[1:0]}}$) and one clock enable signal (DDR[x]_CKE)
- Two on-die termination output signals (DDR[x]_ODT[1:0])

**Figure 7-2. DDR2 and DDR3 Memory Controller Signals**

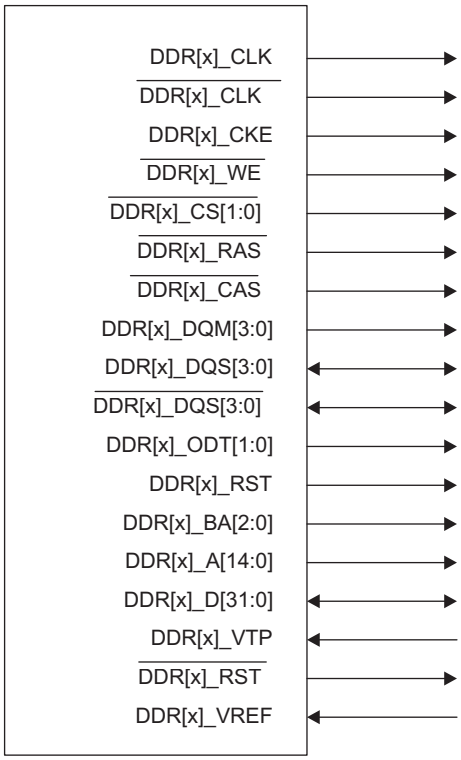



**Table 7-1. DDR2 and DDR3 Memory Controller Signal Descriptions**

| Pin | Description |
|---|---|
| DDR[x]_D[31:0] | Bidirectional data bus. Input for data reads and output for data writes. |
| DDR[x]_A[14:0] | External address output. |
| $\overline{\text{DDR[x]_CS[1:0]}}$ | Chip select output. Allows two banks of DDR2 and DDR3. |
| DDR[x]_DQM[3:0] | Active-low output data mask. |
| DDR[x]_CLK/$\overline{\text{DDR[x]_CLK}}$ | Differential clock outputs. All DDR2 and DDR3 interface signals are synchronous to these clocks. |
| DDR[x]_CKE | Clock enable. Used to select power-down and self-refresh operations. |
| $\overline{\text{DDR[x]_CAS}}$ | Active-low column address strobe. |
| $\overline{\text{DDR[x]_RAS}}$ | Active-low row address strobe. |
| $\overline{\text{DDR[x]_WE}}$ | Active-low write enable. |
| DDR[x]_DQS[3:0]/$\overline{\text{DDR[x]_DQS[3:0]}}$ | Differential data strobe bidirectional signals. Edge-aligned inputs on reads and center-aligned outputs on writes. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-1. DDR2 and DDR3 Memory Controller Signal Descriptions (continued)**

| Pin | Description |
|---|---|
| DDR[x]_ODT[1:0] | ODT[1:0] On-die termination signals to external DDR2 and DDR3 SDRAM per chip select. |
| DDR[x]_BA[2:0] | Bank-address control outputs. |
| DDR[x]_VREF | Memory Controller reference voltage. This voltage must be supplied externally. See the device-specific data manual for more details. |
| DDR[x]_VTP | DDR2 and DDR3 VTP Compensation Resistor Connection. |
| $\overline{\text{DDR[x]_RST}}$ | Reset output. Asynchronous reset for DDR3 devices. |

### 7.2.2 Memory Map

Please see the device-specific data manual for information describing the device memory map.

### 7.2.3 Clock Control

The DDR2/DDR3 clock is derived directly from the DDR PLL's VCO output. The frequency of DDR[x]_CLK can be determined by using the following formula:

DDR[x]_CLK frequency = (DDRPLL input clock frequency x mulitplier)/((pre-divider+1)*post-divider)

The second output clock of the DDR2/DDR3 memory controller $\overline{\text{DDR[x]_CLK}}$, is the inverse of DDR[x]_CLK. You can change the muliplier, pre-divider and post-divider to get the desired DDR[x]_CLK frequency.

Where x=0,1

For detailed information on DDRPLL, see the chip level resources.

### 7.2.4 DDR2/DDR3 Memory Controller Subsytem Overview

The DDR2/DDR3 memory controller can gluelessly interface to most standard DDR2/DDR3 SDRAM devices and supports such features as self-refresh mode and prioritized refresh. In addition, it provides flexibility through programmable parameters such as the refresh rate, CAS latency, and many SDRAM timing parameters. The DDR2/DDR3 subsystem consists of the following:

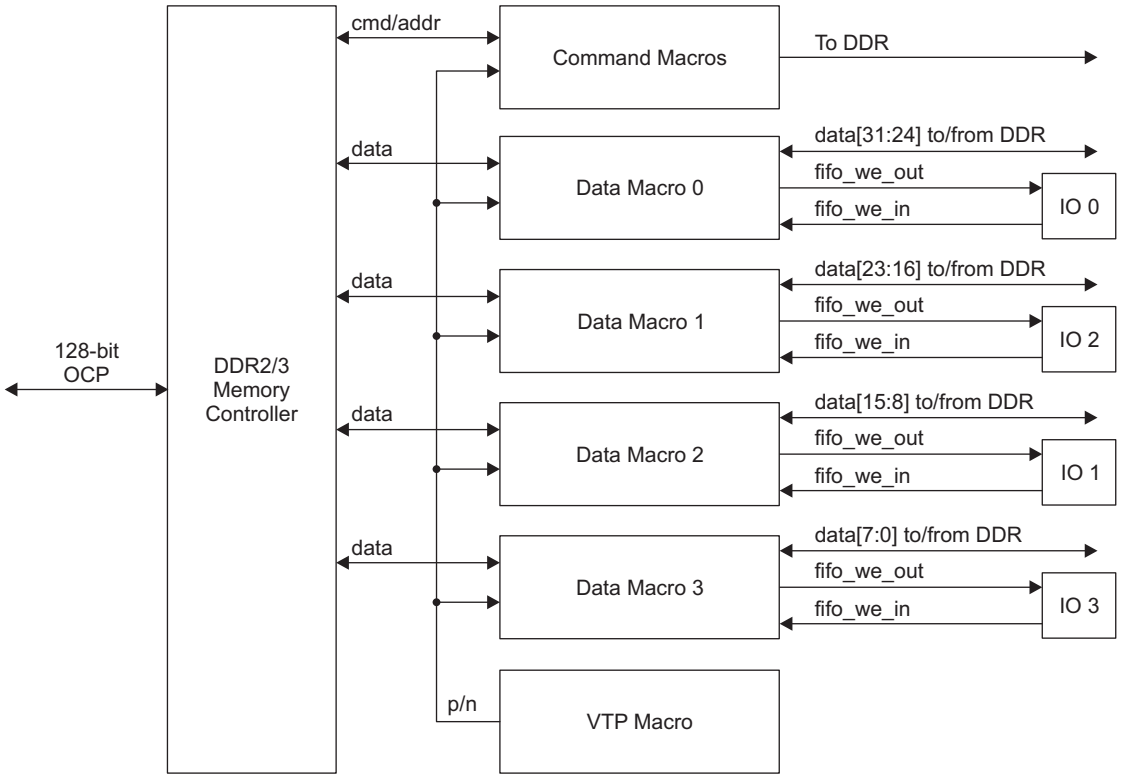
- DDR2/DDR3 memory controller
- Command macro
- Data macro
- VTP controller macro
- IO's for DQS gate

The subsystem supports JEDEC standard-compliant DDR2 and DDR3 devices. It does not support CAS latency of two for DDR2 due to data and command macro limitations. It supports a 128-bit wide OCP interface on the core side for programmability. The subsystem can be used to connect to 16- or 32-bit memory devices.

Figure 7-3 shows the DDR2/DDR3 subsystem block diagram.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 7-3. DDR2/DDR3 Subsystem Block Diagram**




Where
fifo_we_out = DQS enable output for timing match between DQS and system (Memory) clock.
fifo_we_in = DQS enable input for timing match between DQS and system (Memory) clock.

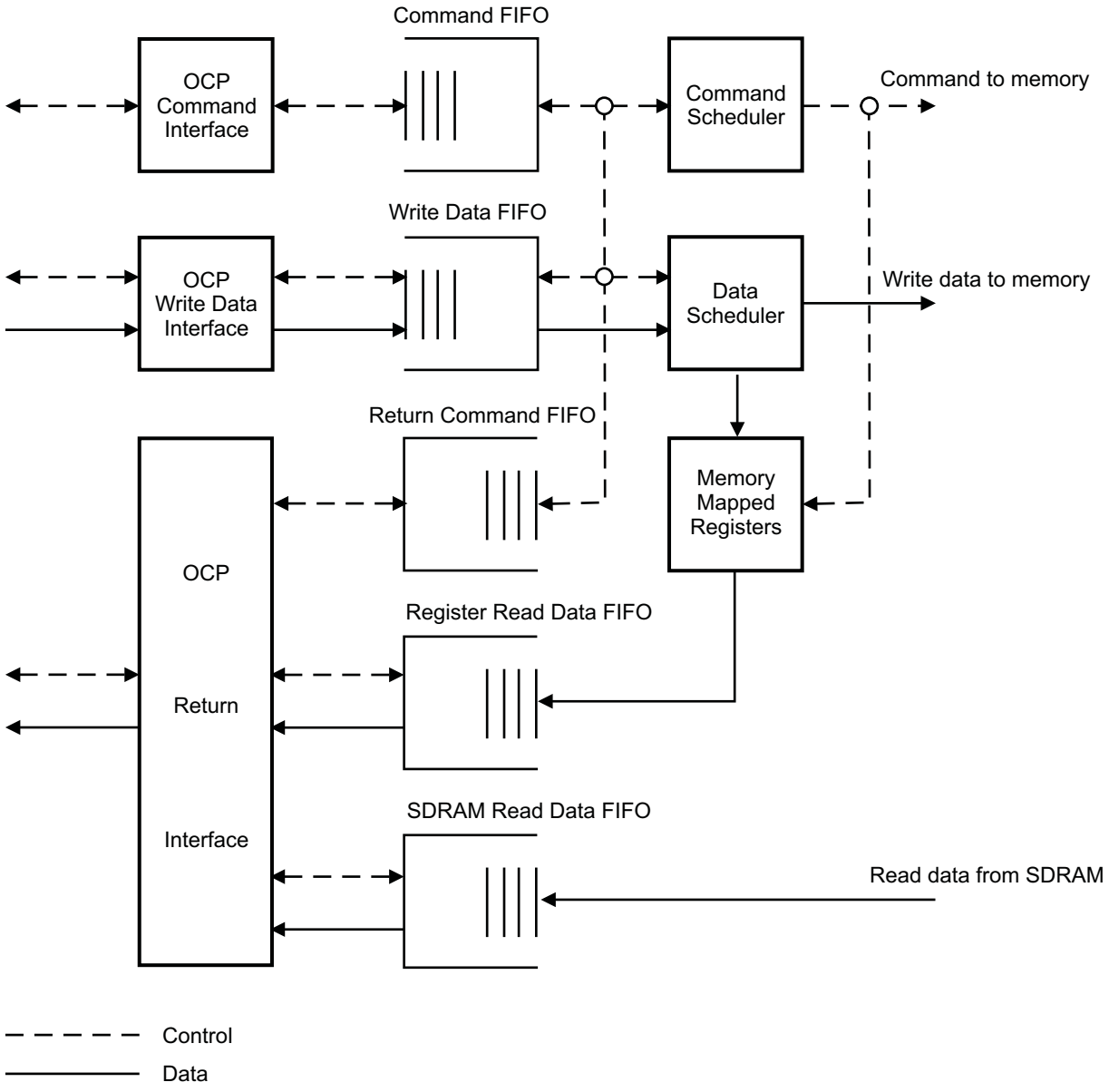
### 7.2.4.1 DDR2/DDR3 Memory Controller Interface

To move data efficiently from on-chip resources to the external DDR2/DDR3 SDRAM device, the DDR2/DDR3 memory controller makes use of a command FIFO, a write data FIFO, a return command FIFO, and two read data FIFOs. The purpose of each FIFO is described below.

Figure 7-4 shows the block diagram of the DDR2/DDR3 memory controller FIFOs. Commands, write data, and read data arrive at the DDR2/DDR3 memory controller parallel to each other. The same peripheral bus is used to write and read data from external memory as well as internal memory-mapped registers.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 7-4. DDR2/DDR3 Memory Controller FIFO Block Diagram**




The command FIFO stores all the commands coming in on the OCP command interface.

The Write Data FIFO stores the write data for all the write transactions coming in on the OCP write data interface.

The Return Command FIFO stores all the return transactions that are to be issued to the OCP return interface. These include the write status return and the read data return commands.

There are two Read Data FIFOs that store the read data to be sent to the OCP return interface. One Read Data FIFO stores read data from the memory mapped registers and other Read Data FIFO stores read data from external memory.

### 7.2.4.2 Data Macro

The data macro consists of eight data channels, one pair of complementary strobes (one pair for 8 bits of data), and one data mask channel (one for eight bits of data).

The data macros consists of PHY data macro, DLLs and IOs integrated into a macro.


Copyright © 2011–2016, Texas Instruments Incorporated

The data macro is a bidirectional interface. It is used to transmit data from the memory controller to the external memory chip during a write operation and receive data from memory and transmit it to the memory controller during a read operation.

During a write operation, the data macro translates 32/16-bit words from memory controller to 8-bit words and transmits them at double the bit rate to the memory along with the strobe. The strobe is center-aligned to the data. Data can be prevented from writing to the memory using data mask signal.

During a read operation, the data macro receives 8-bit DDR data along with the strobe and converts it to 32/16- bit words and transmits them to the memory controller along with the read-valid signals.

#### 7.2.4.3 Command Macro

It consists of the PHY command macro, DLLs and the IOs integrated together. The command macro acts as a unidirectional macro that transmits address and control bits from memory controller to the memory chip. The clocks DDR[x]_CLK and $\overline{\text{DDR[x]_CLK}}$ are used by the memory to register the command and address transmitted on the transmit channels. All address and control signals are transmitted clock-centered with respect to DDR[x]_CLK and $\overline{\text{DDR[x]_CLK}}$. The memory, on the positive edge of DDR[x]_CLK and the negative edge of $\overline{\text{DDR[x]_CLK}}$[x], samples all address and control signals.

#### 7.2.4.4 VTP Controller Macro

The VTP controller macro evaluates silicon performance at current voltage, temperature, and process (VTP) to enable IO drivers to set constant predetermined output driver impedance. The controller operates by comparing driver impedances to the external reference resistor and adjusting driver impedance to obtain an impedance match. The VTP controller supports the following features:

- The VTP controller generates information regarding the voltage, temperature, and process(VTP) on a chip to be shared with the device's IO drivers
- Requires a clock input from the core running at 20 MHz or less
- Requires 56 clock cycles to guarantee the VTP outputs are initially set after reset is removed
- Can be used in static or dynamic update mode of operation
- The VTP controller has internal noise filtering which allows it to control spurious update requests due to noise

Impedance of the drivers and terminations must be updated often even while in operation. In such scenarios where voltage and temperature are variables VTP macro can be configured in dynamic update mode. In contrast, static mode of operation does not allow dynamic calibration of IO impedance, and hence consumes lesser power compared to dynamic update mode.

It is possible that under noisy conditions, dynamic update controller can generate too frequent update requests. Noise can cause the controller to request a change in the impedance that can again be quickly reversed on subsequent clock cycles. To prevent the controller from making excessive numbers of impedance changes, a digital filter is included which can be configured to regulate the update rate. For example, if the user configures the filter value as F2=0,F1=1 and F0=1, then an impedance update will be performed only if four successive update requests are generated from the VTP controller. It is recommended to use a filter value of 011'b.

Table 7-2 shows the configuration details of the digital filter.

**Table 7-2. Digital Filter Configuration**

| F2 | F1 | F0 | Description |
|---|---|---|---|
| 0 | 0 | 0 | Off |
| 0 | 0 | 1 | Update on 2 consecutive update requests |
| 0 | 1 | 0 | Update on 3 consecutive update requests |
| 0 | 1 | 1 | Update on 4 consecutive update requests |
| 1 | 0 | 0 | Update on 5 consecutive update requests |
| 1 | 0 | 1 | Update on 6 consecutive update requests |
| 1 | 1 | 0 | Update on 7 consecutive update requests |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-2. Digital Filter Configuration (continued)**

| F2 | F1 | F0 | Description |
|---|---|---|---|
| 1 | 1 | 1 | Update on 8 consecutive update requests |

#### 7.2.4.5 DQS-Gate IOs

To effectively model the I/O delay on the DQS gating signal during a read request (the DQS receiver and the CLK driver I/Os), the signal is expected to be looped on a single I/O connecting fifo_we_in to fifo_we_out. The board and memory delay, being fairly constant across PTV variations, are calibrated within the IDID using a compensated delay line. The loop-back is done at the die level without bringing the signals out to the package level. Each data macro supports delay compensation independent of each other.

The data and command macros are responsible for system level flight time compensation. The following controls are supported by the DDR2/DDR3 controller subsystem:

- Aligning DDR[x]_DQS w.r.t DDR[x]_CLK during Write Cycle. To produce a given amount of skew to center the DDR[x]_DQS signals vsersus the SDRAM clock (DDR[x]_CLK) the DATA0/1/2/3_REG_PHY_WR_DQS_SLAVE_RATIO_0 registers must be programmed. The value to be loaded in these registers is calculated through the following formula:

  Data Macro 0/1/2/3 Write DQS Slave Ratio Register=256 x ([command delay] – [DDR[x]_DQS delay]) / DDR[x]_CLK clock period.
- Aligning ADDR/CMD w.r.t DDR[x]_CLK
- Aligning DDR[x]_DQ[31:0] w.r.t DDR[x]_DQS during write operation
- Offset DDR[x]_D[31:0] w.r.t DDR[x]_DQS during read operation
- Align FIFO WE window

### *7.2.5 Address Mapping*

Each DDR2/DDR3 memory controller views external DDR2/DDR3 SDRAM as one continuous block of memory. This statement is true regardless of the number of memory devices located on the chip select space. The DDR2/DDR3 memory controller receives DDR2/DDR3 memory access requests along with a 32-bit logical address from the rest of the system. In turn, this memory controller uses the logical address to generate a row/page, column, and bank address for the DDR2/DDR3 SDRAM. The number of column, row, and bank address bits used is determined by the IBANK, RSIZE, and PAGESIZE fields (see Table 7-3). The DDR2/DDR3 memory controller uses up to 15 bits for the row/page address.

**Table 7-3. IBANK, RSIZE and PAGESIZE Fields Information**

| Bit Field | Bit Value | Bit Description |
|---|---|---|
| RSIZE | | Defines the number of address lines to be connected to DDR2/DDR3 memory device |
| | 0 | 9 row bits |
| | 1h | 10 row bits |
| | 2h | 11 row bits |
| | 3h | 12 row bits |
| | 4h | 13 row bits |
| | 5h | 14 row bits |
| | 6h | 15 row bits |
| PAGESIZE | | Defines the page size of each page of the external DDR2/DDR3 memory device |
| | 0 | 256 words (requires 8 column address bits) |
| | 1h | 512 words (requires 9 column address bits) |
| | 2h | 1024 words (requires 10 column address bits) |
| | 3h | 2048 words (requires 11 column address bits) |
| IBANK | | Defines the number of internal banks on the external DDR2/DDR3 memory device |
| | 0 | 1 bank |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-3. IBANK, RSIZE and PAGESIZE Fields Information (continued)**

| Bit Field | Bit Value | Bit Description |
|---|---|---|
| | 1h | 2 banks |
| | 2h | 4 banks |
| | 3h | 8 banks |
| EBANK | | Defines whether DDR2/DDR3 memory controller accesses will use 1 or 2 chip selects |
| | 0 | CS0 only |
| | 1h | CS0, CS1 |

When addressing SDRAM, if the IBANK_POS field in the SDRAM Config register is set to 0, and the EBANK_POS field in the SDRAM Config 2 register is also set to 0, the DDR2/DDR3 memory controller uses the three fields, IBANK, EBANK and PAGESIZE in the SDRAM Config register to determine the mapping from source address to SDRAM row, column, bank, and chip select.

If the IBANK_POS field in the SDRAM Config register is set to 1, 2, or 3, or the EBANK_POS field in the SDRAM Config 2 register is set to 1, the DDR2/DDR3 memory controller uses the four fields - IBANK, EBANK, PAGESIZE, and ROWSIZE in the SDRAM Config register to determine the mapping from source address to SDRAM row, column, bank, and chip select.

In all cases, the DDR2/DDR3 memory controller considers its SDRAM address space to be a single logical block, regardless of the number of physical devices or whether the devices are mapped across 1 or 2 memory controller chip selects.

### 7.2.5.1 Address Mapping when IBANK_POS=0 and EBANK_POS=0

For IBANK_POS=0 and EBANK_POS=0, the effect of address mapping scheme is that, as the source address increments across DDR2/DDR3 memory device page boundaries, the DDR2/DDR3 controller moves onto the same page in the next bank in the current device $\overline{\text{DDR[x]_CS[0]}}$. This movement along the banks of the current proceeds to the same page in the next device (if EBANK=1, $\overline{\text{DDR[x]_CS[1]}}$) and proceeds through the same page in all its banks before moving over to the next page in the first device ($\overline{\text{DDR[x]_CS[0]}}$). The DDR2/DDR3 controller exploits this traversal across internal banks and chip selects while remaining on the same page, to maximize the number of open DDR2/DDR3 memory device banks within the overall memory device space.

Thus, the DDR2/DDR3 controller can keep a maximum of 16 banks (eight internal banks across two chip selects) open at a time, and can interleave among all of them.

**Table 7-4. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=0 and EBANK_POS=0**

| Logical Address | | | |
|---|---|---|---|
| **Row Address** | **Chip Select** | **Bank Address** | **Column Address** |
| | # of bits defined by EBANK of SDRCR | # of bits defined by IBANK of SDRCR | # of bits defined by PAGESIZE of SDRCR |
| 15 bits | EBANK=0 => 0 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| | EBANK=1 => 1 bit | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
| | | IBANK=2 => 2 bits | PAGESIZE=2 => 10 bits |
| | | IBANK=3 => 3 bits | PAGESIZE=3 => 11 bits |

### 7.2.5.2 Address Mapping when IBANK_POS= 1 and EBANK_POS = 0

For IBANK_POS= 1 and EBANK_POS = 0, the interleaving of banks within a device (per chip select) is limited to four banks. However, it can still interleave banks between the two chip selects. Thus, the DDR2/DDR3 controller can keep a maximum of 16 banks (8 internal banks across two chip selects) open at a time, but can only interleave among eight of them.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-5. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=1 and EBANK_POS=0**

| Logical Address | | | | |
|---|---|---|---|---|
| **Bank Address[2]** | **Row Address** | **Chip Select** | **Bank Address[1:0]** | **Column Address** |
| # of bits defined by IBANK of SDRCR | # of bits defined by RSIZE of SDRCR | # of bits defined by EBANK of SDRCR | # of bits defined by IBANK of SDRCR | # of bits defined by PAGESIZE of SDRCR |
| IBANK=0 => 0 bits | RSIZE=0 => 9 bits | EBANK=0 => 0 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| IBANK=1 => 0 bits | RSIZE=1 => 10 bits | EBANK=1 => 1 bit | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
| IBANK=2 => 0 bits | RSIZE=2 => 11 bits | | IBANK=2 => 2 bits | PAGESIZE=2 => 10 bits |
| IBANK=3 => 1 bit | RSIZE=3 => 12 bits | | IBANK=3 => 3 bits | PAGESIZE=3 => 11 bits |
| | RSIZE=4 => 13 bits | | | |
| | RSIZE=5 => 14 bits | | | |
| | RSIZE=6 => 15 bits | | | |

### 7.2.5.3 Address Mapping when IBANK_POS=2 and EBANK_POS = 0

For IBANK_POS=2 and EBANK_POS = 0, the interleaving of banks within a device (per chip select) is limited to two banks. However, it can still interleave banks between the two chip selects. Thus, the DDR2/DDR3 controller can keep a maximum of 16 banks (eight internal banks across two chip selects) open at a time, but can only interleave among four of them.

**Table 7-6. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=2 and EBANK_POS=0**

| Logical Address | | | | |
|---|---|---|---|---|
| **Bank Address[2:1]** | **Row Address** | **Chip Select** | **Bank Address[0]** | **Column Address** |
| # of bits defined by IBANK of SDRCR | # of bits defined by RSIZE of SDRCR | # of bits defined by EBANK of SDRCR | # of bits defined by IBANK of SDRCR | # of bits defined by PAGESIZE of SDRCR |
| IBANK=0 => 0 bits | RSIZE=0 => 9 bits | EBANK=0 => 0 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| IBANK=1 => 0 bits | RSIZE=1 => 10 bits | EBANK=1 => 1 bit | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
| IBANK=2 => 1 bit | RSIZE=2 => 11 bits | | IBANK=2 => 1 bit | PAGESIZE=2 => 10 bits |
| IBANK=3 => 2 bits | RSIZE=3 => 12 bits | | IBANK=3 => 1 bit | PAGESIZE=3 => 11 bits |
| | RSIZE=4 => 13 bits | | | |
| | RSIZE=5 => 14 bits | | | |
| | RSIZE=6 => 15 bits | | | |

### 7.2.5.4 Address Mapping when IBANK_POS= 3 and EBANK_POS = 0

For IBANK_POS= 3 and EBANK_POS = 0, the DDR2/DDR3 controller cannot interleave banks within a device (per chip select). However, it can still interleave banks between the two chip selects. Thus, the DDR2/DDR3 controller can keep a maximum of 16 banks (eight internal banks across two chip selects) open at a time, but can only interleave among two of them.

**Table 7-7. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=3 and EBANK_POS=0**

| Logical Address | | | |
|---|---|---|---|
| **Bank Address** | **Row Address** | **Chip Select** | **Column Address** |
| # of bits defined by IBANK of SDRCR | # of bits defined by RSIZE of SDRCR | # of bits defined by EBANK of SDRCR | # of bits defined by PAGESIZE of SDRCR |
| IBANK=0 => 0 bits | RSIZE=0 => 9 bits | EBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| IBANK=1 => 1 bit | RSIZE=1 => 10 bits | EBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
| IBANK=2 => 2 bits | RSIZE=2 => 11 bits | | PAGESIZE=2 => 10 bits |
| IBANK=3 => 3 bits | RSIZE=3 => 12 bits | | PAGESIZE=3 => 11 bits |
| | RSIZE=4 => 13 bits | | |
| | RSIZE=5 => 14 bits | | |
| | RSIZE=6 => 15 bits | | |



Copyright © 2011–2016, Texas Instruments Incorporated

#### 7.2.5.5 Address Mapping when IBANK_POS = 0 and EBANK_POS = 1

For IBANK_POS = 0 and EBANK_POS = 1, the DDR2/DDR3 memory controller interleaves among all the banks within a device (per chip select). However, the DDR2/DDR3 memory controller cannot interleave banks between the two chip selects. Thus, the DDR2/DDR3 memory controller can keep a maximum of 16 banks (eight internal banks across two chip selects) open at a time, but can only interleave among 8 of them.

**Table 7-8. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=0 and EBANK_POS=1**

| Logical Address | | | |
|---|---|---|---|
| **Chip Select** | **Row Address** | **Bank Address** | **Column Address** |
| # of bits defined by EBANK of SDRCR | # of bits defined by RSIZE of SDRCR | # of bits defined by IBANK of SDRCR | # of bits defined by PAGESIZE of SDRCR |
| EBANK=0 => 0 bits | RSIZE=0 => 9 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| EBANK=1 => 1 bit | RSIZE=1 => 10 bits | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
| | RSIZE=2 => 11 bits | IBANK=2 => 2 bits | PAGESIZE=2 => 10 bits |
| | RSIZE=3 => 12 bits | IBANK=3 => 3 bits | PAGESIZE=3 => 11 bits |
| | RSIZE=4 => 13 bits | | |
| | RSIZE=5 => 14 bits | | |
| | RSIZE=6 => 15 bits | | |

#### 7.2.5.6 Address Mapping when IBANK_POS = 1 and EBANK_POS = 1

For IBANK_POS = 1 and EBANK_POS = 1, the interleaving of banks within a device (per chip select) is limited to four banks. Also, the DDR2/DDR3 memory controller cannot interleave banks between the two chip selects. Thus, the DDR2/DDR3 memory controller can keep a maximum of 16 banks (eight internal banks across two chip selects) open at a time, but can only interleave among four of them.

**Table 7-9. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=1 and EBANK_POS = 1**

| Logical Address | | | | |
|---|---|---|---|---|
| **Chip Select** | **Bank Address[2]** | **Row Address** | **Bank Address[1:0]** | **Column Address** |
| # of bits defined by EBANK of SDRCR | # of bits defined by IBANK of SDRCR | # of bits defined by RSIZE of SDRCR | # of bits defined by IBANK of SDRCR | # of bits defined by PAGESIZE of SDRCR |
| EBANK=0 => 0 bits | IBANK=0 => 0 bits | RSIZE=0 => 9 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| EBANK=1 => 1 bit | IBANK=1 => 0 bits | RSIZE=1 => 10 bits | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
| | IBANK=2 => 0 bits | RSIZE=2 => 11 bits | IBANK=2 => 2 bits | PAGESIZE=2 => 10 bits |
| | IBANK=3 => 1 bit | RSIZE=3 => 12 bits | IBANK=3 => 2 bits | PAGESIZE=3 => 11 bits |
| | | RSIZE=4 => 13 bits | | |
| | | RSIZE=5 => 14 bits | | |
| | | RSIZE=6 => 15 bits | | |

#### 7.2.5.7 Address Mapping when IBANK_POS = 2 and EBANK_POS = 1

For IBANK_POS = 2 and EBANK_POS = 1, the interleaving of banks within a device (per chip select) is limited to two banks. Also, the DDR2/DDR3 memory controller cannot interleave banks between the two chip selects. Thus, the DDR2/DDR3 memory controller can keep a maximum of 16 banks (eight internal banks across two chip selects) open at a time, but can only interleave among two of them.

**Table 7-10. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=2 and EBANK_POS = 1**

| Logical Address | | | | |
|---|---|---|---|---|
| **Chip Select** | **Bank Address[2:1]** | **Row Address** | **Bank Address[0]** | **Column Address** |
| # of bits defined by EBANK of SDRCR | # of bits defined by IBANK of SDRCR | # of bits defined by RSIZE of SDRCR | # of bits defined by IBANK of SDRCR | # of bits defined by PAGESIZE of SDRCR |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-10. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=2 and EBANK_POS = 1 (continued)**

| Logical Address | | | | |
|---|---|---|---|---|
| **Chip Select** | **Bank Address[2:1]** | **Row Address** | **Bank Address[0]** | **Column Address** |
| EBANK=0 => 0 bits | IBANK=0 => 0 bits | RSIZE=0 => 9 bits | IBANK=0 => 0 bits | PAGESIZE=0 => 8 bits |
| EBANK=1 => 1 bit | IBANK=1 => 0 bits | RSIZE=1 => 10 bits | IBANK=1 => 1 bit | PAGESIZE=1 => 9 bits |
| | IBANK=2 => 1 bit | RSIZE=2 => 11 bits | IBANK=2 => 1 bit | PAGESIZE=2 => 10 bits |
| | IBANK=3 => 2 bits | RSIZE=3 => 12 bits | IBANK=3 => 1 bit | PAGESIZE=3 => 11 bits |
| | | RSIZE=4 => 13 bits | | |
| | | RSIZE=5 => 14 bits | | |
| | | RSIZE=6 => 15 bits | | |

#### 7.2.5.8 Address Mapping when IBANK_POS = 3 and EBANK_POS = 1

For IBANK_POS = 3 and EBANK_POS = 1, the DDR2/DDR3 memory controller cannot interleave banks within a device (per chip select) or between the two chip selects. Thus, the DDR2/DDR3 memory controller can keep a maximum of 16 banks (eight internal banks across two chip selects) open at a time, but cannot interleave among of them.

**Table 7-11. OCP Address to DDR2/DDR3 Address Mapping for IBANK_POS=3 and EBANK_POS=1**

| Logical Address | | | |
|---|---|---|---|
| **Chip Select** | **Bank Address** | **Row Address** | **Column Address** |
| # of bits defined by EBANK of SDRCR | # of bits defined by IBANK of SDRCR | # of bits defined by RSIZE of SDRCR | # of bits defined by PAGESIZE of SDRCR |
| EBANK=0 => 0 bits | IBANK=0 => 0 bits | RSIZE=0 => 9 bits | PAGESIZE=0 => 8 bits |
| EBANK=1 => 1 bit | IBANK=1 => 1 bit | RSIZE=1 => 10 bits | PAGESIZE=1 => 9 bits |
| | IBANK=2 => 2 bits | RSIZE=2 => 11 bits | PAGESIZE=2 => 10 bits |
| | IBANK=3 => 3 bits | RSIZE=3 => 12 bits | PAGESIZE=3 => 11 bits |
| | | RSIZE=4 => 13 bits | |
| | | RSIZE=5 => 14 bits | |
| | | RSIZE=6 => 15 bits | |

Since the DDR2/DDR3 memory controller interleaves among less number of banks when IBANK_POS!= 0 or EBANK_POS= 1, these cases are lower in performance than the IBANK_POS= 0 case. Thus these cases are only recommended to be used along with partial array self-refresh where performance can be traded off for power savings.

### *7.2.6 Performance Management*

#### 7.2.6.1 Command Ordering and Scheduling

The DDR2/DDR3 memory controller performs command re-ordering and scheduling in an attempt to achieve efficient transfers with maximum throughput. The goal is to maximize the utilization of the data, address, and command buses while hiding the overhead of opening and closing DDR2/DDR3 SDRAM rows. Command re-ordering takes place within the command FIFO.


Copyright © 2011–2016, Texas Instruments Incorporated

The DDR2/DDR3 memory controller examines all the commands stored in the command FIFO to schedule commands to the external memory. All commands from same master will complete in order, regardless of the master priority. The memory controller does not guarantee ordering between commands from different masters. However, the memory controller will maintain data coherency. Therefore, the memory controller will block a command, regardless of master priority, if that command is to the same block address (2048 bytes) as an older command. Thus, the memory controller might have one or more pending read or write for each master. Among all pending reads, the memory controller selects all reads that have their corresponding SDRAM banks already open. Similarly, among all pending writes, the memory controller selects all writes that have their corresponding SDRAM banks already open.

As a result of the above reordering, at any point of time the memory controller might have several pending reads and writes that have their corresponding banks open. The memory controller then selects the highest priority read from pending reads, and the highest priority write from pending writes. If two or more commands have the highest priority, the memory controller selects the oldest command. As a result, the memory controller might now have a final read and a final write command. The memory controller will pick either the read or the write command depending on the value programmed in the Read Write Execution Threshold register. The memory controller will keep executing reads until the read threshold is met and then switch to executing writes. The memory controller will then keep executing writes until the write threshold is met and then switch back to executing reads. The memory controller will satisfy meeting the threshold values only if that type of command is available for execution, otherwise it will switch to the other type. Similarly, the memory controller will satisfy meeting the threshold value only if the FIFOs for that type have space (Read Data FIFO for reads and Write Status FIFO for writes), otherwise it will switch to the other type.

The memory controller completes executing an OCP command before it switches to another command.

All the accesses to an SDRAM are pipe-lined to maximize the external bus utilization. In other words accesses to an SDRAM are issued back to back such that there are minimum idle cycles between any two accesses. This includes the scheduling listed above to minimize the overhead of opening and closing of SDRAM banks. All of these is done while fulfilling the access timing requirements of an SDRAM.

Besides commands received from on-chip resources, the DDR2/DDR3 memory controller also issues refresh commands. The DDR2/DDR3 memory controller attempts to delay refresh commands as long as possible to maximize performance while meeting the SDRAM refresh requirements. As the DDR2/DDR3 memory controller issues read, write, and refresh commands to DDR2/DDR3 SDRAM device, it follows the priority scheme shown in Table 7-12.

**Table 7-12. DDR2/DDR3 Memory Controller Commands Priority**

| Priority Level | Command Description |
|---|---|
| 0 (Highest priority) | SDRAM refresh request due to Refresh Must level of refresh urgency reached (Section 7.2.6.5) |
| 1 | Request for a read or a write. |
| 2 | SDRAM Activate commands. |
| 3 | SDRAM Deactivate commands. |
| 4 | SDRAM Deep Power-Down request. |
| 5 | SDRAM clock stop or Power-Down request. |
| 6 | SDRAM refresh request due to Refresh May or Release level of refresh urgency reached (Section 7.2.6.5) |
| 7 (Lowest priority) | SDRAM self-refresh request. |

### 7.2.6.2 Command Starvation

The reordering and scheduling rules listed above may lead to command starvation, which is the prevention of certain commands from being processed by the DDR2/DDR3 memory controller. Command starvation results from the following conditions:

- A continuous stream of high-priority read commands can block a low-priority write command
- A continuous stream of DDR2/DDR3 SDRAM commands to a row in an open bank can block commands to the closed row in the same bank.


Copyright © 2011–2016, Texas Instruments Incorporated

To avoid these conditions, the DDR2/DDR3 memory controller can momentarily raise the priority of the oldest command in the command FIFO after a set number of transfers have been made. The COS_COUNT_1,COS_COUNT_2 field in the Peripheral Bus Burst Priority Register (PBBPR) sets the number of the transfers that must be made before the DDR2/DDR3 memory controller will raise the priority of the oldest command. See the Section 7.2.6.4 section for more details.

**NOTE:** Leaving the COS bits at their default value (FFh) in the PBBPR register disables this feature of the DDR2/DDR3 memory controller. This means commands can stay in the command FIFO indefinitely. Therefore, these bits should be set to FEh immediately following reset, to enable this feature with the highest level of allowable memory transfers. It is suggested that system-level prioritization be set to avoid placing high-bandwidth masters on the highest priority levels. These bits can be left as FEh unless advanced bandwidth/prioritization control is required.

### 7.2.6.3 Possible Race Condition

A race condition may exist when certain masters write data to the DDR2/DDR3 memory controller. For example, if master A passes a software message via a buffer in DDR2/DDR3 memory and does not wait for indication that the write completes, when master B attempts to read the software message it may read stale data and therefore receive an incorrect message. In order to confirm that a write from master A has landed before a read from master B is performed, master A must wait for the write completion status from the DDR2/DDR3 memory controller before indicating to master B that the data is ready to be read. If master A does not wait for indication that a write is complete, it must perform the following workaround:

1. Perform the required write.
2. Perform a dummy write to the DDR2/DDR3 memory controller module ID and revision register.
3. Perform a dummy read to the DDR2/DDR3 memory controller module ID and revision register.
4. Indicate to master B that the data is ready to be read after completion of the read in step 3.

The completion of the read in step 3 ensures that the previous write was done.

For a list of the master peripherals that need this workaround, see the device-specific data manual.

### 7.2.6.4 Class of Service (COS)

The commands in the Command FIFO can be mapped to 2 classes of service namely 1 and 2. The mapping of commands to a particular class of service can be done based on the priority or the connection ID. Table 7-83 shows the master connection IDs for the various initiators in the system, as seen by the DDR controllers.

The mapping based on priority can be done by setting the appropriate values in the Priority to Class of Service Mapping register (PRI_COS_MAP). Table 7-12 shows the priority of the commands issued by the DDR2/DDR3 memory controller.

The mapping based on connection ID can be done by setting the appropriate values of connection ID and the masks in the Connection ID to Class of Service Mapping registers (CONNID_COS_1_MAP and CONNID_COS_2_MAP).

There are three connection ID and mask values that can be set for each class of service. In conjunction with the masks, each class of service can have a maximum of 144 connection IDs mapped to it. For example, a connection ID value of 0xFF, along with a mask value of 0x3, will map all connection IDs from 0xF8 to 0xFF to that particular class of service.

Each class of service has an associated latency counter (COS_COUNT). The value of this counter can be set in the Peripheral Bus Burst Priority Register (PBBPR). When the latency counter for a command expires, that is, reaches the value programmed for the class of service to which the command belongs, that command will be the one that is executed next. If there is more than one command that has expired latency counters, the command with the highest priority will be executed first. One exception to this rule is, if the oldest command in the queue has an expired PR_OLD_COUNT, that command will be executed first, irrespective of priority or class of service. This is done to prevent a continuous block effect.


Copyright © 2011–2016, Texas Instruments Incorporated

The connection ID mapping allows the same connection ID to be put in both class of service 1 and 2. Also, a transaction might belong to one class of service if viewed by connection ID and might belong to another class of service if viewed by priority. In these cases, the command will belong to both class of service. The DDR2/DDR3 memory controller will try executing the command as soon as possible, when the smaller of the two counters ( COS_COUNT_1 or COS_COUNT_2) expire.

#### 7.2.6.5 Refresh Scheduling

The DDR2/DDR3 memory controller issues autorefresh (REFR) commands to DDR2/DDR3 SDRAM devices at a rate defined in the refresh rate (REFRESH_RATE) bit field in the SDRAM refresh control register (SDRFC). A refresh interval counter is loaded with the value of the REFRESH_RATE bit field and decrements by 1 each cycle until it reaches zero. Once the interval counter reaches zero, it reloads with the value of the REFRESH_RATE bit. Each time the interval counter expires, a refresh backlog counter increments by 1. Conversely, each time the DDR2/DDR3 memory controller performs a REFR command, the backlog counter decrements by 1. This means the refresh backlog counter records the number of REFR commands the DDR2/DDR3 memory controller currently has outstanding.

The DDR2/DDR3 memory controller issues REFR commands based on the level of urgency. The level of urgency is defined below. Whenever the refresh level of urgency is reached, the DDR2/DDR3 memory controller issues a REFR command before servicing any new memory access requests. Following a REFR command, the DDR2/DDR3 memory controller waits T_RFC cycles, defined in the SDRAM timing 1 register (SDRTIM1), before rechecking the refresh urgency level.

The refresh counters do not operate when the SDRAM memory is in self-refresh mode.

**Table 7-13. Refresh Modes**

| Urgency Level | Description |
|---|---|
| Refresh May | Backlog count is greater than 0. Indicates there is a backlog of REFR commands, when the DDR2/DDR3 memory controller is not busy it will issue the REFR command. |
| Refresh Release | Backlog count is greater than 4. Indicates that the refresh backlog of REFR commands is getting high and when DDR2/DDR3 memory controller is not busy it should issue the REFR command. |
| Refresh Must | Backlog count is greater than 7. Indicates that the refresh backlog of REFR commands is getting excessive and DDR2/DDR3 memory controller should perform an auto refresh cycle before servicing any new memory access requests. |

The DDR2/DDR3 memory controller starts servicing new memory accesses after Refresh Release level is cleared. If any of the commands in the Command FIFO have class of service latency counters that are expired, the DDR2/DDR3 memory controller will not wait for Refresh Release level to be cleared but will only perform one refresh command and exit the refresh state.

### *7.2.7 Performance Counter Usage*

There are two performance counters on each memory controller that can be configured to monitor the DDR bandwidth utilization by various masters in the system. The PERF_CNT_1 and PERF_CNT_2 registers can be used to monitor or calculate the EMIF bandwidth and efficiency. These counters can be configured to count events such as total SDRAM accesses, SDRAM activates, reads, writes, etc. Each counter counts events independent of the other counter. In addition to the ability to count events, the counters can also be configured to filter the events from a particular master based on the Master Connection ID (MConnID). The events to be counted and the filter enabling can be configured using the PERF_CNT_CFG register. The filter value to be used can be configured using the PERF_CNT_SEL register. Each counter can be configured independently.

Table 7-14 shows all the events that can be counted and whether or not a filter can be applied to that particular event.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-14. Performance Counter Filter Configuration**

| CNTRn_CFG | CNTRn_MCONNID_EN | Description |
|---|---|---|
| 0x0 | 0x0 or 0x1 | Count total SDRAM accesses. |
| 0x1 | 0x0 or 0x1 | Count total SDRAM activates. |
| 0x2 | 0x0 or 0x1 | Count total reads. |
| 0x3 | 0x0 or 0x1 | Count total writes. |
| 0x4 - 0xF | 0x0 | Reserved for future use. |

**Note:** The DDR performance counters do NOT count the bytes but the events (commands from various masters). As different commands could be for different byte counts, the performance counter values may not directly correlate with the amount of data that is being transferred. However performance counters can be effectively used to monitor the DDR bus utilization by different masters.

**Example 1**

PERF_CNT_1 register needs to count all write accesses from master 0x7. To enable counting writes, the CNTR1_CFG field in the PERF_CNT_CFG register must be set to 0x3. The MCONNID1 field in the PERF_CNT_SEL must be set to 0x7. Finally, to enable filtering, the CNTR1_MCONNID_EN bit in the PERF_CNT_CFG register must be set to 0x1.

With the above configuration, PERF_CNT_1 will count every write made to the DDR from master 0x7. This will not include accesses from other masters and will not include commands other than writes.

**Example 2**

PERF_CNT_2 register needs to count total accesses to SDRAM irrespective of masters. To enable counting all SDRAM accesses, the CNTR2_CFG field in the PERF_CNT_CFG must be set to 0x0. Finally, to disable filtering, the CNTR1_MCONNID_EN bit in the PERF_CNT_CFG register must be set to 0x0.

With the above configuration, PERF_CNT_2 will count every access made to the DDR. This will include all accesses from all masters.

### 7.2.8 DDR3 Software Write/Read Leveling

The memory controller supports only software leveling. Software leveling is a procedure by which the time delays between DDR3 signals can be compensated by appropriate values programmed in the corresponding slave ratio registers of the DDR PHY. For description of these registers, see Table 7-66. Software leveling can be done independently for each data and command macro. When software leveling is used there is no need to enable the leveling feature of the external DDR3 SDRAM, which is controlled by one of its MR registers. However, during write leveling the ODT function must be on. Proper ODT values must be turned on at the external memory side by setting Rtt_Nom (A9,A6,A2) bits in external DDR3 MR1 register.

For more information about the concept of the software leveling, see http://processors.wiki.ti.com/index.php/TI814x-DDR3-Init-U-Boot.

### 7.2.9 PRCM Sequence for DDR2/DDR3 Memory Controller

The memory controller clock, reset and power are handled by the device PRCM module. Refer to the Power Reset Clock Management (PRCM) chapter for the PRCM register details.

### 7.2.10 Interrupt Support

The controller supports generation of an error interrupt if an unsupported command or a command with unsupported addressing mode is received. The controller supports only Idle, Write, Read, and WriteNonPost commands and only incrementing, wrapping, and 2-dimensional block addressing modes.

### 7.2.11 EDMA Event Support

The DDR2/DDR3 memory controller is a DMA slave peripheral and therefore does not generate EDMA events. Data read and write requests may be made directly by masters including the EDMA controller.


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.2.12 Emulation Considerations

The DDR2/DDR3 memory controller will remain fully functional during emulation halts to allow emulation access to external memory.

### 7.2.13 Power Management

This section defines the power management capabilities and requirements.

#### 7.2.13.1 Self-Refresh Mode

The DDR2/DDR3 memory controller supports self-refresh mode for low power. The memory controller automatically puts the SDRAM into self-refresh after the memory controller is idle for SR_TIM number of DDR clock cycles and the LP_MODE field is set to 2. The LP_MODE and SR_TIM fields can be programmed in the Power Management Control register(PMCR). The memory controller will complete all pending refreshes before it puts the SDRAM into self-refresh. Therefore, after the expiration of SR_TIM, the memory controller will start issuing refreshes to complete the refresh backlog, and then issue a SELF-REFRESH command to the SDRAM.

In self-refresh mode, the memory controller automatically stops the clocks DDR[x]_CLK to the SDRAM. The memory controller maintains DDR[x]_CKE low to maintain the self-refresh state. When the SDRAM is in self-refresh, the memory controller services register accesses as normal. If the LP_MODE field is set not equal to 2, or an SDRAM access is requested while it is in self-refresh, and T_CKE + 1 cycles have elapsed since the SELF-REFRESH command was issued, the memory controller will bring the SDRAM out of self-refresh. The value of T_CKE is taken from SDRAM Timing 2 register.

**Exit sequence of self-refresh mode for the DDR2 device:** The memory controller:

- Enables clocks
- Drives DDR[x]_CKE high
- Waits for T_XSNR + 1 cycles. The value of T_XSNR is taken from SDRAM Timing 2 register
- If the DDR_DISABLE_DLL bit in the SDRAM Config register is 1, issues a LOAD MODE REGISTER command to the extended mode register 1 with the DDR address lines set as follows:

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:13] | 0x0 | Reserved |
| DDR[x]_A[12] | 0x0 | Output buffer enabled |
| DDR[x]_A[11] | 0x0 | RDQS disable |
| DDR[x]_A[10] | DDR2_DDQS | Differential DQS enable value from SDRAM Config register |
| DDR[x]_A[9:7] | 0x0 | Exit OCD calibration mode |
| DDR[x]_A[6] | DDR_TERM[1] | DDR2 termination resistor value from SDRAM Config register |
| DDR[x]_A[5:3] | 0x0 | Additive latency = 0 |
| DDR[x]_A[2] | DDR_TERM[0] | DDR2 termination resistor value from SDRAM Config register |
| DDR[x]_A[1] | SDRAM_DRIVE | SDRAM drive strength from SDRAM Config register |
| DDR[x]_A[0] | 0x1 | Disable DLL |

- Starts an auto-refresh cycle in the next cycle.
- Enters its idle state and can issue any other commands except a write or a read. A write or a read will only be issued after T_XSRD + 1 clock cycles have elapsed since DDR[x]_CKE is driven high. The value of T_XSRD is taken from SDRAM Timing 2 register.

Exit sequence of self-refresh mode for DDR3 device: The memory controller:

- Enables clocks.
- Drives DDR[x]_CKE high.
- Waits for T_XSNR + 1 cycles. The value of T_XSNR is taken from SDRAM Timing 2 register.
- If the DDR_DISABLE_DLL bit in the SDRAM Config register is 1, issues a LOAD MODE REGISTER command to the extended mode register 1 with the DDR address lines set as follows:


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:13] | 0x0 | Reserved |
| DDR[x]_A[12] | 0x0 | Output buffer enabled |
| DDR[x]_A[11] | 0x0 | TDQS disable |
| DDR[x]_A[10] | 0x0 | Reserved |
| DDR[x]_A[9] | DDR_TERM[2] | DDR3 termination resistor value from SDRAM Config register |
| DDR[x]_A[8] | 0x0 | Reserved |
| DDR[x]_A[7] | 0x0 | External DDR3 write leveling feature disabled |
| DDR[x]_A[6] | DDR_TERM[1] | DDR3 termination resistor value from SDRAM Config register |
| DDR[x]_A[5] | SDRAM_DRIVE[1] | SDRAM drive strength from SDRAM Config register |
| DDR[x]_A[4:3] | 0x0 | Additive latency = 0 |
| DDR[x]_A[2] | DDR_TERM[0] | DDR3 termination resistor value from SDRAM Config register |
| DDR[x]_A[1] | SDRAM_DRIVE[0] | SDRAM drive strength from SDRAM Config register |
| DDR[x]_A[0] | 0x1 | Disable DLL |

- Starts an auto-refresh cycle in the next cycle.
- Enters its idle state and can issue any other commands except a write or a read. A write or a read will only be issued after T_XSRD + 1 clock cycles have elapsed since DDR[x]_CKE is driven high. The value of T_XSRD is taken from SDRAM Timing 2 register.

#### 7.2.13.2 Power-Down Mode

The memory controller also supports power-down mode for low power. The memory controller automatically puts the SDRAM into power-down mode after the memory controller is idle for PD_TIM number of DDR clock cycles and the LP_MODE field is set to 4. The LP_MODE and PD_TIM fields can be programmed in the Power Management Control register (PMCR). If the Refresh Must level is not reached before the entry into power-down, the memory controller will not precharge all banks before issuing the POWER-DOWN command. This will result in SDRAM entering active power-down mode.

If the Refresh Must level is reached before the entry into power-down, the memory controller will precharge all banks and issue refreshes until the Refresh Release Level is reached before issuing the POWER-DOWN command. This will result in SDRAM entering precharge power-down mode.

In power-down mode, the memory controller does not stop the clocks DDR[x]_CLK to the SDRAM. The memory controller maintains DDR[x]_CKE low to maintain the power-down state.

When the SDRAM is in power-down, the memory controller services register accesses as normal. If the LP_MODE field is set not equal to 4, or an SDRAM access is requested, or the Refresh Must level is reached while the SDRAM is in power-down, the memory controller will bring the SDRAM out of power-down.

Exit sequence of power-down mode for DDR2 and DDR3: The memory controller

- Drives DDR[x]_CKE high after T_CKE + 1 cycles have elapsed since the POWER-DOWN command was issued. The value of T_CKE is taken from SDRAM Timing 2 register.
- Waits for T_XP + 1 cycles. The value of T_XP is taken from SDRAM Timing 2 register.
- Enters its idle state and can issue any commands.

#### 7.2.13.3 EMIF PHY Clock Gating

The clock to the DDR2/DDR3 PHY can be gated off to achieve power saving. Refer to the EMIF0/1 Clock Gate Control register (EMIF_CLK_GATE).

## 7.3 DDR PHY

### *7.3.1 Functional Overview*

DDR PHY is a fully digital, flexible and advanced solution designed to get ultimate performance form the memory interface. The DDR PHY supports:


Copyright © 2011–2016, Texas Instruments Incorporated

- Bi-directional DQS
- Software write/read leveling

DDR PHY consists of three separate blocks:

- Control Block (Address and Command Block)
- Byte-wide data block
- Master DLL block

The control block handles all the address and command signals. The primary function of the PHY control block is to control the timing of the generation of all the SDRAM address and control signals such as CKE, ODT, CSN, RASN,CASN, WEN, BA (bank address) and ADDR (column/row address). The data slices or macros contain the data signal path and the slave DLLs for write data, write DQS, and read DQS. The master DLL block controls the Ratio logic functions which are used to adjust the signal timing for each signal edge. The operations of these blocks are explained in detail in the following sections.

### 7.3.2 Data Slice Block

The data slice block contains all the logic required to support read and write interface for a single memory data slice of 8 bits. It handles the DQ, DM, DQS, FIFO_WE signals. The block has data macro and multi-rank ratio logic.

### 7.3.3 Master DLL

The master DLL measures the cycle period in terms of a number of taps and passes this number through the ratio logic to the slave delay lines. The DLL is comprised of a delay line, an FSM, a clock divider, a phase detector and output filter. The FSM sends different control values to the delay line until phase detector detects transition from 0 to 1. When cross is detected, control values are provided to output filter which generates dll lock as well as final full cycle delay value. The master DLL manages on-chip variation (OCV) and continuously monitors core clock over PVT.

### 7.3.4 Ratio Logic Block

The ratio logic takes the output from the master DLL, which indicates the number of delay taps required to form a full-cycle shift, and scales it by an input ratio value in the range of 0 to 511 in units of 1/256th of a clock cycle. The result from ratio logic is the number of delay taps for slave delay line.

### 7.3.5 Command Write, Write, and Read Operation

#### 7.3.5.1 Memory Clock inversion Feature

This is a feature for pushing out the DDR[x]_CLK and the corresponding address/command by one half cycle, enabled by asserting CMD0/1/2_PHY_INVERT_CLKOUT_0. There is no need to configure this register if one of the following conditions is met:

- There is no fly-by routing on the board.
- There is no multiple fanout on DDR[x]_CLK (multiple ranks).

#### 7.3.5.2 Write Operation

On a write operation, the DDR PHY accepts signals synchronous to the core or memory clock from controller. DDR[x]_CLK/$\overline{\text{DDR[x]_CLK}}$ are generated inside the DDR PHY and sent out edge-aligned with DDR[x]_CLK/$\overline{\text{DDR[x]_CLK}}$ for DDR2 operation, and delayed from core clock by software write leveling delay for DDR3 operation. DDR[x]_D[31:0] and DDR[x]_DQM[3:0] are phase shifted by approximately 90 degrees with respect to DDR[x]_DQS[3:0].


Copyright © 2011–2016, Texas Instruments Incorporated

#### 7.3.5.3 Read Operation

For a read operation, the DDR[x]_D[31:0] signal sent by memory is edge-aligned with DDR[x]_DQS[3:0], so DDR[x]_DQS[3:0] must be shifted by PHY data slice to center-align DDR[x]_D[31:0] with DDR[x]_DQS[3:0]. This is approximately a 90-degree phase shift, with the exact shift dependent on the result of read DQS eye training in DDR3 applications, or register programming for DDR2 applications. After DDR[x]_D[31:0] is sampled by DDR[x]_DQS[3:0] and converted from DDR to SDR domain, the sampled data is re-synchronized by the core clock through a FIFO. Read data is then sent to the controller along with a data-valid signal.

## 7.4 Electrical Characteristics Control

### *7.4.1 Output Impedance Calibration*

The DDR2/DDR3 controller supports automatic output impedance calibration for DDR3. The ZQ calibration can be enabled per chip select by setting ZQ_CS0EN and ZQ_CS1EN fields in the SDRAM Output Impedance Calibration Config register. The DDR2/DDR3 memory controller supports following three types of ZQ calibration commands:

- ZQINIT – ZQ calibration command during initialization
- ZQCS – ZQ calibration short command
- ZQCL – ZQ calibration long command

For DDR3, the memory controller will automatically issue a ZQINIT command during initialization. The memory controller will wait and block any other command for (ZQ_ZQINIT_MULT+1)*(ZQ_ZQCL_MULT+1)*(T_ZQCS+1) number of clock DDR3 clock cycles every time a ZQINIT command is issued.

The memory controller periodically issues a ZQCS command every time ZQ_REFINTERVAL expires. In other words, the ZQ_REFINTERVAL defines the interval between ZQCS commands. The memory controller will wait and block any other command for (T_ZQCS+1) number of DDR3 clock cycles every time a ZQCS command is issued.

If the ZQ_SFEXITEN field is set to a 1, the memory controller will issue a ZQCL command every time it exits Self-Refresh, Active Power-Down, and Precharge Power-Down. The memory controller will wait and block any other command for (ZQ_ZQCL_MULT+1)*(T_ZQCS+1) number of clock DDR clock cycles every time a ZQCL command is issued.

If a separate calibration resistor is used per device, the ZQ calibration can be performed simultaneously over both the chip selects. To enable ZQ calibration to be performed simultaneously over both chip selects, the ZQ_DUALCALEN field must be set to a 1. If ZQ_DUALCALEN is set to a 0, the EMIF will perform ZQ calibration per chip select serially.

### *7.4.2 IO Configuration and Slew Rate Control*

On this device, programmability is given to control the output driver strength/impedance and slew rate.

Where Rext is the external resistor between device VTP pin and ground. Recommended value is 50 Ohm with 1% tolerance.

The IO output Impedance for each DDR interface is configurable through the corresponding DDRn_IO_CTRL Control Registers, described in this document.

The supported output impedance settings are shown in Table 7-15. The recommended IO buffer impedance setting is 0b011.

**Table 7-15. Output Impedance Settings**

| I2 | I1 | I0 | Output Impedance (Ron) | Ron for Rext=50 Ohms | Drive Strength (mA) for Rext=50 Ohms |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1.6*Rext | 80 | 5 |
| 0 | 0 | 1 | 1.33*Rext | 67 | 6 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-15. Output Impedance Settings (continued)**

| I2 | I1 | I0 | Output Impedance (Ron) | Ron for Rext=50 Ohms | Drive Strength (mA) for Rext=50 Ohms |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 1.14*Rext | 57 | 7 |
| 0 | 1 | 1 | Rext | 50 | 8 Recommended Setting |
| 1 | 0 | 0 | 0.88*Rext | 44 | 9 |
| 1 | 0 | 1 | 0.80*Rext | 40 | 10 |
| 1 | 1 | 0 | 0.73*Rext | 36 | 11 |
| 1 | 1 | 1 | 0.67*Rext | 33 | 12 |

To achieve optimal noise/frequency trade off, the slew rate of the output signal can also be programmed using the slew rate control bits of the corresponding DDRn_IO_CTRL Control Registers described in this document. Table 7-16 shows the programmable slew-rate settings. The recommended value is 00'b.

**Table 7-16. Programmable Slew-rate Settings**

| SR1 | SR0 | Description |
|---|---|---|
| 0 | 0 | Fastest Slew Rate<br>Recommended Setting |
| 0 | 1 | Fast Slew Rate |
| 1 | 0 | Slow Slew Rate |
| 1 | 1 | Slowest Slew Rate |

### *7.4.3 ODT Control*

The DDR2/DDR3 controller supports ODT programming. ODT means On-Die-Termination. It involves terminating the data transfer at the receive end using switchable on-chip resistances. ODT may be required for better system level signal integrity and performance. ODT strength level is a function of output driver strength. Table 7-17 shows the ODT settings.

For a data write to the DDR3 memory, the ODT is controlled through appropriate bits in MR1 register of the DDR3 memory. These are the A9, A6 and A2 bits which are configured by DDR_TERM field in the SDRCR register. When ODT is configured, the external device switches on the terminating resistances in response to the ODT output signal of the DDR2/DDR3 memory controller.

For a data read from the DDR3 memory, the ODT function is controlled in the applications processor's PHY. When it performs a memory read, the internal terminating resistances are switched on to terminate the signal at the PHY receiver if the ODT is enabled in the PHY.

The READ ODT for each DDR interface is configurable through the corresponding DDRPHYCR registers described in the register description section of this document.

**Table 7-17. ODT Settings**

| ODT1 | ODT0 | Description |
|---|---|---|
| 0 | 0 | ODT off |
| 0 | 1 | Reserved |
| 1 | 0 | Termination=Ron (eg., Rth = 50 ohms with Ron=50 ohms) |
| 1 | 1 | Termination=2*Ron (eg., Rth = 100 ohms with Ron=50 ohms) |

Refer to DDR PHY Control Register (DDRPHYCR) and DDR PHY Control Shadow register (DDRPHYCSR) for register programming.


Copyright © 2011–2016, Texas Instruments Incorporated

## 7.5 DDR2/DDR3 SDRAM Memory Initialization

### 7.5.1 DDR2 SDRAM Memory Initialization

After cold reset, if the MEMTYPE field in the SDRCR is equal to 2 and the INITREF_DIS bit in the SDRRCR is set to 0, DDR2/DDR3 controller performs DDR2 SDRAM initialization sequence as follows.

1. Drives DDR[x]_CKE low
2. After 16 SDRAM refresh rate intervals, issues a NOP command with DDR[x]_CKE held high. The SDRAM refresh rate is as defined in SDRRCR.
3. Issues PRECHARGE command with DDR[x]_A[10] held high to indicate all banks.
4. Issues a LOAD MODE REGISTER command to the extended mode register 2(DDR[x]_BA[2:0]=2h) with DDR[x]_A[x] set as follows:

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:8] | 0x0 | Reserved |
| DDR[x]_A[7] | SRT | Self-Refresh temperature range from SDRAM Refresh Control register |
| DDR[x]_A[6:4] | 0x0 | Reserved |
| DDR[x]_A[3] | 0x0 | DCC disable |
| DDR[x]_A[2:0] | PASR | Partial array self-refresh from SDRAM Refresh Control register |

5. Issues a LOAD MODE REGISTER command to the extended mode register 3(DDR[x]_BA[2:0]=3h) with DDR[x]_A[x]=0h.
6. Issues a LOAD MODE REGISTER command to the extended mode register 1(DDR[x]_BA[2:0]=0x1) with DDR[x]_A[x] set as follows:

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:13] | 0x0 | Reserved |
| DDR[x]_A[12] | 0x0 | Output buffer enabled |
| DDR[x]_A[11] | 0x0 | RDQS disable |
| DDR[x]_A[10] | DDR2_DDQS | Differential DQS enable value from SDRAM Config register |
| DDR[x]_A[9:7] | 0x0 | Exit OCD calibration mode |
| DDR[x]_A[6] | DDR_TERM[1] | DDR2 termination resistor value from SDRAM Config register |
| DDR[x]_A[5:3] | 0x0 | Additive latency = 0 |
| DDR[x]_A[2] | DDR_TERM[0] | DDR2 termination resistor value from SDRAM Config register |
| DDR[x]_A[1] | SDRAM_DRIVE | SDRAM drive strength from SDRAM Config register |
| DDR[x]_A[0] | 0x0 | Enable DLL |

7. Issues a LOAD MODE REGISTER command to the mode register (DDR[x]_BA[2:0]=0x0) with DDR[x]_A[x] set as follows:

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:13] | 0x0 | Reserved |
| DDR[x]_A[12] | 0x0 | Fast exit active power-down exit time |
| DDR[x]_A[11:9] | T_WR[2:0] | Write recovery for auto precharge from SDRAM Timing 1 register |
| DDR[x]_A[8] | 0x1 | DLL reset |
| DDR[x]_A[7] | 0x0 | Normal mode |
| DDR[x]_A[6:4] | CL[2:0] | CAS latency from SDRAM Config register |
| DDR[x]_A[3] | 0x0 | Sequential burst type |
| DDR[x]_A[2:0] | 0x3 | Burst length of 8 |

8. After 267 clock cycles, issues a PRECHARGE command with DDR[x]_A[10] held high to indicate all banks.
9. After two AUTO REFRESH commands, issues LOAD MODE REGISTER command to the mode register(DDR[x]_BA[2:0]=0h) with DDR[x]_A bits set as follows:


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:9] | Equal to step 7 | |
| DDR[x]_A[8] | 0x0 | DLL not reset |
| DDR[x]_A[7:0] | Equal to step 7 | |

10. Issues a LOAD MODE REGISTER command to the exteded mode register 1(DDR[x]_BA[2:0]=1h) with DDR[x]_A bits set as follows:

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:10] | Equal to step 6 | |
| DDR[x]_A[9:7] | 0x7 | Default OCD calibration |
| DDR[x]_A[6:0] | Equal to step 6 | |

11. Issues a LOAD MODE REGISTER command to the extended mode register 1(DDR[x]_BA[2:0]) with DDR[x]_A bits equal to step 6.
12. If the DDR_DISABLE_DLL bit in the SDRAM Config register is 1, issues a LOAD MODE REGISTER command to extended mode register 1(DDR[x]_BA[2:0]=1h) with DDR[x]_A bits, set as follows:

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:1] | Equal to step 6 | |
| DDR[x]_A[0] | 0x1 | Disable DLL |

13. The DDR2/DDR3 controller enters its idle state.

The controller also performs the initialization sequence whenever the SDRCR is written, if already INITREF_DIS bit of SDRRCR is set to 1. But in this case, the controller starts at step 3.

### 7.5.2 DDR3 SDRAM Memory Initialization

After cold reset, if the MEMTYPE field in the SDRCR is equal to 3 and the INITREF_DIS bit in the SDRRCR is set to 0, DDR2/DDR3 controller performs DDR3 SDRAM initialization sequence as follows.

1. Drives DDR[x]_CKE low.
2. After 16 SDRAM refresh rate intervals, issues a NOP command with DDR[x]_CKE held high. The SDRAM refresh rate is as defined in SDRRCR.
3. Issues a LOAD MODE REGISTER command to the extended mode register 2(DDR[x]_BA[2:0]=2h) with DDR[x]_A[x] set as follows:

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:11] | 0x0 | Reserved |
| DDR[x]_A[10:9] | DYN_ODT | Dynamic ODT value from SDRAM Config register |
| DDR[x]_A[8] | 0x0 | Reserved |
| DDR[x]_A[7] | SRT | Self-Refresh temperature range from SDRAM Refresh Control register |
| DDR[x]_A[6] | ASR | Auto self-refresh enable from SDRAM Refresh Control register |
| DDR[x]_A[5:3] | CWL | CAS write latency from SDRAM Config register |
| DDR[x]_A[2:0] | PASR | Partial array self-refresh from SDRAM Refresh Control register |

4. Issues a LOAD MODE REGISTER command to the extended mode register 3(DDR[x]_BA[2:0]=3h) with DDR[x]_A[15:0]=0h.
5. Issues a LOAD MODE REGISTER command to the extended mode register 1(DDR[x]_BA[2:0]=0x1) with DDR[x]_A[x] set as follows:

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:13] | 0x0 | Reserved |
| DDR[x]_A[12] | 0x0 | Output buffer enabled |
| DDR[x]_A[11] | 0x0 | TDQS disable |
| DDR[x]_A[10] | 0x0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[9] | DDR_TERM[2] | DDR3 termination resistor value from SDRAM Config register |
| DDR[x]_A[8] | 0x0 | Reserved |
| DDR[x]_A[7] | 0x0 | External DDR3 write leveling feature disabled |
| DDR[x]_A[6] | DDR_TERM[1] | DDR3 termination resistor value from SDRAM Config register |
| DDR[x]_A[5] | SDRAM_DRIVE[1] | SDRAM drive strength from SDRAM Config register |
| DDR[x]_A[4:3] | 0x0 | Additive latency = 0 |
| DDR[x]_A[2] | DDR_TERM[0] | DDR3 termination resistor value from SDRAM Config register |
| DDR[x]_A[1] | SDRAM_DRIVE[0] | SDRAM drive strength from SDRAM Config register |
| DDR[x]_A[0] | 0x0 | DDR_DISABLE_DLL value from SDRAM Config register |

6. Issues a LOAD MODE REGISTER command to the mode register 0(DDR[x]_BA[2:0]=0x0) with DDR[x]_A[x] set as follows:

| Bits | Value | Description |
|---|---|---|
| DDR[x]_A[15:13] | 0x0 | Reserved |
| DDR[x]_A[12] | 0x0 | Fast exit active power-down exit time |
| DDR[x]_A[11:9] | T_WR[2:0] | Write recovery for auto precharge from SDRAM Timing 1 register |
| DDR[x]_A[8] | 0x1 | DLL reset |
| DDR[x]_A[7] | 0x0 | Normal mode |
| DDR[x]_A[6:4] | CL[3:1] | CAS latency from SDRAM Config register |
| DDR[x]_A[3] | 0x0 | Sequential burst type |
| DDR[x]_A[2] | CLI[0] | CAS latency from SDRAM Config register |
| DDR[x]_A[1:0] | 0x0 | Burst length of 8 |

7. Issues a ZQCL command to start long ZQ calibration.
8. Issues an AUTO REFRESH command.
9. The memory controller enters its idle state.

The controller also performs the initialization sequence, whenever the SDRCR is written, if the INITREF_DIS bit of the SDRRCR register is set to 0x1. But in this case, the controller starts at step 3.

## 7.6 Interfacing the DDR2/DDR3 Memory Controller to DDR Memory Devices

The following sections describe the architecture of the DDR2/DDR3 memory controller and how to configure it to perform read and write operations to DDR2/DDR3 SDRAM devices. Refer the device-specific data manual to get details of supported configurations and topologies.

### *7.6.1 Configuring DDR2/DDR3 Memory Controller Registers to Meet DDR2 SDRAM Specifications*

The DDR2/DDR3 memory controller allows a high degree of programmability for shaping DDR2 accesses. This provides the DDR2/DDR3 memory controller with the flexibility to interface with a variety of DDR2 devices. By programming the SDRAM Configuration Register (SDRCR), SDRAM Refresh Control Register (SDRRCR), SDRAM Timing 1 Register (SDRTIM1), SDRAM Timing 2 Register (SDRTIM2), and SDRAM Timing 3 Register (SDRTIM3), the DDR2/DDR3 memory controller can be configured to meet the data sheet specification for JESD79-2E compliant DDR2 SDRAM devices.

As an example, the following sections describe how to configure each of these registers for access to two 1Gb, 16-bit wide DDR2 SDRAM devices, where each device has the following configuration:

- SDRAM Clock Frequency : 400 MHz
- SDRAM Data Rate : 800 MHz
- Number of banks: 8
- Page size: 1024 words


Copyright © 2011–2016, Texas Instruments Incorporated

- CAS latency: 5

#### 7.6.1.1 Programming the SDRAM Configuration Register (SDRCR)

The SDRAM configuration register (SDRCR) contains register fields that configure the DDR2 memory controller to match the data bus width, CAS latency, number of banks, and page size of the attached DDR2 memory. Table 7-18 shows the resulting SDRCR configuration.

**Table 7-18. SDRCR Configuration**

| Field | Value | Function Selection |
|---|---|---|
| SDRAM_TYPE | 2h | To select DDR2 Memory Type |
| DDR_TERM | 1h | To select 75 ohm termination register |
| DDR2_DDQS | 1h | To select Differential Ended DQS |
| DDR_DISABLE_DLL | 0h | Enable DLL for Normal Operation |
| SDRAM_DRIVE | 0h | To select a drive strength |
| NARROW_MODE | 0h | To configure the DDR2 memory controller for a 32-bit data bus width |
| CL | 5h | To select CAS latency of 5 |
| IBANK | 3h | To select 8 internal DDR2 banks |
| EBANK | 0h | To select single chip select, CS0 |
| PAGESIZE | 2h | To select 1024-word page size |

#### 7.6.1.2 Programming the SDRAM Refresh Control Register (SDRRCR)

The SDRAM refresh control register (SDRRCR) configures the DDR2 memory controller to meet the refresh requirements of the attached DDR2 device. SDRRCR also allows the DDR2 memory controller to enter and exit self refresh. In this example, we assume that the DDR2 memory controller is in self-refresh mode.

The RR field in SDRRCR is defined as the rate at which the attached DDR2 device is refreshed in DDR2 cycles. The value of this field may be calculated using the following equation:

*Refresh_Rate = DDR2 clock frequency (in MHz) x DDR2 memory fresh rate (in $\mu$s)*

Table 7-19 displays the DDR2-800 refresh rate specification.

**Table 7-19. DDR2-800 Refresh Rate Specification**

| Symbol | Value | Description |
|---|---|---|
| tREF | 7.8µs | Average Periodic Refresh Interval |

Therefore, the value for the Refresh Rate (RR) can be calculated as follows:

*RR = 400MHz x 7.8µs = 3120=C30h*

Table 7-20 shows the RR calculated value.

**Table 7-20. Refresh Rate Calculated Value**

| Filed | Value | Function Selection |
|---|---|---|
| INITREF_DIS | 0h | To enable DDR3 Initialization and Refreshes |
| RR | C30h | Set to C30h DDR2 clock cycles to meet the DDR2 memory refresh rate requirement |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.6.1.3 Configuring SDRAM Timing Registers (SDRTIM1, SDRTIM2 and SDTIM3)

The SDRAM timing 1 register (SDRTIM1), SDRAM timing 2 register (SDRTIM2), and SDRAM timing 3 register (SDRTIM3) configure the DDR2 memory controller to meet the data sheet timing parameters of the attached DDR2 device. Each field in SDRTIM1, SDRTIM2 and SDTIM3 register corresponds to a timing parameter in the DDR2 data sheet specification. Table 7-21, Table 7-22, and Table 7-23 display the register field name and corresponding DDR2 data sheet parameter name, along with the data sheet value. These tables also provide a formula to calculate the register field value and displays the resulting calculation. Each of the equations include a minus 1 because the register fields are defined in terms of DDR2|| clock cycles minus 1.

**Table 7-21. SDRTIM1 Configuration**

| Register Field Name | DDR3 SDRAM Data Sheet Parameter Name | Description | Example Data Sheet Value | Unit | Formula (Register Field Must be ≥) | Field Value |
|---|---|---|---|---|---|---|
| T_RP | tRP | Precharge command to refresh or activate command | 15 | ns | (tRP x fDDR[x]_CLK)-1 | 5 |
| T_RCD | tRCD | Activate command to read/write command | 15 | ns | (tRCD x fDDR[x]_CLK)-1 | 5 |
| T_WR | tWR | Write recovery time | 15 | ns | (tWR x fDDR[x]_CLK)-1 | 5 |
| T_RAS | tRAS | Active to precharge command | 40 | ns | (tRAS x fDDR[x]_CLK)-1 | 15 |
| T_RC | tRC | Activate to Activate command in the same bank | 55 | ns | (tRC x fDDR[x]_CLK)-1 | 21 |
| T_RRD | tRRD/tFAW | Activate to Activate command in a different bank | 50 (For x16 mode) | ns | (tFAW x fDDR[x]_CLK/4)-1 for 8 bank DDR3 memory | 4 |
| T_WTR | tWTR | Write to read command delay | 7.5 | ns | (tWTR x fDDR[x]_CLK)-1 | 2 |

**Table 7-22. SDRTIM2 Configuration**

| Register Field Name | DDR3 SDRAM Data Sheet Parameter Name | Description | Example Data Sheet Value | Unit | Formula(Register Field Must be ≥) | Field Value |
|---|---|---|---|---|---|---|
| T_XP | tXP | Exit precharge power-down to any non-read command | 2 | ns | (tXP x fDDR[x]_CLK)-1 | 1 |
| T_XSNR | tXSNR | Exit self-refresh to a non-read command | 137.5 | ns | (tXSNR x fDDR[x]_CLK)-1 | 54 |
| T_XSRD | tXSRD | Exit self-refresh to a read command | 200 | tCK | (tXSRD -1) | 199 |
| T_RTP | tRTP | Read to precharge command delay | 7.5 | ns | (tRTP x fDDR[x]_CLK)-1 | 2 |
| T_CKE | tCKE | CKE minimum pulse width (high and low pulse width) | 3 | tCK | (tCKE-1) | 2 |

**Table 7-23. SDRTIM3 Configuration**

| Register Field Name | DDR3 SDRAM Data Sheet Parameter Name | Description | Example Data Sheet Value | Unit | Formula(Register Field Must be ≥) | Field Value |
|---|---|---|---|---|---|---|
| T_RFC | tRFC | Auto-refresh to active/auto-refresh command time | 127.5 | ns | (tRFC x fDDR[x]_CLK)-1 | 50 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-23. SDRTIM3 Configuration (continued)**

| Register Field Name | DDR3 SDRAM Data Sheet Parameter Name | Description | Example Data Sheet Value | Unit | Formula(Register Field Must be ≥) | Field Value |
|---|---|---|---|---|---|---|
| T_RASMAX | | Maximum number of refresh rate intervals from Active to Precharge command | | | This field must be programmed to 0xF for DDR2/DDR3 | 15 |

#### 7.6.1.4 Configuring the DDR PHY Control Register

The DDR2/DDR3 PHY control register (DDRPHYCR) contains a read latency (RL) and ODT fields that need to be configured properly for the DDR2 memory controller to register data properly.

**Table 7-24. DDR2/DDR3 Memory Controller Control Register (DDRPHYCR)**

| Field | Value | Description |
|---|---|---|
| DYN_PWRDN_EN | 1 | When set to 1, dynamic power down is enabled. IO receiver is powered down when not doing any reads |
| IDLE_LOCAL_ODT | 0x0 | ODT off (recommended value) |
| RD_LOCAL_ODT | 0x2 | Read ODT (recommended value) |
| READ_LATENCY | 8 | CL + 4 cycles selected. Should be CL + 1 <= RL <= CL + 7 cycles. |

The value to write to register DDRPHYCR is 0x00170208.

#### 7.6.1.5 Configuring the DDR IO Control Register(DDRn_IO_CTRL)

The DDR2/DDR3 IO_CTRL register has to be configured with appropriate values to select the IO output impedance and Slew rates. A high precision 50 Ohm resistor should be connected between VTP pin and ground in order to achieve the output impedance as mentioned below.

**Table 7-25. Configuring DDRn_IO_CTRL**

| Field | Value | Function Selection |
|---|---|---|
| CS_SLEW | 0h | Fastest slew rate for chip select |
| CS_Impedance | 3h | Chip select signal impedance = 50 Ohms (=8mA Drive Strength) |
| CMD_SLEW | 0h | Fastest slew rate for command lines |
| CMD_Impedance | 3h | Command lines impedance = 50 Ohms (=8mA drive strength) |
| DATA_SLEW | 0h | Fastest slew rate for data pins |
| DATA_Impedance | 3h | Data impedance = 50 Ohms (=8mA Drive Strength) |

The value to write to DDRn_IO_CTRL register is 0x00030303.

### *7.6.2 Configuring DDR2/DDR3 Memory Controller Registers to Meet DDR3 SDRAM Specifications*

This section describes the memory controller setup for DDR3. The following sections describe how to configure each of the registers for access to four 1Gb, 8-bit wide DDR3 SDRAM devices, where each device has the following configuration:

- SDRAM Clock Frequency : 400 MHz
- SDRAM Data Rate : 800 MHz
- Number of banks: 8
- Page size: 1024 words
- CAS latency: CL = 6


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.6.2.1 Programming the SDRAM Configuration Register (SDRCR)

The SDRAM configuration register (SDRCR) contains register fields that configure the DDR3 memory controller to match the data bus width, CAS latency, number of banks, and page size of the attached DDR3 memory. Table 7-26 shows the resulting SDRCR configuration.

**Table 7-26. SDRCR Configuration**

| Field | Value | Function Selection |
|---|---|---|
| SDRAM_TYPE | 3h | To select DDR3 Memory Type |
| DDR_TERM | 1h | To select RZQ/4 |
| DDR2_DDQS | 1h | To select Differential Ended DQS (DDR3 only supports differential ended) |
| DYN_ODT | 2h | DDR3 Dynamic ODT (RZQ/2) |
| DDR_DISABLE_DLL | 0h | 0h Enable DLL for Normal Operation |
| SDRAM_DRIVE | 0h | To select a drive strength of RZQ/6 |
| CWL | 0h | DDR3 CAS Write Latency of 5 |
| NARROW_MODE | 0h | To configure the DDR3 memory controller for a 32-bit data bus width |
| CL | 4h | To select CAS latency of 6 |
| IBANK | 3h | To select 8 internal DDR3 banks |
| EBANK | 0h | To select single chip select, CS0 |
| PAGESIZE | 2h | To select 1024-word page size |

The value to write to register SDRCR is 0x61C01032.

### 7.6.2.2 Programming the SDRAM Refresh Control Register (SDRRCR)

The SDRAM refresh control register (SDRRCR) configures the DDR3 memory controller to meet the refresh requirements of the attached DDR3 device. SDRRCR also allows the DDR3 memory controller to enter and exit self refresh. In this example, we assume that the DDR3 memory controller is in self-refresh mode.

The RR field in SDRRCR is defined as the rate at which the attached DDR3 device is refreshed in DDR3 cycles. The value of this field may be calculated using the following equation:

*Refresh_Rate = DDR2/DDR3 clock frequency (in MHz) x DDR2/DDR3 fresh rate (in μs)*

Table 7-27 displays the DDR3-800 refresh rate specification.

**Table 7-27. Periodic Refresh Interval Specification**

| Symbol | Value | Description |
|---|---|---|
| tREF | 7.8μs | Average Periodic Refresh Interval |

RR = 400MHz x 7.8μs = 3120 = C30h

Table 7-28 shows the SDRRCR configuration.

**Table 7-28. SDRRCR Configuration**

| Field | Value | Function Selection |
|---|---|---|
| INITREF_DIS | 0h | Enable SDRAM initialization and Refreshes |
| SRT | 0h | Normal Operating Temperature range |
| ASR | 1h | Auto Self Refresh Enable |
| PASR | 0h | Full Array Refresh |
| REFRESH_RATE | C30h | Refresh Rate |


Copyright © 2011–2016, Texas Instruments Incorporated

The value to write to register SDRRCR register is 0x10000C30.


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.6.2.3 Configuring SDRAM Timing Registers (SDRTIM1, SDRTIM2, SDTRIM3)

The SDRAM timing 1 register (SDRTIM1), SDRAM timing 2 register (SDRTIM2), and SDRAM timing 3 register (SDRTIM3) configure the DDR3 memory controller to meet the data sheet timing parameters of the attached DDR3 device. Each field in SDRTIM1, SDRTIM2 and SDTIM3 corresponds to a timing parameter in the DDR3 data sheet specification. Table 7-29, Table 7-30, and Table 7-31 display the register field name and corresponding DDR3 data sheet parameter name along with the data sheet value. These tables also provide a formula to calculate the register field value and displays the resulting calculation. Each of the equations include a minus 1 because the register fields are defined in terms of DDR3 clock cycles minus 1.

**Table 7-29. SDRTIM1 Configuration**

| Register Field Name | DDR3 SDRAM Datasheet Parameter Name | Description | Example Datasheet Value | Unit | Formula (Register Field must be ≥) | Field Value |
|---|---|---|---|---|---|---|
| T_RP | tRP | Precharge command to refresh or activate command | 13.75 | ns | (tRP/DDR[X]_CLK period)-1 | 5 |
| T_RCD | tRCD | Activate command to read/write command | 13.75 | ns | (tRCD/DDR[X]_CLK period)-1 | 5 |
| T_WR | tWR | Write recovery time | 15 | ns | (tWR/DDR[X]_CLK period)-1 | 5 |
| T_RAS | tRAS | Active to precharge command | 35 | ns | (tRAS/DDR[X]_CLK period)-1 | 13 |
| T_RC | tRC | Activate to Activate command in the same bank | 48.75 | ns | (tRC/DDR[X]_CLK period)-1 | 19 |
| T_RRD | tRRD or tFAW | Activate to Activate command in a different bank | 35 | ns | (tFAW/(4*DDR[X]_CLK period))-1 | 3 |
| T_WTR | tWTR | Write to read command delay | 10 | ns | (tWTR/DDR[X]_CLK period)-1 | 3 |

The value to write to register SDRTIM1 is 0x0AAAD4DB.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-30. SDRTIM2 Configuration**

| Register Field Name | DDR3 SDRAM Datasheet Parameter Name | Description | Example Datasheet Value | Unit | Formula (Register Field must be ≥) | Field Value |
|---|---|---|---|---|---|---|
| T_XP | tXP | Exit precharge power-down to any non-read command | 3 | tCK | tXP - 1 | 2 |
| T_XSNR | tXSNR | Exit self-refresh to a non-read command | 120 | ns | (tXSNR/DDR[X]_CLK period)-1 | 47 |
| T_XSRD | tXSRD | Exit self-refresh to a read command | 512 | tCK | tXSRD - 1 | 511 |
| T_RTP | tRTP | Read to precharge command delay | 10 | ns | (tRTP/DDR[X]_CLK period)-1 | 3 |
| T_CKE | tCKE | CKE minimum pulse width (high and low pulse width) | 3 | tCK | tCKE - 1 | 2 |

The value to write to register SDRTIM2 is 0x202F7FDA.

**Table 7-31. SDRTIM3 Configuration**

| Register Field Name | DDR3 SDRAM Datasheet Parameter Name | Description | Example Datasheet Value | Unit | Formula (Register Field must be ≥) | Field Value |
|---|---|---|---|---|---|---|
| T_PDLL_UL | | Number of cycles for DLL to unlock | | tCK | This field must be written with 5 | 5 |
| T_CSTA | tCSTA | Write-to-write or read-to-read data phase | 1 | tCK | tCSTA - 1 | 0 |
| T_ZQCS | tZQCS | Number of cycles for ZQCS command | 64 | tCK | tZQCS - 1 | 63 |
| T_RFC | tRFC | Auto-refresh to active/auto-refresh command time | 110 | ns | (tRFC/DDR[X]_CLK period)-1 | 43 |
| T_RAS_MAX | | Maximum number of refresh rate intervals from Active to Precharge command | | | | 15 |

The value to write to register SDRTIM3 is 0x501F82BF.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 7.6.2.4 Configuring the DDR PHY Control Register

The DDR2/DDR3 PHY control register (DDRPHYCR) contains a read latency (RL) and ODT fields that needs to be configured properly for the DDR3 memory controller to register data properly.

**Table 7-32. PHY Control Register (DDRPHYCR)**

| Field | Value | Function Selection |
|---|---|---|
| DYN_PWRDN_EN | 1h | Power down IO receiver when not performing a read |
| RDEYE_LVL_DIS | 1h | Disable read eye auto-leveling |
| GATE_LVL_DIS | 1h | Disable read gate auto-leveling |
| WR_LVL_DIS | 1h | Disable write auto-leveling |
| PHY_RST | 0h | PHY in functional mode |
| IDLE_LOCAL_ODT | 0h | ODT off |
| RD_LOCAL_ODT | 2h | 50 Ohms Read ODT termination |
| READ_LATENCY | 9h | CL + 4 cycles selected. Should be CL + 1 <= RL <= CL + 7 cycles. |

The value to write to register DDRPHYCR is 0x00170209.

#### 7.6.2.5 Configuring the ZQCR Control Register

The DDR3 ZQCR register has to be configured with appropriate values to enable periodic ZQ calibration.

The ZQ Refresh interval can be calculated with the following equation:

*ZQ Refresh Interval = 0.5%/( (Tsens × Tdriftrate) + (Vsens × Vdriftrate)).*

As per the DDR3 specification, the maximum values for Tsens and Vsens are as follows:

Tsens = 1.5%/°C
Vsens = 0.15%/mV

The temperature and voltage drift rates depend on the system environment. For illustrative purpose the following values are assumed.

Tdriftrate = 1.2°C/s
Vdriftrate = 10 mV/s

ZQ calibration Interval = 0.5%/((1.5%/°C
x1.2°C/s)+(0.15%/mV x10 mV/s) = 151.515ms approx.

Number of Refresh periods to make up for the ZQ calibration Interval = 151.515ms/7.8µs = 19425= 0x4BE1.

**Table 7-33. ZQCR Configuration**

| Field | Value | Function Selection |
|---|---|---|
| ZQ_CS1EN | 0h | Disable ZQ calibration for CS1 |
| ZQ_CS0EN | 1h | Enable ZQ calibration for CS0 |
| ZQ_DUALCALEN | 0h | Disable dual calibration |
| ZQ_SEFXITEN | 1h | ZQCL enabled after a Self Refresh Exit |
| ZQ_ZQINIT_MULT | 1h | number of ZQCL intervals that make up a ZQINIT interval - 1 |
| ZQ_ZQCL_MULT | 3h | Number of ZQCS intervals that make up a ZQCL interval - 1 |
| ZQ_REFINTERVAL | 4BE1h | Number of refresh intervals to make up the delay between ZQ commands |

The value to write to ZQCR register is 0x50074BE1.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 7.6.2.6 Configuring the DR IO Control Register (DDR n_IO_CTRL)

The DDR2/DDR3 IO_CTRL register has to be configured with appropriate values to select the IO output impedance and Slew rates. A high precision 50 Ohm resistor should be connected between VTP pin and ground in order to achieve the output impedance as mentioned below.

**Table 7-34. DDR2/DDR3 IO_CTRL Configuration**

| Field | Value | Function Selection |
|---|---|---|
| CS_SLEW | 0h | Fastest Slew Rate for Chip Select |
| CS_IMPEDENCE | 3h | Chip Select signal impedance = 50 Ohms (=8mA Drive Strength) |
| CMD_SLEW | 0h | Fastest Slew Rate for Command lines |
| CMD_IMPEDENCE | 3h | Command lines impedance = 50 Ohms (=8mA Drive Strength) |
| DATA_SLEW | 0h | Fastest Slew Rate for Data pins |
| DATA_IMPEDENCE | 3h | Data impedance = 50 Ohms (=8mA Drive Strength) |

The value to write to DDRn_IO_CTRL register is 0x00030303.


Copyright © 2011–2016, Texas Instruments Incorporated

## 7.7 DDR2/DDR3 SDRAM Device Configuration Steps

The following are recommended steps to initialize connected SDRAM devices:

1. Configure the DDR ADPLL to generate the desired frequency of operation.(Refer to the device's Technical Reference Manual for ADPLL configuration details).
2. Perform the PRCM sequences for DDR controllers and DMM (refer to the device's Technical Reference Manual for DMM details)
3. Configure the DDR related Control module registers (IO impedance, Slew Rate, and more).
4. Perform VTP0/1 calibration.
5. Configure DDR0/1 PHY registers with calculated ratio values based on the board routing delays (see DDR2/DDR3 PHY register for details). Note that in case of a DDR2-based system, the default PHY register configurations need not be changed if the DDR PCB layout guidelines in the device-specific data manual are followed strictly.
6. Program LISA/PAT mapping registers of DMM for the desired address mapping, interleaving/non-interleaving configurations, etc.
7. Program DDR memory controller MMRs (For configuring PHY read latency, device type, various timing parameters, addressing modes etc. Please note that all MMRs are initialized with proper configurations before clearing the INITREF_DIS bit, as the same triggers the SDRAM device initialization procedure. Refer to DDR2/DDR3 SDRAM Memory initialization section for more detailed register configuration procedure.


Copyright © 2011–2016, Texas Instruments Incorporated

## 7.8 DDR2/DDR3 Configuration Registers

### 7.8.1 DDR2/DDR3 Memory Controller Registers

Table 7-35 lists the memory-mapped registers for the DDR2/DDR3 memory controller.

**Table 7-35. DDR2/DDR3 Memory Controller Registers**

| Address Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 4h | SDRSTAT | SDRAM Status Register | Section 7.8.1.1 |
| 8h | SDRCR | SDRAM Configuration Register | Section 7.8.1.2 |
| Ch | SDRCR2 | SDRAM Configuration Register 2 | Section 7.8.1.3 |
| 10h | SDRRCR | SDRAM Refresh Control Register | Section 7.8.1.4 |
| 14h | SDRRCSR | SDRAM Refresh Control Shadow Register | Section 7.8.1.5 |
| 18h | SDRTIM1 | SDRAM Timing 1 Register | Section 7.8.1.6 |
| 1Ch | SDRTIM1SR | SDRAM Timing 1 Shadow Register | Section 7.8.1.7 |
| 20h | SDRTIM2 | SDRAM Timing 2 Register | Section 7.8.1.8 |
| 24h | SDRTIM2SR | SDRAM Timing 2 Shadow Register | Section 7.8.1.9 |
| 28h | SDRTIM3 | SDRAM Timing 3 Register | Section 7.8.1.10 |
| 2Ch | SDRTIM3SR | SDRAM Timing 3 Shadow Register | Section 7.8.1.11 |
| 38h | PMCR | Power Management Control Register | Section 7.8.1.12 |
| 3Ch | PMCSR | Power Management Control Shadow Register | Section 7.8.1.13 |
| 54h | PBBPR | Peripheral Bus Burst Priority Register | Section 7.8.1.14 |
| 80h | PERF_CNT_1 | Performance Counter 1 Register | Section 7.8.1.15 |
| 84h | PERF_CNT_2 | Performance Counter 2 Register | Section 7.8.1.16 |
| 88h | PERF_CNT_CFG | Performance Counter Config Register | Section 7.8.1.17 |
| 8Ch | PERF_CNT_SEL | Performance Counter Master Region Select Register | Section 7.8.1.18 |
| A0h | EOI | End of Interrupt Register | Section 25.9.1.3 |
| A4h | SOIRSR | System OCP Interrupt Raw Status Register | Section 7.8.1.20 |
| ACh | SOISR | System OCP Interrupt Status Register | Section 7.8.1.21 |
| B4h | SOIESR | System OCP Interrupt Enable Set Register | Section 7.8.1.22 |
| BCh | SOIECR | System OCP Interrupt Enable Clear Register | Section 7.8.1.23 |
| C8h | ZQCR | SDRAM Output Impedance Calibration Configuration Register | Section 7.8.1.24 |
| E4h | DDRPHYCR | DDR PHY Control Register | Section 7.8.1.25 |
| E8h | DDRPHYCSR | DDR PHY Control Shadow Register | Section 7.8.1.26 |
| 100h | PRI_COS_MAP | Priority to Class of Service Mapping Register | Section 7.8.1.27 |
| 104h | CONNID_COS_1_MAP | Connection ID to Class of Service 1 Mapping Register | Section 7.8.1.28 |
| 108h | CONNID_COS_2_MAP | Connection ID to Class of Service 2 Mapping Register | Section 7.8.1.29 |
| 120h | RD_WR_EXEC_THRSH | Read Write Execution Threshold Register | Section 7.8.1.30 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.1 SDRAM Status Register (SDRSTAT)

The SDRAM status register (SDRSTAT) is shown and described in the figure and table below.

**Figure 7-5. SDRAM Status Register (SDRSTAT)**

| 31 | 30 | 29 | 28 24 |
|---|---|---|---|
| BE | DUAL_CLK_MODE | FAST_INIT | Reserved |
| R-0 | R-0 | R-0 | R-0 |

| 23 16 |
|---|
| Reserved |
| R-0 |

| 15 8 |
|---|
| Reserved |
| R-0 |

| 7 3 | 2 | 1 0 |
|---|---|---|
| Reserved | PHY_DLL_READY | Reserved |
| R--0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-36. SDRAM Status Register (SDRSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | BE | | Defines whether the memory controller is in big or little endian mode. |
| | | 0 | Little Endian |
| | | 1 | |
| 30 | DUAL_CLK_MODE | | Dual Clock mode. Defines whether the ocp_clk and m_clk are asynchronous. |
| | | 0 | Not sync |
| | | 1 | Sync |
| 29 | FAST_INIT | | Fast Init. Defines whether the memory controller fast initialization mode has been enabled. |
| | | 0 | Disabled |
| | | 1 | Enabled |
| 28-3 | Reserved | | Reserved |
| 2 | PHY_DLL_READY | | DDR PHY Ready. Reflects the value active high that defines whether the DDR PHY is ready for normal operation. |
| | | 0 | |
| | | 1 | |
| 1-0 | Reserved | | Reserved |

### 7.8.1.2 SDRAM Configuration Register(SDRCR)

The SDRAM configuration register (SDRCR) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 7-6. SDRAM Configuration Register (SDRCR)**

| 31–29 | 28–27 | 26–24 | 23 | 22–21 | 20 | 19–18 | 17–16 |
|---|---|---|---|---|---|---|---|
| SDRAM_TYPE | IBANK_POS | DDR_TERM | DDR2_DDQS | DYN_ODT | DDR_DISABLE_DLL | SDRAM_DRIVE | CWL |
| R/W-2 | R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14–13 | 10–9 | 8 |
|---|---|---|---|
| NARROW_MODE | CL | ROWSIZE | |
| R/W-0 | R/W-5 | R/W-0 | |

| 7 | 6–4 | 3 | 2–0 |
|---|---|---|---|
| ROWSIZE | IBANK | EBANK | PAGESIZE |
| R/W-0 | R/W-2 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-37. SDRAM Configuration Register (SDRCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-29 | SDRAM_TYPE | | SDRAM Type Selection |
| | | 0 | Reserved |
| | | 1h | Reserved |
| | | 2h | DDR2 |
| | | 3h | DDR3 |
| | | | All other values are reserved |
| 28-27 | IBANK_POS | | Internal Bank position selection |
| | | 0 | Set 0,1,2,3. Refer to Table 7-4 to Table 7-11 for more details |
| | | 1 | |
| 26-24 | DDR_TERM | | DDR2 & DDR3 termination resistor value |
| | | 0 | Disable |
| | | | **For DDR2 Only :** |
| | | 1h | 75 Ohm (Recommended) |
| | | 2h | 150 Ohm |
| | | 3h | 50 Ohm |
| | | | **For DDR3 Only** |
| | | 1h | RZQ/4 (Recommended value of external resistor RZQ is 240 ohm) |
| | | 2h | Reserved |
| | | 3h | Reserved |
| | | 4h | Reserved |
| | | 5h | Reserved |
| 23 | DDR2_DDQS | | DDR2 Differential DQS Enable |
| | | 0 | Single Ended DQS |
| | | 1 | Differential Ended DQS<br>**Note: DDR3 supports only Differential DQS** |
| 22-21 | DYN_ODT | | Controls the Dynamic ODT on the DDR3 device side. Refer to DDRPHYCR for configuring the ODT on the DDR PHY side. |
| | | 0 | OFF |
| | | 1h | RZQ/4 |
| | | 2h | RZQ/2 (Recommended value of external resistor RZQ is 240 ohm) |
| 20 | DDR_DISABLE_DLL | | Disable DLL Select |
| | | 0 | Enable DLL for Normal Operation. This bit must be set to low for normal operation. |
| | | 1 | Disable DLL |
| 19-18 | SDRAM_DRIVE | | SDRAM Drive strength. Configures output drive impedance control value of the DDR2/DDR3 SDRAM memory |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-37. SDRAM Configuration Register (SDRCR) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| | | | **For DDR2** |
| | | 0 | Full drive strength |
| | | 1h | Reduced drive strength |
| | | | **For DDR3** |
| | | 0 | RZQ/6 drive strength |
| | | 1h | RZQ/7 drive strength |
| 17-16 | CWL | | DDR3 CAS Write Latency |
| | | 0 | CWL of 5 |
| | | 1h | CWL of 6 |
| | | 2h | CWL of 7 |
| | | 3h | CWL of 8 |
| | | | **For DDR2: NA** |
| 15-14 | NARROW_MODE | | Data Bus Width |
| | | 0 | 32 bits |
| | | 1 | 16 bits |
| 13-10 | CL | | CAS Latency |
| | | | **For DDR2 CAS Latency** |
| | | 2 | CAS latency of 2 |
| | | 3 | CAS latency of 3 |
| | | 4 | CAS latency of 4 |
| | | 5 | CAS latency of 5 |
| | | 6 | CAS latency of 6 |
| | | 7 | CAS latency of 7 |
| | | | **For DDR3 CAS Latency** (1) |
| | | 2 | CAS latency of 5 |
| | | 4 | CAS latency of 6 |
| | | 6 | CAS latency of 7 |
| | | 8 | CAS latency of 8 |
| | | 10 | CAS latency of 9 |
| | | 12 | CAS latency of 10 |
| | | 14 | CAS latency of 11 |
| 9-7 | ROWSIZE | | Row Size Selection. Don't care if both EBANK_POS and IBANK_POS are 0. |
| | | 0 | 9 Rows bits |
| | | 1h | 10 Rows bits |
| | | 2h | 11 Rows bits |
| | | 3h | 12 Rows bits |
| | | 4h | 13 Rows bits |
| | | 5h | 14 Rows bits |
| | | 6h | 15 Rows bits |
| | | 7h | Reserved |
| 6-4 | IBANK | | Bank Selection. Defines number of internal banks on the external DDR2/DDR3 memory. |
| | | 0 | 1 Bank |
| | | 1h | 2 Banks |
| | | 2h | 4 Banks |
| | | 3h | 8 Banks |
| | | 6-7h | Reserved |

(1) DDR3 devices typically specify Max and Min allowable CAS latency. EMIF must be set to an acceptable value within that range.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-37. SDRAM Configuration Register (SDRCR) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 3 | EBANK | | Chip Selection. Defines whether SDRAM accesses will use 1 or 2 chip selects |
| | | 0 | CS0 only |
| | | 1 | CS0, CS1 |
| 2-0 | PAGESIZE | | Page Size Selection. Defines the page size of each page of the external DDR2/DDR3 memory |
| | | 0 | 256 word page requiring 8 column address bits |
| | | 1h | 512 word page requiring 9 column address bits |
| | | 2h | 1024 word page requiring 10 column address bits |
| | | 3h | 2048 word page requiring 11 column address bits |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.3 SDRAM Configuration Register 2 (SDRCR2)

The SDRAM Configuration Register 2 (SDRCR2) is shown in the figure and table below.

**Figure 7-7. SDRAM Configuration Register 2 (SDRCR2)**

| 31 | 28 | 27 | 26 | 16 |
|---|---|---|---|---|
| Reserved | | EBANK_POS | Reserved | |
| R-0 | | R/W-0 | R-0 | |

| 15 | 0 |
|---|---|
| Reserved | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-38. SDRAM Configuration Register 2 (SDRCR2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | Reserved | 0 | Reserved |
| 27 | EBANK_POS | | External Bank Position |
| | | 0 | Assign external bank address bits from lower OCP address bits |
| | | 1 | Assign external bank address bits from higher OCP address bits |
| 26-0 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.4 SDRAM Refresh Control Register (SDRRCR)

The SDRAM Refresh Control Register (SDRRCR) is described in the figure and table below.

**Figure 7-8. SDRAM Refresh Control Register (SDRRCR)**

| 31 | 30 | 29 | 28 | 27 | 26 24 | 23 16 |
|---|---|---|---|---|---|---|
| INITREF_DIS | Rsvd | SRT | ASR | Rsvd | PASR | Reserved |
| R/W-1 | R/W-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R-0 |

| 15 0 |
|---|
| Refresh_Rate |
| R/W-1388h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-39. SDRAM Refresh Control Register (SDRRCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | INITREF_DIS | | Initialization and Refresh Disable |
| | | 0 | Enable SDRAM Initialization and Refreshes |
| | | 1 | Disable SDRAM Initialization and Refreshes but carries out SDRAM Write/Read transactions. |
| 30 | Reserved | 0h | Reserved |
| 29 | SRT | | DDR2 and DDR3 Self Refresh Temperature Range |
| | | 0 | Normal Operating Temperature range |
| | | 1 | Extended Operating Temperature range |
| | | | For DDR3, this bit must be set to '0' if ASR is set to 1. A write to this field will cause the DDR2/DDR3 to start the initialization sequence |
| 28 | ASR | | DDR3 Auto Self Refresh Enable |
| | | 0 | Manual Self Refresh reference indicated by SRT field |
| | | 1 | Auto Self Refresh Enable |
| | | | A write to this field causes the DDR3 to start the SDRAM initialization sequence. |
| 27 | Reserved | | Reserved |
| 26-24 | PASR | | Partial Array Self Refresh. These bits determine what portion of the array will be refreshed with the external memory is in partial array self refresh mode. These bits gets loaded into the EMR of the external DDR2/DDR3 memory during initialization. If the external memory does not support Partial Array Self Refresh, this field should be set to 0h. A write to this field will cause the EMIF to start the SDRAM initialization sequence. |
| | | | **For DDR2/DDR3 set** |
| | | 0h | Full Array |
| | | 1h or 5h | 1/2 Array |
| | | 2h or 6h | 1/4 Array |
| | | 3h or 7h | 1/8 Array |
| | | 4h | 3/4 Array |
| | | | to be refreshed. A write to this field will cause the memory controller to start the SDRAM initialization sequence. |
| 23-16 | Reserved | | Reserved |
| 15-0 | REFRESH_RATE | 1388h | Refresh rate. Defines the rate at which the attached DDR2/DDR3 devices are refreshed. The value of this field is calculated with the following equation: |
| | | | *Refresh_Rate = DDR2/DDR3 clock frequency (in MHz) x DDR2/DDR3 fresh rate (in µs)* |
| | | | Where DDR2/DDR3 refresh rate is derived from DDR2/DDR3 data sheet |

### 7.8.1.5 SDRAM Refresh Control Shadow Register (SDRRCSR)

The SDRAM Refresh Control Shadow Register (SDRRCSR) is shown in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 7-9. SDRAM Refresh Control Shadow Register (SDRRCSR)**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | REFRESH_RATE_SHDW | |
| R-0 | | R/W-1388h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-40. SDRAM Refresh Control Shadow Register (SDRRCSR) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | | Reserved |
| 15-0 | REFRESH_RATE_SHDW | 1388h | Shadow field for refresh rate in SDRRCR. This field is loaded into the REFRESH_RATE_SHDW field in SDRRCR when Slave Idle Acknowledge(SIdleAck) is asserted. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.6 SDRAM Timing 1 Register (SDRTIM1)

The SDRAM Timing 1 Register (SDRTIM1) is described in the figure and table below.

**Figure 7-10. SDRAM Timing 1 Register (SDRTIM1)**

| 31 29 | 28 25 | 24 21 | 20 17 |
|---|---|---|---|
| Reserved | T_RP | T_RCD | T_WR |
| R-0 | R/W-4h | R/W-4h | R/W-4h |

| 16 12 | 11 6 | 5 3 | 2 0 |
|---|---|---|---|
| T_RAS | T_RC | T_RRD | T_WTR |
| R/W-11h | R/W-16h | R/W-3h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-41. SDRAM Timing 1 Register (SDRTIM1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-29 | Reserved | | Reserved |
| 28-25 | T_RP | 4h | Specifies the mimimum number of DDR[X]_CLK cycles from a precharge command to a refresh or activate command, minus 1. Corresponds to tRP AC timing parameter in the DDR2/DDR3 data sheet.<br>Calculated by T_RP=(tRP/DDR[X]_CLK period)-1 |
| 24-21 | T_RCD | 4h | Specifies the mimimum number of DDR[X]_CLK cycles from an activate command to read or write command, minus 1. Corresponds to tRCD AC timing parameter in the DDR2/DDR3 data sheet.<br>Calculated by T_RP=(tRCD/DDR[X]_CLK period)-1 |
| 20-17 | T_WR | 4h | Specifies the minimum number of DDR[X]_CLK cycles from the last write transfer to a pre-charge command, minus 1. Corresponds to tWR AC timing parameter in the DDR2/DDR3 data sheet.<br>Calculated by T_RP=(tWR/DDR[X]_CLK period)-1 |
| 16-12 | T_RAS | 11h | Specifies the mimimum number of DDR[X]_CLK cycles from an active command to a pre-charge command, minus 1. Corresponds to tRAS AC timing parameter in the DDR2/DDR3 data sheet.<br>Calculated by T_RP=(tRAS/DDR[X]_CLK period)-1 |
| 11-6 | T_RC | 16h | Specifies the mimimum number of DDR[X]_CLK cycles from an active command to an active command, minus 1. Corresponds to tRC AC timing parameter in the DDR2/DDR3 data sheet.<br>Calculated by T_RP=(tRC/DDR[X]_CLK period)-1 |
| 5-3 | T_RRD | 3h | Specifies the mimimum number of DDR[X]_CLK cycles from an activate command to an activate command in a different bank, minus 1. Corresponds to tRRD AC timing parameter in the DDR2/DDR3 data sheet.<br>Calculated by T_RRD=(tRRD/DDR[X]_CLK period)-1<br>For an 8-bank DDR2 or DDR3 this field must be equal to ((tFAW/(4*DDR[X]_CLK))-1) |
| 2-0 | T_WTR | 1h | Specifies the minimum number of DDR[X]_CLK cycles from the last write to a read command, minus 1. Corresponds to the tWTR AC timing parameter in the DDR2/DDR3 data sheet.<br>Calculated by T_WTR=(tWTR/DDR[X]_CLK period)-1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.7 SDRAM Timing 1 Shadow Register (SDRTIM1SR)

The SDRAM Timing 1 Shadow Register (SDRTIM1SR) is described in the figure and table below.

**Figure 7-11. SDRAM Timing 1 Shadow Register (SDRTIM1SR)**

| 31 29 | 28 25 | 24 21 | 20 17 |
|---|---|---|---|
| Reserved | T_RP_SHDW | T_RCD_SHDW | T_WR_SHDW |
| R-0 | R/W-4h | R/W-4h | R/W-4h |

| 16 12 | 11 6 | 5 3 | 2 0 |
|---|---|---|---|
| T_RAS_SHDW | T_RC_SHDW | T_RRD_SHDW | T_WTR_SHDW |
| R/W-11h | R/W-16h | R/W-3h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-42. SDRAM Timing 1 Shadow Register (SDRTIM1SR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-29 | Reserved | | Reserved |
| 28-25 | T_RP_SHDW | 4h | Shadow field for T_RP in SDRTIMR1. This field is loaded into T_RP field in SDRAM Timing 1 Register when SIdleAck is asserted. |
| 24-21 | T_RCD_SHDW | 4h | Shadow field for T_RCD in SDRTIMR1. This field is loaded into T_RCD field in SDRAM Timing 1 Register when SIdleAck is asserted. |
| 20-17 | T_WR_SHDW | 4h | Shadow field for T_WR in SDRTIMR1. This field is loaded into T_WR field in SDRAM Timing 1 Register when SIdleAck is asserted. |
| 16-12 | T_RAS_SHDW | 11h | Shadow field for T_RAS in SDRTIMR1. This field is loaded into T_RAS field in SDRAM Timing 1 Register when SIdleAck is asserted. |
| 11-6 | T_RC_SHDW | 16h | Shadow field for T_RC in SDRTIMR1. This field is loaded into T_RC field in SDRAM Timing 1 Register when SIdleAck is asserted. |
| 5-3 | T_RRD_SHDW | 3h | Shadow field for T_RRD in SDRTIMR1. This field is loaded into T_RRD field in SDRAM Timing 1 Register when SIdleAck is asserted. |
| 2-0 | T_WTR_SHDW | 1h | Shadow field for T_WTR in SDRTIMR1. This field is loaded into T_WTR field in SDRAM Timing 1 Register when SIdleAck is asserted. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.8 SDRAM Timing 2 Register (SDRTIM2)

The SDRAM Timing 2 Register (SDRTIM2) is shown in the figure and table below.

**Figure 7-12. SDRAM Timing 2 Register (SDRTIM2)**

| 31 | 30 28 | 27 25 | 24 16 |
|---|---|---|---|
| Rsvd | T_XP | Reserved | T_XSNR |
| R-0 | R/W-1h | R-0 | R/W-8Bh |

| 15 6 | 5 3 | 2 0 |
|---|---|---|
| T_XSRD | T_RTP | T_CKE |
| R/W-C7h | R/W-1h | R/W-2h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-43. SDRAM Timing 2 Register (SDRTIM2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | | Reserved |
| 30-28 | T_XP | 1h | Specifies the minimum number of DDR[X]_CLK cycles from power down exit to any other command except read command, minus 1. For DDR2, this field must be greater than tXP or tCKE. Corresponds to tXP or tCKE AC timing parameter in the DDR2/DDR3 data sheet. |
| 27-25 | Reserved | | Reserved |
| 24-16 | T_XSNR | 0-8Bh | Specifies the minimum number of DDR[X]_CLK cycles from a self-refresh exit to any other command except a read command, minus 1. Corresponds to tXSNR/tXS AC timing parameter in the DDR2/DDR3 data sheet.<br>Calculated by T_XSNR=(tXSNR/DDR[X]_CLK period)-1 |
| 15-6 | T_XSRD | 0-C7h | Specifies the minimum number of DDR[X]_CLK cycles from a self-refresh exit to a read command, minus 1. Corresponds to tXSRD/tXSDLL AC timing parameter in the DDR2/DDR3 data sheet.<br>Calculated by T_XSRD=(tXSRD or tXSDLL)-1 |
| 5-3 | T_RTP | 1h | Specifies the minimum number of DDR[X]_CLK cycles from a last read command to a precharge command, minus 1. Corresponds to tRTP AC timing parameter in the DDR2/DDR3 data sheet.<br>Calculated by T_RTP=(tRTP/DDR[X]_CLK period)-1 |
| 2-0 | T_CKE | 2h | Specifies the minimum number of DDR[X]_CLK cycles between transitions on the DDR[X]_CKE[X] pin, minus 1. Corresponds to the tCKE AC timing parameter in the DDR2 data sheet.<br>Calculated by T_CKE=(tCKE/DDR[X]_CLK period)-1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.9 SDRAM Timing 2 Shadow Register (SDRTIM2SR)

The SDRAM Timing 2 Shadow Register (SDRTIM2SR) is shown in the figure and table below.

**Figure 7-13. SDRAM Timing 2 Shadow Register (SDRTIM2SR)**

| 31 | 30 28 | 27 25 | 24 16 |
|---|---|---|---|
| Rsvd | T_XP_SHDW | Reserved | T_XSNR_SHDW |
| R-0 | R/W-1h | 0h | R/W-8Bh |

| 15 6 | 5 3 | 2 0 |
|---|---|---|
| T_XSRD_SHDW | T_RTP_SHDW | T_CKE_SHDW |
| R/W-C7h | R/W-1h | R/W-2h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-44. SDRAM Timing 2 Shadow Register (SDRTIM2SR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved |
| 30-28 | T_XP_SHDW | 1h | Shadow field for T_XP in SDRTIMR2. This field is loaded into the T_XP field in the SDRAM Timing 2 Register when SIdleAck is asserted. |
| 27-25 | Reserved | 0h | Reserved |
| 24-16 | T_XSNR_SHDW | 8Bh | Shadow field for T_XSNR in SDRTIMR2. This field is loaded into the T_XSNR field in the SDRAM Timing 2 Register when SIdleAck is asserted. |
| 15-6 | T_XSRD_SHDW | C7h | Shadow field for T_XSRD in SDRTIMR2. This field is loaded into the T_XSRD field in the SDRAM Timing 2 Register when SIdleAck is asserted. |
| 5-3 | T_RTP_SHDW | 1h | Shadow field for T_RTP in SDRTIMR2. This field is loaded into the T_RTP field in the SDRAM Timing 2 Register when SIdleAck is asserted. |
| 2-0 | T_CKE_SHDW | 2h | Shadow field for T_CKE in SDRTIMR2. This field is loaded into theT_CKE field in the SDRAM Timing 2 Register when SIdleAck is asserted. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.10 SDRAM Timing 3 Register (SDRTIM3)

The SDRAM Timing 3 Register (SDRTIM3) is shown in the figure and table below.

**Figure 7-14. SDRAM Timing 3 Register (SDRTIM3)**

| 31 28 | 27 24 | 23 21 | 20 15 |
|---|---|---|---|
| T_PDLL_UL | T_CSTA | T_CKESR | T_ZQCS |
| R-5h | R/W-0h | R-7h | R/W-3Fh |

| 14 13 | 12 4 | 3 0 |
|---|---|---|
| Reserved | T_RFC | T_RAS_MAX |
| R-3h | R/W-82h | R/W-Fh |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-45. SDRAM Timing 3 Register (SDRTIM3) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | T_PDLL_UL | 5h | Specifies number of DDR[X]_CLK cycles for PHY DLL to unlock. A value of N will be equal to N x 128 clocks. This field must be written with a value of 5. |
| 27-24 | T_CSTA | 0h | Specifies the mimimum number of DDR[X]_CLK cycles between write-to-write or read-to-read data phases to different chip selects, minus one. This is a don't care when only only one chip select is being used. Refer the device data manual for the number of supported chip selects. |
| 23-21 | T_CKESR | 7h | Specifies the minimum number of DDR[x]_CLK cycles for which the memory must remain in self refresh, minus one. |
| 20-15 | T_ZQCS | 0-3Fh | Specifies the mimimum number of DDR[X]_CLK cycles for a ZQCS command, minus one. This is applicable only for DDR3. Corresponds to TZQCS=(tZQCS-1) Where tZQCS in DDR clock cycles. |
| 14-13 | Reserved | 3h | Reserved |
| 12-4 | T_RFC | 0-82h | Specifies the mimimum number of DDR[X]_CLK cycles from Refresh or Load Mode to Refresh or Activate, minus one.<br>Calculated by T_RFC=(tRFC/DDR[x]_CLK period)-1 |
| 3-0 | T_RAS_MAX | 0-Fh | Specifies the maximum number of refresh rate intervals from Active to Precharge command. This field must be programmed to Fh for DDR2/DDR3 SDRAM types. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.11 SDRAM Timing 3 Shadow Register (SDRTIM3SR)

The SDRTIM3SR is shown in the figure and table below.

**Figure 7-15. SDRAM Timing 3 Shadow Register (SDRTIM3SR)**

| 31–28 | 27–24 | 23–21 | 20–15 |
|---|---|---|---|
| T_PDLL_UL_SHDW | T_CSTA_SHDW | Reserved | T_ZQCS_SHDW |
| R/W-0h | R/W-0h | R/W-7h | R/W-3Fh |

| 14–13 | 12–4 | 3–0 |
|---|---|---|
| Reserved | T_RFC _SHDW | T_RAS_MAX _SHDW |
| R-3h | R/W-82h | R/W-Fh |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-46. SDRAM Timing 3 Shadow Register (SDRTIM3SR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | T_PDLL_UL_SHDW | 0h | Shadow field for T_PDLL_UL in SDRTIMR3. This field is loaded into T_PDLL_UL field in SDRAM Timing 3 Register when SIdleAck is asserted. This field must be written with a value of 5 |
| 27-24 | T_CSTA_SHDW | 0-h | Shadow field for T_CSTA in SDRTIMR3. This field is loaded into T_CSTA field in SDRAM Timing 3 Register when SIdleAck is asserted. |
| 23-21 | Reserved | 7h | Reserved |
| 20-15 | T_ZQCS _SHDW | 0-3Fh | Shadow field for T_ZQCS in SDRTIMR3. This field is loaded into T_ZQCS field in SDRAM Timing 3 Register when SIdleAck is asserted. |
| 14-13 | Reserved | 3h | Reserved |
| 12-4 | T_RFC _SHDW | 0-1FFh | Shadow field for T_RFC in SDRTIMR3. This field is loaded into T_RFC field in SDRAM Timing 3 Register when SIdleAck is asserted. |
| 3-0 | T_RAS_MAX _SHDW | 0-Fh | Shadow field for T_RAS_MAX in SDRTIMR3. This field is loaded into T_RASMAX field in SDRAM Timing 3 Register when SIdleAck is asserted. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.12 Power Management Control Register (PMCR)

The Power Management Control Register (PMCR) is shown in the figure and table below.

**Figure 7-16. Power Management Control Register (PMCR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 12 | 11 | 10 8 | 7 4 | 3 0 |
|---|---|---|---|---|
| PD_TIM | DPD_EN | LP_MODE | SR_TIM | CS_TIM |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-47. Power Management Control Register (PMCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | | Reserved |
| 15-12 | PD_TIM | | Power Management timer for Power Down. The DDR controller puts the external SDRAM in Power-Down mode after the DDR controller is idle for these number of DDR clock cycles and if LP_MODE field is set to 4h. |
| | | 0 | Immediately enter Power down Mode |
| | | 1h | After 16 DDR cycles |
| | | 2h | After 32 DDR cycles |
| | | 3h | After 64 DDR cycles |
| | | 4h | After 128 DDR cycles |
| | | 5h | After 256 DDR cycles |
| | | 6h | After 512 DDR cycles |
| | | 7h | After 1024 DDR cycles |
| | | 8h | After 2048 DDR cycles |
| | | 9h | After 4096 DDR cycles |
| | | Ah | After 8192 DDR cycles |
| | | Bh | After 16384 DDR cycles |
| | | Ch | After 32768 DDR cycles |
| | | Dh | After 65536 DDR cycles |
| | | Eh | After 131072 DDR cycles |
| | | Fh | After 262144 DDR cycles |
| 11 | DPD_DEN | | Deep Power Down Enable |
| | | 0 | Normal Operation |
| | | 1 | Overrides LP_MODE field setting |
| 10-8 | LP_MODE | | Automatic Power Management Enable |
| | | 1h | Clock Stop Mode |
| | | 2h | Self Refresh Mode |
| | | 4h | Power Down Mode |
| | | 0, 3h, 5h-7h | Disable Automatic power management |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 7-47. Power Management Control Register (PMCR) Field Descriptions (continued)**

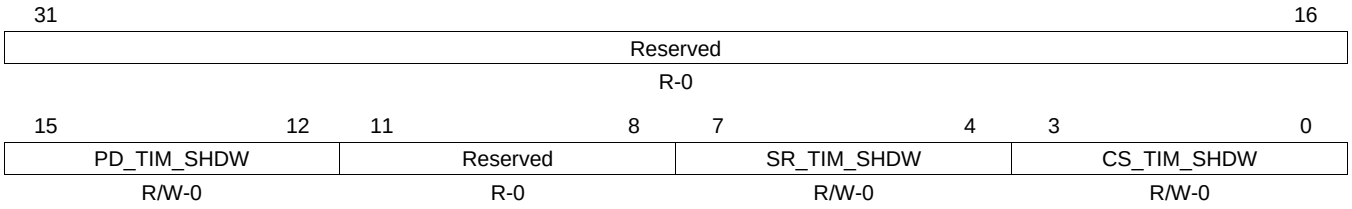
| Bit | Field | Value | Description |
|---|---|---|---|
| 7-4 | SR_TIM | | Power Management timer for Self Refresh. The DDR controller puts the external SDRAM in Self Refresh Mode after the DDR controller is idle for these number of DDR clock cycles and if LP_MODE is set to 2. When Self refresh mode is enabled, it is required to program this bit fileds with a value of 7 or higher to avoid sending more Refresh commands to the DDR memory than allowed maximum during a refresh period. |
| | | 0 | Immediately enter Self Refresh Mode |
| | | 1h | Enter Self Refresh Mode after 16 DDR Clocks |
| | | 2h | Enter Self Refresh Mode after 32 DDR Clocks |
| | | 3h | Enter Self Refresh Mode after 64 DDR Clocks |
| | | 4h | Enter Self Refresh Mode after 128 DDR Clocks |
| | | 5h | Enter Self Refresh Mode after 256 DDR Clocks |
| | | 6h | Enter Self Refresh Mode after 512 DDR Clocks |
| | | 7h | Enter Self Refresh Mode after 1024 DDR Clocks |
| | | 8h | Enter Self Refresh Mode after 2048 DDR Clocks |
| | | 9h | Enter Self Refresh Mode after 4096 DDR Clocks |
| | | Ah | Enter Self Refresh Mode after 8192 DDR Clocks |
| | | Bh | Enter Self Refresh Mode after 16384 DDR Clocks |
| | | Ch | Enter Self Refresh Mode after 32768 DDR Clocks |
| | | Dh | Enter Self Refresh Mode after 65536 DDR Clocks |
| | | Eh | Enter Self Refresh Mode after 131072 DDR Clocks |
| | | Fh | Enter Self Refresh Mode after 262144 DDR Clocks |
| 3-0 | CS_TIM | | Power Management timer for Clock Stop. The DDR controller puts the external SDRAM in clock stop mode after the DDR controller is idle for these number of DDR clock cycles and if LP_MODE field is set to 1. |
| | | 0 | Immediately enter Clock Stop Mode |
| | | 1h | Enter Clock Stop Mode after 16 clocks |
| | | 2h | Enter Clock Stop Mode after 32 clocks |
| | | 3h | Enter Clock Stop Mode after 64 clocks |
| | | 4h | Enter Clock Stop Mode after 128 clocks |
| | | 5h | Enter Clock Stop Mode after 256 clocks |
| | | 6h | Enter Clock Stop Mode after 512 clocks |
| | | 7h | Enter Clock Stop Mode after 1024 clocks |
| | | 8h | Enter Clock Stop Mode after 2048 clocks |
| | | 9h | Enter Clock Stop Mode after 4096 clocks |
| | | Ah | Enter Clock Stop Mode after 8192 clocks |
| | | Bh | Enter Clock Stop Mode after 16384 clocks |
| | | Ch | Enter Clock Stop Mode after 32768 clocks |
| | | Dh | Enter Clock Stop Mode after 65536 clocks |
| | | Eh | Enter Clock Stop Mode after 131072 clocks |
| | | Fh | Enter Clock Stop Mode after 262144 clocks |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.13 Power Management Control Shadow Register (PMCSR)

The Power Management Control Shadow Register (PMCSR) is shown in the figure and table below.

**Figure 7-17. Power Management Control Shadow Register (PMCSR)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-48. Power Management Control Shadow Register (PMCSR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | | Reserved |
| 15-12 | PD_TIM_SHDW | 0h | Shadow field for PD_TIM in PMCR. This field is loaded into PD_TIM field in Power Management Control Register when SidleAck is asserted. |
| 11-8 | Reserved | | Reserved |
| 7-4 | SR_TIM_SHDW | 0h | Shadow field for SR_TIM in PMCR. This field is loaded into SR_TIM field in Power Management Control Register when SidleAck is asserted. |
| 3-0 | CS_TIM_SHDW | 0h | Shadow field for CS_TIM in PMCR. This field is loaded into CS_TIM field in Power Management Control Register when SidleAck is asserted. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.14 Peripheral Bus Burst Priority Register (PBBPR)

The Peripheral Bus Burst Priority Register (PBBPR) is shown in the figure and table below.

**Figure 7-18. Peripheral Bus Burst Priority Register (PBBPR)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| Reserved | | COS_COUNT_1 | |
| R-0 | | R/W-FFh | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| COS_COUNT_2 | | PR_OLD_COUNT | |
| R/W-FFh | | R/W-FFh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-49. Peripheral Bus Burst Priority Register (PBBPR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Reserved | | Reserved |
| 23-16 | COS_COUNT_1 | FFh | Priority Raise Counter for class of service 1. Number of module clock cycles after which the DDR2/DDR3 memory controller momentarily raises the priority of the class of service 1 commands in the Command FIFO. Memory transfer clock cycles programmed will be N x 16 clocks. |
| 15-8 | COS_COUNT_2 | FFh | Priority Raise Counter for class of service 2. Number of module clock cycles after which the DDR2/DDR3 memory controller momentarily raises the priority of the class of service 1 commands in the Command FIFO. Memory transfer clock cycles programmed will be N x 16 clocks. |
| 7-0 | PR_OLD_COUNT | FFh | Priority Raise Old Counter. Number of clock cycles after which the DDR2/DDR3 memory controller momentarily raises the priority of the oldest command in the Command FIFO. Memory transfer clock cycles programmed will be N x 16 clocks. |

### 7.8.1.15 Performance Counter 1 Register (PERF_CNT_1)

The Performance Counter 1 register (PERF_CNT_1) is shown in the figure and table below.

**Figure 7-19. Performance Counter 1 Register (PERF_CNT_1)**

| 31 | 0 |
|---|---|
| COUNTER1 | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-50. Performance Counter 1 Register (PERF_CNT_1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | COUNTER1 | | 32-bit counter that can be configured as specified in the Performance Counter Config Register and Performance Counter Master Region Select Register. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.16 Performance Counter 2 Register (PERF_CNT_2)

The Performance Counter 2 register (PERF_CNT_2) is shown in the figure and table below.

**Figure 7-20. Performance Counter 2 Register (PERF_CNT_2)**

| 31 | 0 |
|---|---|
| COUNTER2 | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-51. Performance Counter 2 Register (PERF_CNT_2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | COUNTER2 | | 32-bit counter that can be configured as specified in the Performance Counter Config Register and Performance Counter Master Region Select Register. |

### 7.8.1.17 Performance Counter Config Register (PERF_CNT_CFG)

The Performance Counter Config register (PERF_CNT_CFG) is shown in the figure and table below.

**Figure 7-21. Performance Counter Config Register (PERF_CNT_CFG)**

| 31 | 30 | 29 20 | 19 16 |
|---|---|---|---|
| CNTR2_MCONNID_EN | Reserved | Reserved | CMTR2_CFG |
| R/W-0 | R/W-0 | R-0 | R/W-1 |

| 15 | 14 | 13 4 | 3 0 |
|---|---|---|---|
| CNTR1_MCCONNID_EN | Reserved | Reserved | CNTR1_CFG |
| R/W-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-52. Performance Counter Config Register (PERF_CNT_CFG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | CNTR2_MCONNID_EN | | MConnID filter enable for Performance Counter 2 register |
| 30 | Reserved | | Reserved |
| 29-20 | Reserved | | Reserved |
| 19-16 | CNTR2_CFG | | Filter configuration for Performance Counter 2. Refer to Section 7.8.1.16 for details |
| 15 | CNTR1_MCONNID_EN | | MConnID filter enable for Performance Counter 1 register |
| 14 | Reserved | | Reserved |
| 13-4 | Reserved | | Reserved |
| 3-0 | CNTR1_CFG | | Filter configuration for Performance Counter 1. Refer to Table 7-14 for details. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.18 Performance Counter Master Region Select Register (PERF_CNT_SEL)

The Performance Counter Master Region Select register (PERF_CNT_SEL) is shown in the figure and table below.

**Figure 7-22. Performance Counter Master Region Select Register (PERF_CNT_SEL)**

| 31 | 24 | 23 | 18 | 17 | 16 |
|---|---|---|---|---|---|
| MCONNID2 | | Reserved | | Reserved | |
| R/W-0 | | R-0 | | R/W-0 | |

| 15 | 8 | 7 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| MCONNID1 | | Reserved | | Reserved | |
| R/W-0 | | R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-53. Performance Counter Master Region Select Register (PERF_CNT_SEL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | MCONNID2 | | MConnID for Performance Counter2 register. |
| 23-18 | Reserved | | Reserved |
| 17-16 | Reserved | | Reserved |
| 15-8 | MCONNID1 | | MConnID for Performance Counter1 register |
| 7-2 | Reserved | | Reserved |
| 1-0 | Reserved | | Reserved |

### 7.8.1.19 End of Interrupt Register (EOI)

The End of Interrupt Register (EOI) is shown in the figure and table below.

**Figure 7-23. End of Interrupt Register (EOI)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | EOI |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-54. End of Interrupt Register (EOI) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | | Reserved |
| 0 | EOI | | Software End of Interrupt Control. |
| | | 0 | System OCP Interrupt |
| | | 1 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.20 System OCP Interrup RAW Status Register (SOIRSR)

The System OCP Interrup RAW Status Register (SOIRSR) is shown in the figure and table below.

**Figure 7-24. System OCP Interrup RAW Status Register (SOIRSR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | ERR_SYS |
| R-0 | | WOS-0 |

LEGEND: R = Read only; WOS = Write 1 to set, write of 0 has no effect; -*n* = value after reset

**Table 7-55. System OCP Interrup RAW Status Register (SOIRSR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | | Reserved |
| 0 | ERR_SYS | | Raw status of system OCP interrupt for command or address error. |
| | | 0 | Writing a 0 has no effect. |
| | | 1 | Write 1 to set the raw status. |

### 7.8.1.21 System OCP Interrupt Status Register (SOISR)

The System OCP Interrupt Status Register (SOISR) is shown in the figure and table below.

**Figure 7-25. System OCP Interrupt Status Register (SOISR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | ERR_SYS |
| R-0 | | WOC-0 |

LEGEND: R = Read only; WOC = Write 1 to clear, write of 0 has no effect; -*n* = value after reset

**Table 7-56. System OCP Interrupt Status Register (SOISR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | | Reserved |
| 0 | ERR_SYS | | Enabled status of system OCP interrupt for SDRAM command or address error. |
| | | 0 | Writing a 0 has no effect. |
| | | 1 | Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e., even if not enabled). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.22 System OCP Interrupt Enable Set Register (SOIESR)

The System OCP Interrupt Enable Set Register (SOIESR) is shown in the figure and table below.

**Figure 7-26. System OCP Interrupt Enable Set Register (SOIESR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | ERRSYSSET |
| R-0 | | WOS-0 |

LEGEND: R = Read only; WOS = Write 1 to set, a write of 0 has no effect; -*n* = value after reset

**Table 7-57. System OCP Interrupt Enable Set Register (SOIESR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | | Reserved |
| 0 | ERRSYSSET | | Enable set for sytem OCP interrupt for SDRAM command or address error. Write a 1 to set ERRSYSSET and the ERRSYSCLR bit in the System OCP Interrupt Enable Clear Register. |
| | | 0 | Interrupt is not Enabled |
| | | 1 | Interrupt is enabled. |

### 7.8.1.23 System OCP Interrupt Enable Clear Register (SOIECR)

The System OCP Interrupt Enable Clear Register (SOIECR) is shown in the figure and table below.

**Figure 7-27. System OCP Interrupt Enable Clear Register (SOIECR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | ERRSYSCLR |
| R-0 | | WOS-0 |

LEGEND: R/W = Read/Write; R = Read only; WOC = Write 1 to clear, a write of 0 has no effect; -*n* = value after reset

**Table 7-58. System OCP Interrupt Enable Clear Register (SOIECR) Field Descriptions**

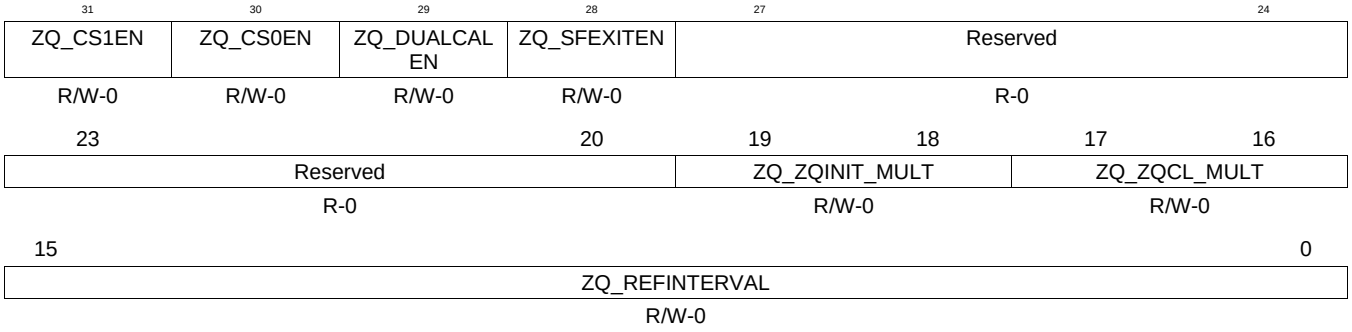
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | ERRSYSCLR | | Enable Clear for system OCP interrupt for SDRAM command or address error. Write a 1 to Clear ERRSYSCLR and the ERRSYSSET bit in the System OCP Interrupt Enable Set Register |
| | | 0 | No effect |
| | | 1 | Interrupt is enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.24 SDRAM Output Impedance Calibration Configuration Register (ZQCR)

The SDRAM Output Impedance Calibration Configuration Register (ZQCR) is shown in the figure and table below.

**Figure 7-28. SDRAM Output Impedance Calibration Configuration Register (ZQCR)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-59. SDRAM Output Impedance Calibration Configuration Register (ZQCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | ZQ_CS1EN | | Writing a 1 enables ZQ calibration for CS1 |
| 30 | ZQ_CS0EN | | Writing a 1 enables ZQ calibration for CS0 |
| 29 | ZQ_DUALCALEN | | ZQ Dual Calibration Enable.<br>Allows both ranks (chip selects) to be ZQ calibrated simultaneously.<br>Setting this bit requires both chip selects to have a separate calibration resistor per device. |
| 28 | ZQ_SFEXITEN | 0<br>1 | ZQCL on Self-Refresh, Active Power-Down and Precharge Power-Down exit enable.<br>Writing a 1 enables the issuing of ZQCL on Self-Refresh, Active Power-Down and Precharge Power Down exit. |
| 27-20 | Reserved | | Reserved |
| 19-18 | ZQ_ZQINIT_MULT | | Indicates number of ZQCL intervals that make up a ZQINIT interval, minus 1 |
| 17-16 | ZQ_ZQCL_MULT | | Indicates number of ZQCS intervals that make up a ZQCL interval, minus 1 |
| 15-0 | ZQ_REFINTERVAL | | Number of refresh periods between ZQCS commands. Supports between one refresh period to 256 ms between ZQCS calibration commands.<br>Refresh period is defined by REFRESH_RATE field in SDRAM Refresh Control Register. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.25 DDR PHY Control Register (DDRPHYCR)

The DDR PHY Control Register (DDRPHYCR) is shown in the figure and table below.

**Figure 7-29. DDR PHY Control Register (DDRPHYCR)**

| 31 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|
| Reserved | | DYN_PWRDN_EN | Rsvd | RDEYE_LVL_DIS | GATE_LVL_DIS | WR_LVL_DIS |
| R-0 | | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 5 | 4 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PHY_RST | Rsvd | IDLE_LOCAL_ODT | | Reserved | | RD_LOCAL_ODT | | Reserved | | READ_LATENCY | |
| R/W-0 | R/W-0 | R/W-0 | | R-0 | | R/W-0 | | R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-60. DDR PHY Control Register (DDRPHYCR) Field Descriptions**

| Bit | Field | Value | Description | |
|---|---|---|---|---|
| 31-21 | Reserved | 0 | Reserved | |
| 20 | DYN_PWRDN_EN | 0 | Enables dynamically powering down the IO Receiver when not performing a read. This will help in reducing DDR IO power. It is recommended to set this bit to '1' always. | |
| 19 | Reserved | 0 | Reserved | |
| 18 | RDEYE_LVL_DIS | 0 | When set to 1, read eye auto-leveling is disabled. This is applicable only for DDR3 | For proper work, it is reccomended that each of these three bits must be set to 0b1. |
| 17 | GATE_LVL_DIS | 0 | When set to 1, read gate auto-leveling is disabled. This is applicable only for DDR3 | |
| 16 | WR_LVL_DIS | 0 | When set to 1, write auto-leveling is disabled. This is applicable only for DDR3 | |
| 15 | PHY_RST | 0 | When set to 1, DDR PHY is reset .Both data and command macros will be reset | |
| 14 | Reserved | 0 | Reserved | |
| 13-12 | IDLE_LOCAL_ODT | 0 | This bit fields specify the ODT selection on the DDR Controller side when the breceiver is powered down. IDLE_LOCAL_ODT is applicable only when DYN_PWRDN_EN is set to '1'. Please refer to Table1-16 for the supported ODT termination values.<br>Recommended value is 0x0 [ODT OFF] | |
| 11-10 | Reserved | 0 | Reserved | |
| 9-8 | RD_LOCAL_ODT | 0 | This bit fields specifies the ODT selection on the DDR PHY side during READ. To control the ODT on the DDR device side, refer to the SDRCR register details. Please refer to Table1-16 for the supported ODT termination values.<br>Recommended value is 0x2 [50 Ohms termination] | |
| 7-5 | Reserved | 0 | Reserved | |
| 4-0 | READ_LATENCY | 0 | Defines the latency for read data from DDR SDRAM in number of 1x cycles. The value applied should be equal to the required value minus one. The maximum read latency supported by the DDR PHY is equal to CAS latency plus 7 clock cycles. The minimum read latency must be equal to CAS latency plus 1 clock cycles. | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.26 DDR PHY Control Shadow Register (DDRPHYCSR)

TheDDR PHY Control Shadow Register (DDRPHYCSR) is shown in the figure and table below.

**Figure 7-30. DDR PHY Control Shadow Register (DDRPHYCSR)**

| 31 – 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|
| Reserved | DYN_PWRDN_EN | Rsvd | RDEYE_LVL_DIS | GATE_LVL_DIS | WR_LVL_DIS |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 – 12 | 11 – 10 | 9 – 8 | 7 – 5 | 4 – 0 |
|---|---|---|---|---|---|---|
| PHY_RST | Reserved | IDLE_LOCAL_ODT | Reserved | RD_LOCAL_ODT | Reserved | READ_LATENCY |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-61. DDR PHY Control Shadow Register (DDRPHYCSR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-21 | Reserved | | Reserved |
| 20 | DYN_PWRDN_EN | | Shadow field for DYN_PWRDN_EN in DDRPHYCR. This field is loaded into DYN_PWRDN_EN field in DDR PHY Control Register when SIdleAck is asserted |
| 19 | Reserved | | Reserved |
| 18 | RDEYE_LVL_DIS | | Shadow field for RDEYE_LVL_DIS in DDRPHYCR. This field is loaded into RDEYE_LVL_DIS field in DDR PHY Control Register when SIdleAck is asserted. |
| 17 | GATE_LVL_DIS | | Shadow field for GATE_LVL_DIS in DDRPHYCR. This field is loaded into GATE_LVL_DIS field in DDR PHY Control Register when SIdleAck is asserted. |
| 16 | WR_LVL_DIS | | Shadow field for WR_LVL_DIS in DDRPHYCR. This field is loaded into WR_LVL_DIS field in DDR PHY Control Register when SIdleAck is asserted |
| 15 | PHY_RST | | Shadow field for PHY_RST in DDRPHYCR. This field is loaded into PHY_RST field in DDR PHY Control Register when SIdleAck is asserted. |
| 14 | Reserved | | Reserved |
| 13-12 | IDLE_LOCAL_ODT | | This bit fields specify the ODT selection on the DDR Controller side when the breceiver is powered down. IDLE_LOCAL_ODT is applicable only when DYN_PWRDN_EN is set to '1'.<br>Please refer to Table1-16 for the supported ODT termination values. |
| 11-10 | Reserved | | Recommended value is 0x0 [ODT OFF] |
| 9-8 | RD_LOCAL_ODT | | This bit fields specifies the ODT selection on the DDR PHY side during READ. To control the ODT on the DDR device side, refer to the SDRCR register details.<br>Please refer to Table1-16 for the supported ODT termination values. |
| 7-5 | Reserved | | Recommended value is 0x2 [50 Ohms termination] |
| 4-0 | READ_LATENCY | | Shadow field for READ_LATENCY in DDRPHYCR. This field is loaded into READ_LATENCY field in DDR PHY Control Register when SIdleAck is asserted. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.27 Priority to Class of Service Mapping Register (PRI_COS_MAP)

The Priority to Class of Service Mapping Register(PRI_COS_MAP) is shown in the figure and table below.

**Figure 7-31. Priority to Class of Service Mapping Register (PRI_COS_MAP)**

| 31 | 30 | 16 |
|---|---|---|
| PRI_COS_MAP_EN | Reserved | |
| R/W-0 | R-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PRI_7_COS | | PRI_6_COS | | PRI_5_COS | | PRI_4_COS | | PRI_3_COS | | PRI_2_COS | | PRI_1_COS | | PRI_0_COS | |
| R/W-0 | | R/W-0 | | R/W-0 | | R/W-0 | | R/W-0 | | R/W-0 | | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-62. Priority to Class of Service Mapping Register (PRI_COS_MAP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | PRI_COS_MAP_EN | 0 | Disable mapping |
| | | 1 | Enable priority to class of service mapping |
| 30-16 | Reserved | | Reserved |
| 15-14 | PRI_7_COS | | Class of service for commands with priority of 7. |
| | | 0 or 3 | Setting a value of 0 or 3 will not assign any class of service. |
| | | 1 or 2 | Commands with priority 7 is mapped to class of service 1 or 2 |
| 13-12 | PRI_6_COS | | Class of service for commands with priority of 6. |
| | | 0 or 3 | Setting a value of 0 or 3 will not assign any class of service. |
| | | 1 or 2 | Commands with priority 6 is mapped to class of service 1 or 2 |
| 11-10 | PRI_5_COS | | Class of service for commands with priority of 5. |
| | | 0 or 3 | Setting a value of 0 or 3 will not assign any class of service. |
| | | 1 or 2 | Commands with priority 5 is mapped to class of service 1 or 2 |
| 9-8 | PRI_4_COS | | Class of service for commands with priority of 4. |
| | | 0 or 3 | Setting a value of 0 or 3 will not assign any class of service. |
| | | 1 or 2 | Commands with priority 4 is mapped to class of service 1 or 2 |
| 7-6 | PRI_3_COS | | Class of service for commands with priority of 3. |
| | | 0 or 3 | Setting a value of 0 or 3 will not assign any class of service. |
| | | 1 or 2 | Commands with priority 3 is mapped to class of service 1 or 2 |
| 5-4 | PRI_2_COS | | Class of service for commands with priority of 2. |
| | | 0 or 3 | Setting a value of 0 or 3 will not assign any class of service. |
| | | 1 or 2 | Commands with priority 2 is mapped to class of service 1 or 2 |
| 3-2 | PRI_1_COS | | Class of service for commands with priority of 1. |
| | | 0 or 3 | Setting a value of 0 or 3 will not assign any class of service. |
| | | 1 or 2 | Commands with priority 1 is mapped to class of service 1 or 2 |
| 1-0 | PRI_0_COS | | Class of service for commands with priority of 0. |
| | | 0 or 3 | Setting a value of 0 or 3 will not assign any class of service. |
| | | 1 or 2 | Commands with priority 0 is mapped to class of service 1 or 2 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.28 Connection ID to Class of Service 1 Mapping Register (CONNID_COS_1_MAP)

The Connection ID to Class of Service 1 Mapping Register(CONNID_COS_1_MAP) is shown in the figure and table below.

**Figure 7-32. Connection ID to Class of Service 1 Mapping Register(CONNID_COS_1_MAP)**

| 31 | 30 23 | 22 20 | 19 16 |
|---|---|---|---|
| CONNID_COS_1_MAP_EN | CONNID_1_COS_1 | MSK_1_COS_1 | CONNID_2_COS_1 |
| R/W-0 | R/W-0 | R/W-0 4 | R/W-0 |

| 15 12 | 11 10 | 9 2 | 1 0 |
|---|---|---|---|
| CONNID_2_COS_1 | MSK_2_COS_1 | CONNID_3_COS_1 | MSK_3_COS_1 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-63. Connection ID to Class of Service 1 Mapping Register(CONNID_COS_1_MAP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | CONNID_COS_1_MAP_EN | | Connection ID to class of service |
| | | 0 | Disable mapping |
| | | 1 | Enable connection ID to class of service |
| 30-23 | CONNID_1_COS_1 | | Connection ID value 1 for class of service 1. |
| 22-20 | MSK_1_COS_1 | | Mask for connection ID value 1 for class of service 1. |
| | | 0 | Disable masking |
| | | 1 | Mask connection ID bit 0 |
| | | 2 | Mask connection ID bits 1:0 |
| | | 3 | Mask connection ID bits 2:0 |
| | | 4 | Mask connection ID bits 3:0 |
| | | 5 | Mask connection ID bits 4:0 |
| | | 6 | Mask connection ID bits 5:0 |
| | | 7 | Mask connection ID bits 6:0 |
| 19-12 | CONNID_2_COS_1 | | Connection ID value 2 for class of service 1 |
| 11-10 | MSK_2_COS_1 | | Mask for connection ID value 2 for class of service 1 |
| | | 0 | Disable masking |
| | | 1 | Mask connection ID bit 0 |
| | | 2 | Mask connection ID bits 1:0 |
| | | 3 | Mask connection ID bits 2:0 |
| 9-2 | CONNID_3_COS_1 | | Connection ID value 3 for class of service 1. |
| 1-0 | MSK_3_COS_1 | | Mask for connection ID value 3 for class of service 1 |
| | | 0 | Disable masking |
| | | 1 | Mask connection ID bit 0 |
| | | 2 | Mask connection ID bits 1:0 |
| | | 3 | Mask connection ID bits 2:0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.29 Connection ID to Class of Service 2 Mapping Register (CONNID_COS_2_MAP)

The Connection ID to Class of Service 2 Mapping Register (CONNID_COS_2_MAP) is shown in the figure and table below.

**Figure 7-33. Connection ID to Class of Service 2 Mapping Register (CONNID_COS_2_MAP)**

| 31 | 30 23 | 22 20 | 19 16 |
|---|---|---|---|
| CONNID_COS_2_MAP_EN | CONNID_1_COS_2 | MSK_1_COS_2 | CONNID_2_COS_2 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 12 | 11 10 | 9 2 | 1 0 |
|---|---|---|---|
| CONNID_2_COS_2 | MSK_2_COS_2 | CONNID_3_COS_2 | MSK_3_COS_2 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-64. Connection ID to Class of Service 2 Mapping Register (CONNID_COS_2_MAP) Field Descriptions**

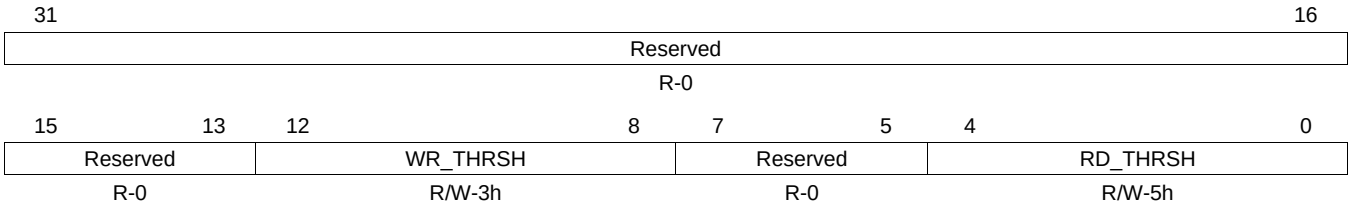
| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | CONNID_COS_2_MAP_EN | | Connection ID to class of service |
| | | 0 | Disable mapping |
| | | 1 | Enable connection ID to class of service |
| 30-23 | CONNID_1_COS_2 | | Connection ID value 1 for class of service 2. |
| 22-20 | MSK_1_COS_2 | | Mask for connection ID value 1 for class of service 2. |
| | | 0 | Disable masking |
| | | 1 | Mask connection ID bit 0 |
| | | 2 | Mask connection ID bits 1:0 |
| | | 3 | Mask connection ID bits 2:0 |
| | | 4 | Mask connection ID bits 3:0 |
| | | 5 | Mask connection ID bits 4:0 |
| | | 6 | Mask connection ID bits 5:0 |
| | | 7 | Mask connection ID bits 6:0 |
| 19-12 | CONNID_2_COS_2 | | Connection ID value 2 for class of service 2 |
| 11-10 | MSK_2_COS_2 | | Mask for connection ID value 2 for class of service 2 |
| | | 0 | Disable masking |
| | | 1 | Mask connection ID bit 0 |
| | | 2 | Mask connection ID bits 1:0 |
| | | 3 | Mask connection ID bits 2:0 |
| 9-2 | CONNID_3_COS_2 | | Connection ID value 3 for class of service 2. |
| 1-0 | MSK_3_COS_2 | | Mask for connection ID value 3 for class of service 2 |
| | | 0 | Disable masking |
| | | 1 | Mask connection ID bit 0 |
| | | 2 | Mask connection ID bits 1:0 |
| | | 3 | Mask connection ID bits 2:0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.1.30 Read Write Execution Threshold Register (RD_WR_EXEC_THRSH)

The Read Write Execution Threshold Register (RD_WR_EXEC_THRSH) is shown in the figure and table below.

**Figure 7-34. Read Write Execution Threshold Register (RD_WR_EXEC_THRSH)**




LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 7-65. Read Write Execution Threshold Register (RD_WR_EXEC_THRSH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-13 | Reserved | | Reserved |
| 12-8 | WR_THRSH | 3h | Write Threshold. Number of SDRAM write bursts after which the DDR2/DDR3 memory controller arbitration will switch to executing read commands. The value programmed is always minus one the required number. |
| 7-5 | Reserved | | Reserved |
| 4-0 | RD_THRSH | 5h | Read Threshold. Number of SDRAM read bursts after which the DDR2/DDR3 memory controller arbitration will switch to executing write commands. The value programmed is always minus one the required number. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.2 DDR2/DDR3 PHY Registers

Table 7-66 lists the memory-mapped registers for the DDR2/DDR3 PHY. Configure the DDR PHY Control Register for calibration of board delay and other parameters to get the SDRAM device working at a different speed. See the device-specific data manual for the base memory address of these registers.

**Table 7-66. Memory-Mapped Registers for DDR2 and DDR3 PHY**

| Register Name | Type | Description | Reset | Offset |
|---|---|---|---|---|
| CMD0_REG_PHY_CTRL_SLAVE_RATIO_0 | RW | DDR PHY Command 0 Address/Command Slave Ratio Register | 0x80 | 0x01C |
| CMD0_REG_PHY_DLL_LOCK_DIFF_0 | RW | DDR PHY Command 0 Address/Command DLL Lock Difference Register | 0x4 | 0x028 |
| CMD0_REG_PHY_INVERT_CLKOUT_0 | RW | DDR PHY Command 0 Invert Clockout Selection Register | 0x0 | 0x02C |
| CMD1_REG_PHY_CTRL_SLAVE_RATIO_0 | RW | DDR PHY Command 1 Address/Command Slave Ratio Register | 0x80 | 0x050 |
| CMD1_REG_PHY_DLL_LOCK_DIFF_0 | RW | DDR PHY Command 1 Address/Command DLL Lock Difference Register | 0x4 | 0x05C |
| CMD1_REG_PHY_INVERT_CLKOUT_0 | RW | DDR PHY Command 1 Invert Clockout Selection Register | 0x0 | 0x060 |
| CMD2_REG_PHY_CTRL_SLAVE_RATIO_0 | RW | DDR PHY Command 2 Address/Command Slave Ratio Register | 0x80 | 0x084 |
| CMD2_REG_PHY_DLL_LOCK_DIFF_0 | RW | DDR PHY Command 2 Address/Command DLL Lock Difference Register | 0x4 | 0x090 |
| CMD2_REG_PHY_INVERT_CLKOUT_0 | RW | DDR PHY Command 2 Invert Clockout Selection Register | 0x0 | 0x094 |
| DATA0_REG_PHY_RD_DQS_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 0 Read DQS Slave Ratio Register | 0x04010040 | 0x0C8 |
| DATA0_REG_PHY_WR_DQS_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 0 Write DQS Slave Ratio Register | 0x0 | 0x0DC |
| DATA0_REG_PHY_FIFO_WE_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 0 DQS Gate Slave Ratio Register | 0x0 | 0x108 |
| DATA0_REG_PHY_DQ_OFFSET_0 | RW | Offset value from DQS to DQ for Data Macro 0 | 0x40 | 0x11C |
| DATA0_REG_PHY_WR_DATA_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 0 Write Data Slave Ratio Register | 0x04010040 | 0x120 |
| DATA0_REG_PHY_USE_RANK0_DELAYS | RW | DDR PHY Data Macro 0 Delay Selection Register | 0x0 | 0x134 |
| DATA0_REG_PHY_DLL_LOCK_DIFF_0 | RW | DDR PHY Data Macro 0 DLL Lock Difference Register | 0x4 | 0x138 |
| DATA1_REG_PHY_RD_DQS_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 1 Read DQS Slave Ratio Register | 0x04010040 | 0x16C |
| DATA1_REG_PHY_WR_DQS_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 1 Write DQS Slave Ratio Register | 0x0 | 0x180 |
| DATA1_REG_PHY_FIFO_WE_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 1 DQS Gate Slave Ratio Register | 0x0 | 0x1AC |
| DATA1_REG_PHY_DQ_OFFSET_1 | RW | Offset value from DQS to DQ for Data Macro 1 | 0x40 | 0x1C0 |
| DATA1_REG_PHY_WR_DATA_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 1 Write Data Slave Ratio Register | 0x04010040 | 0x1C4 |
| DATA1_REG_PHY_USE_RANK0_DELAYS | RW | DDR PHY Data Macro 1 Delay Selection Register | 0x0 | 0x1D8 |
| DATA1_REG_PHY_DLL_LOCK_DIFF_0 | RW | DDR PHY Data Macro 1 DLL Lock Difference Register | 0x4 | 0x1DC |
| DATA2_REG_PHY_RD_DQS_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 2 Read DQS Slave Ratio Register | 0x04010040 | 0x210 |
| DATA2_REG_PHY_WR_DQS_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 2 Write DQS Slave Ratio Register | 0x0 | 0x224 |
| DATA2_REG_PHY_FIFO_WE_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 2 DQS Gate Slave Ratio Register | 0x0 | 0x250 |



Copyright © 2011–2016, Texas Instruments Incorporated

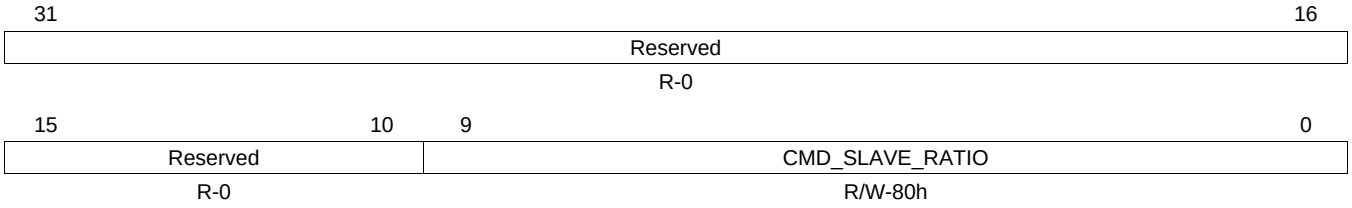
**Table 7-66. Memory-Mapped Registers for DDR2 and DDR3 PHY (continued)**

| Register Name | Type | Description | Reset | Offset |
|---|---|---|---|---|
| DATA2_REG_PHY_DQ_OFFSET_2 | RW | Offset value from DQS to DQ for Data Macro 2 | 0x40 | 0x264 |
| DATA2_REG_PHY_WR_DATA_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 2 Write Data Slave Ratio Register | 0x04010040 | 0x268 |
| DATA2_REG_PHY_USE_RANK0_DELAYS | RW | DDR PHY Data Macro 2 Delay Selection Register | 0x0 | 0x27C |
| DATA2_REG_PHY_DLL_LOCK_DIFF_0 | RW | DDR PHY Data Macro 2 DLL Lock Difference Register | 0x4 | 0x280 |
| DATA3_REG_PHY_RD_DQS_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 3 Read DQS Slave Ratio Register | 0x04010040 | 0x2B4 |
| DATA3_REG_PHY_WR_DQS_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 3 Write DQS Slave Ratio Register | 0x0 | 0x2C8 |
| DATA3_REG_PHY_FIFO_WE_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 3 DQS Gate Slave Ratio Register | 0x0 | 0x2F4 |
| DATA3_REG_PHY_DQ_OFFSET_3 | RW | Offset value from DQS to DQ for Data Macro 3 | 0x40 | 0x308 |
| DATA3_REG_PHY_WR_DATA_SLAVE_RATIO_0 | RW | DDR PHY Data Macro 3 Write Data Slave Ratio Register | 0x04010040 | 0x30C |
| DATA3_REG_PHY_USE_RANK0_DELAYS | RW | DDR PHY Data Macro 3 Delay Selection Register | 0x0 | 0x320 |
| DATA3_REG_PHY_DLL_LOCK_DIFF_0 | RW | DDR PHY Data Macro 3 DLL Lock Difference Register | 0x4 | 0x324 |

### 7.8.2.1 DDR PHY Command 0/1/2 Address/Command Slave Ratio Register (CMD0/1/2_REG_PHY_CTRL_SLAVE_RATIO_0)

The DDR PHY Command 0/1/2 Address/Command Slave Ratio Register (CMD0/1/2_REG_PHY_CTRL_SLAVE_RATIO_0) is shown in the figure and table below.

**Figure 7-35. DDR PHY Command 0/1/2 Address/Command Slave Ratio Register (CMD0/1/2_REG_PHY_CTRL_SLAVE_RATIO_0)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-67. DDR PHY Command 0/1/2 Address/Command Slave Ratio Register (CMD0/1/2_REG_PHY_CTRL_SLAVE_RATIO_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | | Reserved |
| 9-0 | CMD_SLAVE_RATIO | 0-80h | Ratio value for address/command launch timing in DDR PHY macro. This is the fraction of a clock cycle represented by the shift to be applied to the read DQS in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. |

### 7.8.2.2 DDR PHY Command 0/1/2 Address/Command DLL Lock Difference Register ( MD0/1/2_REG_PHY_DLL_LOCK_DIFF_0)

The DDR PHY Command 0/1/2 Address/Command DLL Lock Difference Register (CMD0/1/2_REG_PHY_DLL_LOCK_DIFF_0) is shown in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 7-36. DDR PHY Command 0/1/2 Address/Command DLL Lock Difference Register (CMD0/1/2_REG_PHY_DLL_LOCK_DIFF_0)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | DLL_LOCK_DIFF | |
| R-0 | | R/W-4h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-68. DDR PHY Command 0/1/2 Address/Command DLL Lock Difference Register ( CMD0/1/2_REG_PHY_DLL_LOCK_DIFF_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | | Reserved |
| 3-0 | DLL_LOCK_DIFF | 0-4h | The max number of delay line taps variation allowed while maintaining the master DLL lock.This is calculated as total jitter/ delay line tap size, where total jitter is half of (incoming clock jitter (pp) + delay line jitter (pp)). |

### 7.8.2.3 DDR PHY Command 0/1/2 Invert Clockout Selection Register (CMD0/1/2_REG_PHY_INVERT_CLKOUT_0)

The CDDR PHY Command 0/1/2 Invert Clockout Selection Register( CMD0/1/2_REG_PHY_INVERT_CLKOUT_0) is shown in the figure and table below.

**Figure 7-37. DDR PHY Command 0/1/2 Invert Clockout Selection Register (CMD0/1/2_REG_PHY_INVERT_CLKOUT_0)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | INVERT_CLK_SEL |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-69. DDR PHY Command 0/1/2 Invert Clockout Selection Register( CMD0/1/2_REG_PHY_INVERT_CLKOUT_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | INVERT_CLK_SEL | | Inverts the polarity of DRAM clock. |
| | | 0 | Core clock is passed on to DRAM |
| | | 1 | inverted core clock is passed on to DRAM |

### 7.8.2.4 DDR PHY Data Macro 0/1/2/3 Read DQS Slave Ratio Register (DATA0/1/2/3_REG_PHY_RD_DQS_SLAVE_RATIO_0)

The DDR PHY Data Macro 0/1/2/3 Read DQS Slave Ratio Register(DATA0/1/2/3_REG_PHY_RD_DQS_SLAVE_RATIO_0) is shown in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 7-38. DDR PHY Data Macro 0/1/2/3 Read DQS Slave Ratio Register (DATA0/1/2/3_REG_PHY_RD_DQS_SLAVE_RATIO_0))**

| 31 | 19 | 16 |
|---|---|---|
| Reserved | RD_DQS_SLAVE_RATIO_CS1 | |
| R/W-40h | R/W-40h | |

| 15 | 10 | 9 | 0 |
|---|---|---|---|
| RD_DQS_SLAVE_RATIO_CS1 | | RD_DQS_SLAVE_RATIO_CS0 | |
| R/W-40h | | R/W-40h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-70. DDR PHY Data Macro 0/1/2/3 Read DQS Slave Ratio Register (DATA0/1/2/3_REG_PHY_RD_DQS_SLAVE_RATIO_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | Reserved | | Reserved |
| 19-10 | RD_DQS_SLAVE_RATIO_CS1 | 40h | Ratio value for Read DQS slave DLL for CS1.<br>This is the fraction of a clock cycle represented by the shift to be applied to the read DQS in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. |
| 9-0 | RD_DQS_SLAVE_RATIO_CS0 | 40h | Ratio value for Read DQS slave DLL for CS0.<br>This is the fraction of a clock cycle represented by the shift to be applied to the read DQS in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.2.5 DDR PHY Data Macro 0/1/2/3 Write DQS Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DQS_SLAVE_RATIO_0)

The DDR PHY Data Macro 0/1/2/3 Write DQS Slave Ratio Register(DATA0/1/2/3_REG_PHY_WR_DQS_SLAVE_RATIO_0) is shown in the figure and table below.

**Table 7-71. DDR PHY Data Macro 0/1/2/3 Write DQS Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DQS_SLAVE_RATIO_0)**

| 31 | 20 | 19 | 16 |
|---|---|---|---|
| Reserved | | WR_DQS_SLAVE_RATIO_CS1 | |
| R-0 | | R/W-0h | |

| 15 | 10 | 9 | 0 |
|---|---|---|---|
| Reserved | | WR_DQS_SLAVE_RATIO_CS0 | |
| R-0 | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-72. DDR PHY Data Macro 0/1/2/3 Write DQS Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DQS_SLAVE_RATIO_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | Reserved | 0 | Reserved |
| 19-10 | WR_DQS_SLAVE_ RATIO_CS1 | | Ratio value for Write DQS slave DLL for CS1.<br>This is the fraction of a clock cycle represented by the shift to be applied to the write DQS in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. |
| 9-0 | WR_DQS_SLAVE_ RATIO_CS0 | | Ratio value for Write DQS slave DLL for CS0.<br>This is the fraction of a clock cycle represented by the shift to be applied to the write DQS in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. |

### 7.8.2.6 DDR PHY Data Macro 0/1/2/3 DQS Gate Slave Ratio Register (DATA0/1/2/3_REG_PHY_FIFO_WE_SLAVE_RATIO_0)

The DDR PHY Data Macro 0/1/2/3 DQS Gate Slave Ratio Register (DATA0/1/2/3_REG_PHY_FIFO_WE_SLAVE_RATIO_0) is shown in the figure and table below.

**Figure 7-39. DDR PHY Data Macro 0/1/2/3 DQS Gate Slave Ratio Register(DATA0/1/2/3_REG_PHY_FIFO_WE_SLAVE_RATIO_0)**

| 31 | 20 | 19 | 16 |
|---|---|---|---|
| Reserved | | FIFO_WE_SLAVE_RATIO_CS1 | |
| R-0 | | R/W-0 | |

| 15 | 10 | 9 | 0 |
|---|---|---|---|
| FIFO_WE_SLAVE_RATIO_CS1 | | FIFO_WE_SLAVE_RATIO_CS0 | |
| R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-73. DDR PHY Data Macro 0/1/2/3 DQS Gate Slave Ratio Register (DATA0/1/2/3_REG_PHY_FIFO_WE_SLAVE_RATIO_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | Reserved | 0 | Reserved |
| 19-10 | RD_DQS_GATE _SLAVE_RATIO_CS1 | 0h | Ratio value for fifo we for CS1. |
| 9-0 | RD_DQS_GATE _SLAVE_RATIO_CS0 | 0h | Ratio value for fifo we for CS0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.2.7 DDR PHY Data Macro 0/1/2/3 Write Data Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DATA_SLAVE_RATIO_0)

The DDR PHY Data Macro 0/1/2/3 Write Data Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DATA_SLAVE_RATIO_0) is shown in the figure and table below.

**Figure 7-40. DDR PHY Data Macro 0/1/2/3 Write Data Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DATA_SLAVE_RATIO_0)**

| 31 | 20 | 19 | 16 |
|---|---|---|---|
| Reserved | | WR_DATA_SLAVE_RATIO_CS1 | |
| R-40h | | R/W-40h | |

| 15 | 10 | 9 | 0 |
|---|---|---|---|
| WR_DATA_SLAVE_RATIO_CS1 | | WR_DATA_SLAVE_RATIO_CS0 | |
| R/W-40h | | R/W-40h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-74. DDR PHY Data Macro 0/1/2/3 Write Data Slave Ratio Register (DATA0/1/2/3_REG_PHY_WR_DATA_SLAVE_RATIO_0) Field Descriptions**

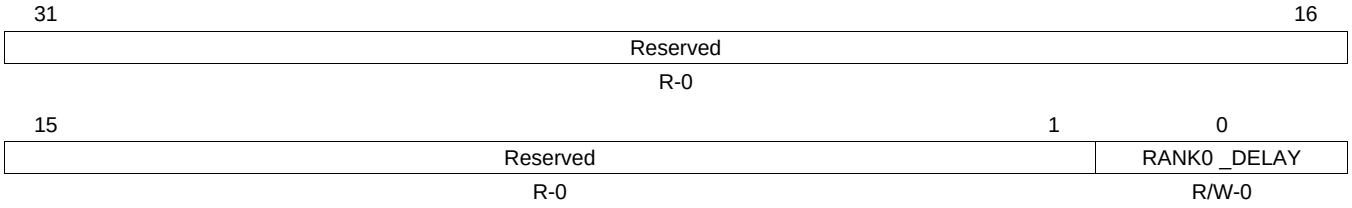
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | Reserved | 40h | Reserved |
| 19-10 | WR_DATA _SLAVE_RATIO_CS1 | 40h | Ratio value for write data slave DLL for CS1.Memory-Mapped Registers for DDR2/3<br>This is the fraction of a clock cycle represented by the shift to be applied to the write DQ muxes in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. |
| 9-0 | WR_DATA_SLAVE_RATIO_CS0 | 40h | Ratio value for write data slave DLL for CS0.<br>This is the fraction of a clock cycle represented by the shift to be applied to the write DQ muxes in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.2.8 DDR PHY Data Macro 0/1/2/3 Delay Selection Register (DATA0/1/2/3_REG_PHY_USE_RANK0_DELAYS)

The DATA0/1/2/3_REG_PHY_USE_RANK0_DELAYS is shown in Figure 7-41 and described in Table 7-75.

**Figure 7-41. DDR PHY Data Macro 0/1/2/3 Delay Selection Register (DATA0/1/2/3_REG_PHY_USE_RANK0_DELAYS)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

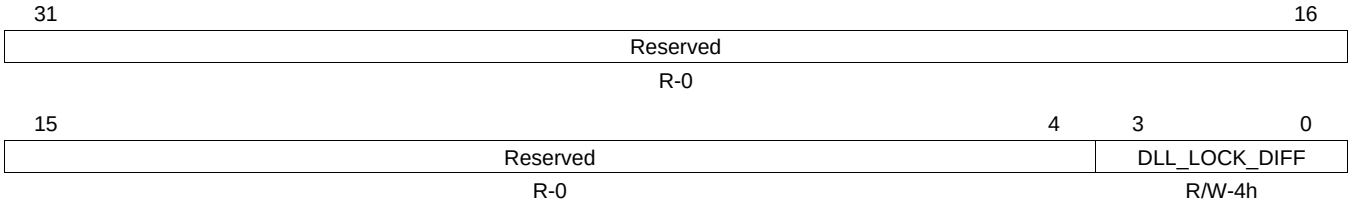
**Table 7-75. DDR PHY Data Macro 0/1/2/3 Delay Selection Register (DATA0/1/2/3_REG_PHY_USE_RANK0_DELAYS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | PHY_USE_RANK0_ DELAYS_0 | | Delay Selection |
| | | 0 | Each Rank uses its own delay. (Recommended). This is applicable only in case of DDR3 |
| | | 1 | Rank 0 delays are used for all ranks. This must be set to 1 for DDR2. |

### 7.8.2.9 DDR PHY Data Macro 0/1/2/3 DLL Lock Difference Register (DATA0/1/2/3_REG_PHY_DLL_LOCK_DIFF_0)

The DDR PHY Data Macro 0/1/2/3 DLL Lock Difference (DATA0/1/2/3_REG_PHY_DLL_LOCK_DIFF_0) register is shown in Figure 7-42 and described in Table 7-76.

**Figure 7-42. DDR PHY Data Macro 0/1/2/3 DLL Lock Difference Register (DATA0/1/2/3_REG_PHY_DLL_LOCK_DIFF_0)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-76. DDR PHY Data Macro 0/1/2/3 DLL Lock Difference Register (DATA0/1/2/3_REG_PHY_DLL_LOCK_DIFF_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | | Reserved |
| 3-0 | DLL_LOCK_DIFF | | Lock Difference |
| | | 0-4h | The max number of delay line taps variation allowed while maintaining the master DLL lock. This should always be written with the default value of 4h. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.3 DDR Related Control Module Registers

This section describes the EMIF PHY Clock Gate, VTP and DDR IO configuration registers which reside in the device control module register map. Refer below register description for more details. See the device-specific data manual for the base memory address of these registers.

**Table 7-77. DDR Related Control Module Registers**

| Register Name | Type | Register Description | Register Reset | Address Offset |
|---|---|---|---|---|
| EMIF_CLK_GATE | RW | EMIF0/1 PHY Clock gate Control Register | 0xC | 0x694 |
| DDR0_IO_CTRL | RW | DDR Memory Controller_0 IO Control Register | 0x02131313 | 0xE04 |
| DDR1_IO_CTRL | RW | DDR Memory Controller_1 IO Control Register | 0x02131313 | 0xE08 |
| DDR_VTP_CTRL_0 | RW | DDR VTP Control Register | 0x67 | 0xE0C |
| DDR_VTP_CTRL_1 | RW | DDR VTP Control Register | 0x67 | 0xE10 |

#### 7.8.3.1 EMIF0/1 Clock Gate Control Register

The EMIF0/1 Clock Gate Control Register ( EMIF_CLK_GATE) described in the figure and table below.

**Figure 7-43. EMIF0/1 Clock Gate Control Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | DDR1_CKE_ STATUS | DDR0_CKE_ STATUS | DDRPHY1_ CLK_GATE | DDRPHY0_ CLK_GATE |
| R-0 | | R-1 | R-1 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 7-78. EMIF0/1 Clock Gate Control Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | DDR1_CKE_ STATUS | 1 | CKE status for DDR1 |
| 2 | DDR0_CKE_ STATUS | 1 | CKE status for DDR0 |
| 1 | DDRPHY1_ CLK_GATE | | Enables/disables the clock to the DDR1 PHY . |
| | | 0 | DDR1 PHY clock enabled (running) |
| | | 1 | DDR1 PHY clock disabled (stopped) |
| 0 | DDRPHY0_ CLK_GATE | | Enables/disables the clock to the DDR0 PHY |
| | | 0 | DDR0 PHY clock enabled (running) |
| | | 1 | DDR0 PHY clock disabled (stopped) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.3.2 DDR Memory Controller0_IO Control Register (DDR0_IO_CTRL)

The DDR Memory Controller_0 IO Control Register (DDR0_IO_CTRL) is shown in the figure and table below.

**Figure 7-44. DDR Memory Controller0_ IO Control Register (DDR0_IO_CTRL)**

| 31 | 30 | 29 | 21 | 20 | 19 | 18 | 16 |
|---|---|---|---|---|---|---|---|
| DDR3_RST_DEF_VAL | DDR_WUCLK_DISABLE | Reserved | | CS_SLEW | | CS_impedance | |
| R/W-0h | R/W-0h | R-10h | | R/W-2h | | R/W-3h | |

| 15 | 13 | 12 | 11 | 10 | 8 | 7 | 5 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | CMD_SLEW | | CMD_impedance | | Reserved | | DATA_SLEW | | DATA_impedance | |
| R-0 | | R/W-2h | | R/W-3h | | R-0 | | R/W-2h | | R/W-3h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-79. DDR Memory Controller0_IO Control Register (DDR0_IO_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | DDR3_RST_DEF_VAL | | DDR3 reset control |
| | | 0 | Normal operation. If a reset occurs on the DDR controller, the DDR[x]_RST signal will also go active (low) to reset the DDR3 memory. |
| | | 1 | If a reset occurs on the DDR controller, the DDR[x]_RST signal will NOT go active (low). |
| 30 | DDR_WUCLK_DISABLE | | Disables the internal 32 kHz wakeup clock used to clock the DDR memory controller subsystem before the DDR controller is configured. This clock may be disabled once the DDR controller has been configured, enabled and DDR initialization is complete. |
| | | 0 | Wakeup clock to the DDR controller is enabled (running) |
| | | 1 | Wakeup clock to the DDR controller is disabled (stopped) |
| 29-21 | Reserved | | Reserved |
| 20-19 | CS_SLEW [1:0] | 2h | Selects the slew rate of CS pins. Recommended slew rate value is 0x0 |
| 18-16 | CS_impedance [2:0] | 3h | Selects the Output impedance(Drive Strength) of chip select pin. Please refer to Table1-14 for the supported O/P impedance values.<br>Recommended O/P impedance value is 0x3 (8mA Drive strength) |
| 15-13 | Reserved | | Reserved |
| 12:11 | CMD_SLEW [1:0] | 2h | Selects the slew rate of command/address pins. Recommended slew rate value is 0x0. |
| 10-8 | CMD_impedance [2:0] | 3h | Selects the Output impedance(Drive Strength) of command/address pins. Please refer to Table1-14 for the supported O/P impedance values.<br>Recommended O/P impedance value is 0x3 (8mA Drive strength) |
| 7-5 | Reserved | | Reserved |
| 4-3 | DATA_SLEW [1:0] | 2h | Selects the slew rate of data pins. Recommended slew rate value is 0x0 |
| 2-0 | DATA_impedance [2:0] | 3h | Selects the Output impedance(Drive Strength) of Data pins. Please refer to Table1-14 for the supported O/P impedance values.<br>Recommended O/P impedance value is 0x3 (8mA Drive strength) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.3.3 DDR Memory Controller1_ IO Control Register (DDR1_IO_CTRL)

The DDR Memory Controller_1 IO Control Register (DDR1_IO_CTRL) is shown in the figure and table below.

**Figure 7-45. DDR Memory Controller1_ IO Control Register (DDR1_IO_CTRL)**

| 31 | 30 | 29 | 21 | 20 | 19 | 18 | 16 |
|---|---|---|---|---|---|---|---|
| DDR3_RST_DEF_VAL | DDR_WUCLK_DISABLE | Reserved | | CS_SLEW | | CS_impedance | |
| R/W-0h | R/W-0h | R-10h | | R/W-2h | | R/W-3h | |

| 15 | 13 | 12 | 11 | 10 | 8 | 7 | 5 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | CMD_SLEW | | CMD_impedance | | Reserved | | DATA_SLEW | | DATA_impedance | |
| R-0 | | R/W-2h | | R/W-3h | | R-0 | | R/W-2h | | R/W-3h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-80. DDR Memory Controller1_IO Control Register (DDR1_IO_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | DDR3_RST_DEF_VAL | | DDR3 reset control |
| | | 0 | Normal operation. If a reset occurs on the DDR controller, the DDR[x]_RST signal will also go active (low) to reset the DDR3 memory. |
| | | 1 | If a reset occurs on the DDR controller, the DDR[x]_RST signal will NOT go active (low). |
| 30 | DDR_WUCLK_DISABLE | | Disables the internal 32 kHz wakeup clock used to clock the DDR memory controller subsystem before the DDR controller is configured. This clock may be disabled once the DDR controller has been configured, enabled and DDR initialization is complete. |
| | | 0 | Wakeup clock to the DDR controller is enabled (running) |
| | | 1 | Wakeup clock to the DDR controller is disabled (stopped) |
| 29-21 | Reserved | | Reserved |
| 20-19 | CS_SLEW [1:0] | 2h | Selects the slew rate of CS pins. Recommended slew rate value is 0x0 |
| 18-16 | CS_impedance [2:0] | 3h | Selects the Output impedance(Drive Strength) of chip select pin. Please refer to Table1-14 for the supported O/P impedance values.<br>Recommended O/P impedance value is 0x3 (8mA Drive strength) |
| 15-13 | Reserved | | Reserved |
| 12:11 | CMD_SLEW [1:0] | 2h | Selects the slew rate of command/address pins. Recommended slew rate value is 0x0. |
| 10-8 | CMD_impedance [2:0] | 3h | Selects the Output impedance(Drive Strength) of command/address pins. Please refer to Table1-14 for the supported O/P impedance values.<br>Recommended O/P impedance value is 0x3 (8mA Drive strength) |
| 7-5 | Reserved | | Reserved |
| 4-3 | DATA_SLEW [1:0] | 2h | Selects the slew rate of data pins. Recommended slew rate value is 0x0 |
| 2-0 | DATA_impedance [2:0] | 3h | Selects the Output impedance(Drive Strength) of Data pins. Please refer to Table1-14 for the supported O/P impedance values.<br>Recommended O/P impedance value is 0x3 (8mA Drive strength) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.3.4 DDR VTP Control Register (DDR_VTP_CTRL_0)

The DDR VTP Control Register (DDR_VTP_CTRL_0) is shown in the figure and table below.

**Figure 7-46. DDR VTP Control Register (DDR_VTP_CTRL_0)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ENABLE | READY | LOCK | FILTER | | CLRz |
| R-0 | | R/W-1h | R/W-1h | R/W-0 | R/W-3h | | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-81. DDR VTP Control Register (DDR_VTP_CTRL_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | ENABLE | | VTP macro enable |
| | | 0 | VTP macro in bypass mode |
| | | 1 | Enabled |
| 5 | READY | | IO training sequence completion status |
| | | 0 | IO training is not completed |
| | | 1 | IO training is completed |
| 4 | LOCK | | Dynamic Update Locking |
| | | 0 | Dynamic update enabled based on VTP variations. |
| | | 1 | Dynamic impedance update is disabled and VTP macro powered down. |
| 3-1 | FILTER | 0 | Digital filter bits to control the update rate. |
| | | 1h | Update on 2 consecutive update requests |
| | | 2h | Update on 3 consecutive update requests |
| | | 3h | Update on 4 consecutive update requests |
| | | 4h | Update on 5 consecutive update requests |
| | | 5h | Update on 6 consecutive update requests |
| | | 6h | Update on 7 consecutive update requests |
| | | 7h | Update on 8 consecutive update requests |
| 0 | CLRz | | Clears the flops, start count again, after low going pulse |


Copyright © 2011–2016, Texas Instruments Incorporated

### 7.8.3.5 DDR VTP Control Register (DDR_VTP_CTRL_1)

The DDR VTP Control Register (DDR_VTP_CTRL_1) is shown in the figure and table below.

**Figure 7-47. DDR VTP Control Register 1 (DDR_VTP_CTRL_1)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ENABLE | READY | LOCK | FILTER | | CLRz |
| R-0 | | R/W-1h | R/W-1h | R/W-0 | R/W-3h | | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 7-82. DDR VTP Control Registerv 1 (DDR_VTP_CTRL_1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | ENABLE | | VTP macro enable |
| | | 0 | VTP macro in bypass mode |
| | | 1 | Enabled |
| 5 | READY | | IO training sequence completion status |
| | | 0 | IO training is not completed |
| | | 1 | IO training is completed |
| 4 | LOCK | | Dynamic Update Locking |
| | | 0 | Dynamic update enabled based on VTP variations. |
| | | 1 | Dynamic impedance update is disabled and VTP macro powered down. |
| 3-1 | FILTER | 0 | Digital filter bits to control the update rate. |
| | | 1h | Update on 2 consecutive update requests |
| | | 2h | Update on 3 consecutive update requests |
| | | 3h | Update on 4 consecutive update requests |
| | | 4h | Update on 5 consecutive update requests |
| | | 5h | Update on 6 consecutive update requests |
| | | 6h | Update on 7 consecutive update requests |
| | | 7h | Update on 8 consecutive update requests |
| 0 | CLRz | | Clears the flops, start count again, after low going pulse |

### 7.8.3.6 VTP Initialization

VTP-controller can be used in static or dynamic update modes. For both the modes VTP- controller is required to be initialized after every power-up cycle or for recalibration. The ‘READY’ bit reflects the status of initialization sequence completion.

To initialize the VTP-controller a negative going CLRz pulse should be asserted for at least 4ns. Once CLRz is removed, user needs to poll for READY bit to go high to know when the impedance gets fully trained. The READY bit stays high until CLRz is asserted again to re-initialize the training sequence.

When the LOCK bit is set Dynamic update ceases and VTP controller enters static mode. In order to save power, in static mode, VTP-controller goes into power-down mode. It is to be noted that in the LOCK mode, VTP-controller cannot be cleared (CLRz is disabled).

Table 7-83 shows the master connection IDs for the various initiators in the system, as seen by the DDR controllers.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 7.8.3.6.1 DDR Subsystem Level Master Connection IDs

Table 7-83 shows the connection IDs of different masters in the system as seen at DDR subsystem level.

**Table 7-83. Master Connection IDs**

| Master | MConnID |
|---|---|
| | (At DDR subsystem level ) |
| Cortex A8 | 0x0 |
| DSP | 0x20 |
| DSP-CFG | 0x24 |
| MMU | 0x28 |
| Media Controller | 0x38 |
| TPTC0_RD | 0x60 |
| TPTC0_WR | 0x64 |
| TPTC1_RD | 0x68 |
| TPTC1_WR | 0x6C |
| TPTC2_RD | 0x70 |
| TPTC2_WR | 0x74 |
| TPTC3_RD | 0x78 |
| TPTC3_WR | 0x7C |
| SGX | 0x80 |
| HDVPSS1 | 0x90 |
| HDVPSS2 | 0x94 |
| HDVICP | 0xA0 |
| ISS | 0xB0 |
| 3PSW | 0xC0 |
| FDIF | 0xC4 |
| USB1 | 0xD0 |
| USB2 | 0xD4 |
| PATA | 0xE0 |
| SATA | 0xE4 |
| PCIE | 0xE8 |


Copyright © 2011–2016, Texas Instruments Incorporated

# Enhanced Direct Memory Access Controller

This document describes the features and operations of the enhanced direct memory access (EDMA3) controller.

The EDMA3 is a high-performance, multichannel, multithreaded DMA controller that allows you to program a wide variety of transfer geometries and transfer sequences.

Section 8.1 provides an overview of features and terminology. Section 8.2 provides the details of the architecture and the common operations of the EDMA3 channel controller (EDMA3CC) and the EDMA3 transfer controller (EDMA3TC). Section 8.3 contains examples and common usage scenarios. describes the memory-mapped registers that are associated with the EDMA3 controller.

**Topic** **Page**


**8.1 Overview** ..... **1304**
**8.2 Features** ..... **1304**
**8.3 Terminology Used in This Document** ..... **1305**
**8.4 EDMA Transfer Examples** ..... **1350**
**8.5 EDMA3 Registers** ..... **1369**
**8.6 Appendix A** ..... **1446**
**8.7 Setting Up a Transfer** ..... **1448**



Copyright © 2011–2016, Texas Instruments Incorporated

## 8.1 Overview

The enhanced direct memory access (EDMA3) controller's primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the device.

Typical usage includes, but is not limited to the following:

- Servicing software-driven paging transfers (for example, transfers from external memory, such as DDR2 to internal device memory, such as DSP L2 SRAM).
- Servicing event-driven peripherals, such as a serial port.
- Performing sorting or sub-frame extraction of various data structures.
- Offloading data transfers from the main device CPU(s) or DSP(s) (see the device-specific data manual for specific peripherals that are accessible via the EDMA3 controller.).

The EDMA3 controller has a different architecture from the previous EDMA2 controller on the TMS320C621x/C671x DSPs and TMS320C64x DSPs. (See the *EDMA v3.0 (EDMA3) Migration Guide for TMS320C645x DSP* (SPRAAB9) for more information on new/advanced features.)

The EDMA3 controller consists of two principal blocks:

- EDMA3 channel controller (EDMA3CC).
- EDMA3 transfer controller(s) (EDMA3TC).

The EDMA3 channel controller serves as the user interface for the EDMA3 controller. The EDMA3CC includes parameter RAM (PaRAM), channel control registers, and interrupt control registers. The EDMA3CC serves to prioritize incoming software requests or events from peripherals and submits transfer requests (TRs) to the transfer controller.

The EDMA3 transfer controllers are slaves to the EDMA3 channel controller that is responsible for data movement. The transfer controller issues read/write commands to the source and destination addresses that are programmed for a given transfer. The operation is transparent to user.

## 8.2 Features

The EDMA3 channel controller has following features:

- Fully orthogonal transfer description:
  - Three transfer dimensions.
  - A-synchronized transfers: one-dimension serviced per event.
  - AB-synchronized transfers: two-dimensions serviced per event.
  - Independent indexes on source and destination.
  - Chaining feature allows a 3-D transfer based on a single event.
- Flexible transfer definition:
  - Increment or FIFO transfer addressing modes.
  - Linking mechanism allows automatic PaRAM set update.
  - Chaining allows multiple transfers to execute with one event.
- Interrupt generation for the following:
  - Transfer completion.
  - Error conditions.
- Debug visibility:
  - Queue water marking/threshold.
  - Error and status recording to facilitate debug.
- 64 DMA channels:
  - Event synchronization.
  - Manual synchronization (CPU(s) write to event set register).
  - Chain synchronization (completion of one transfer triggers another transfer).
- Eight QDMA channels:


Copyright © 2011–2016, Texas Instruments Incorporated

  - QDMA channels trigger automatically upon writing to a parameter RAM (PaRAM) set entry.
  - Support for programmable QDMA channel to PaRAM mapping.
- 512 PaRAM sets:
  - Each PaRAM set can be used for a DMA channel, QDMA channel, or link set.
- Four transfer controllers/event queues. You program the system-level priority of these queues. (See the device data manual for the possible system priorities.)
- 16 event entries per event queue.
- Memory protection support:
  - Proxy memory protection for TR submission.
  - Active memory protection for accesses to PaRAM and registers.

The EDMA3 transfer controller has the following features:

- Four transfer controllers (TC).
- 128-bit wide read and write ports per TC.
- Up to four in-flight transfer requests (TRs).
- Programmable priority level.
- Supports two-dimensional transfers with independent indexes on source and destination (EDMA3CC manages the 3rd dimension).
- Support for increment or constant addressing mode transfers.
- Interrupt and error support.
- Memory-Mapped Register (MMR) bit fields are fixed position in 32-bit MMR regardless of endianness.

## 8.3 Terminology Used in This Document

The following is a brief explanation of some terms that are used in this document:

| Term | Meaning |
|---|---|
| A-synchronized transfer | A transfer type where one dimension is serviced per synchronization event. |
| AB-synchronized transfer | A transfer type where two dimensions are serviced per synchronization event. |
| Chaining | A trigger mechanism in which a transfer can be initiated at the completion of another transfer or sub-transfer. |
| CPU(s) | The main processing engine or engines on a device. The CPU is typically a DSP or general-purpose processor (see the device-specific data manual to learn more about the CPU on your system.) |
| DMA channel | One of the 64 channels that external, manual, or chained events can trigger. All direct memory access (DMA) channels exist in the EDMA3CC. |
| Dummy set or dummy PaRAM set | A PaRAM set for which at least one of the count fields is equal to 0 and at least one of the count fields is nonzero. All of the count fields are cleared in a null PaRAM set. |
| Dummy transfer | A dummy set results in the EDMA3CC performing a dummy transfer. This is not an error condition. A null set results in an error condition. |
| EDMA3 channel controller (EDMA3CC) | The EDMA3CC is the portion of the EDMA3 that you program. The EDMA3CC contains the parameter RAM (PaRAM), event processing logic, DMA/QDMA channels, and event queues. The EDMA3CC service events (external, manual, chained, and QDMA) and is responsible for submitting transfer requests to the transfer controllers (EDMA3TC) that perform the actual transfer. |
| EDMA3 programmer | Any entity on the chip that has read/write access to the EDMA3 registers and can program an EDMA3 transfer. |
| EDMA3 transfer controller(s) (EDMA3TC) | Transfer controllers are the transfer engines for the EDMA3 controller. They perform the read/writes, as dictated by the EDMA3CC's transfer requests. |
| Enhanced direct memory access (EDMA3) controller | EDMA3 consists of the EDMA3 channel controller (EDMA3CC) and the EDMA3 transfer controller(s) (EDMA3TC), referred to as EDMA3 in this document. |
| L3 | System bus infrastructure that arbitrates and decodes bus transactions from multiple masters to multiple slaves. |
| Link parameter set | A PaRAM set that is used for linking. |



Copyright © 2011–2016, Texas Instruments Incorporated

| Term | Meaning |
|---|---|
| Linking | The mechanism of reloading a PaRAM set with new transfer characteristics on completion of the current transfer. |
| Memory-mapped slave | All on-chip memories, off-chip memories, and slave peripherals. These typically rely on the EDMA3 (or other master peripheral) to perform transfers to and from them. |
| Master peripherals | All peripherals that are capable of initiating read and write transfers to the system that may not solely rely on the EDMA3 for their data transfers. |
| Null set or null PaRAM set | A PaRAM set that has all count fields cleared (except for the link field). A dummy PaRAM set has at least one of the count fields nonzero. |
| Null transfer | A trigger event for a null PaRAM set results in the EDMA3CC performing a null transfer. This is an error condition. A dummy transfer is not an error condition. |
| Parameter RAM (PaRAM) | Programmable RAM that stores PaRAM sets that DMA channels, QDMA channels, and linking uses. |
| Parameter RAM (PaRAM) set | The PaRAM set is a 32-byte EDMA3 channel transfer definition. Each parameter set consists of eight words (that are four bytes each) that store the context for a DMA/QDMA/link transfer. A PaRAM set includes source address, destination address, counts, indexes, and options. |
| Parameter RAM (PaRAM) set entry | A PaRAM set entry is one of the eight four-byte components of the parameter set. |
| QDMA channel | A QDMA channel is one of the channels that you can trigger when writing to the trigger word (TRWORD) of a PaRAM set. All QDMA channels exist in the EDMA3CC. |
| Slave end points | Slave end points are all on-chip memories, off-chip memories, and slave peripherals. Slave end points may rely on the EDMA3 to perform transfers to and from them. |
| Transfer request (TR) | A command for data movement that is issued from the EDMA3CC to the EDMA3TC. A TR includes source and destination addresses, counts, indexes, and options. |
| Trigger event | A trigger event is an action that causes the EDMA3CC to service the channel and to submit a transfer request to the EDMA3TC. Trigger events for the DMA channels include events that are triggered manually, externally, and by chain. Trigger events for QDMA channels include events that are triggered automatically and by link. |
| Trigger word | For QDMA channels, the trigger word specifies the PaRAM set entry that results in a QDMA trigger event when it is written. The trigger word is programmed via the QDMA channel map register (QCHMAP) and can point to any of the PaRAM set entries. |
| TR synchronization (sync) event | See Trigger event. |


Copyright © 2011–2016, Texas Instruments Incorporated

# EDMA3 Architecture

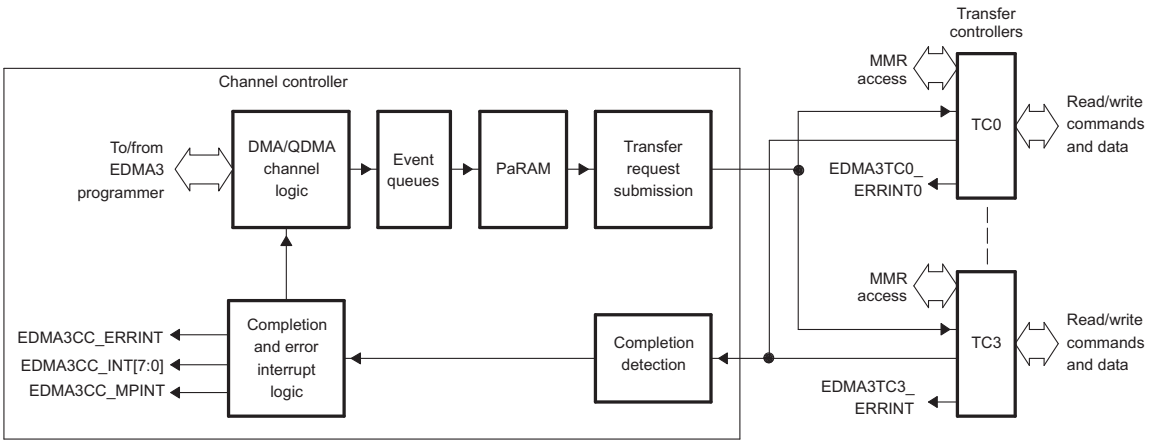
This chapter discusses the architecture of the EDMA3 controller.

## 8.3.1 Functional Overview

### 8.3.1.1 EDMA3 Controller Block Diagram

Figure 8-1 shows a block diagram for the EDMA3 controller.

**Figure 8-1. EDMA3 Controller Block Diagram**




### 8.3.1.2 EDMA3 Channel Controller (EDMA3CC)

Figure 8-2 shows a functional block diagram of the EDMA3 channel controller (EDMA3CC).

The main blocks of the EDMA3CC are as follows:

- Parameter RAM (PaRAM): The PaRAM maintains parameter sets for channel and reload parameter sets. You must write the PaRAM with the transfer context for the desired channels and link parameter sets. EDMA3CC processes sets based on a trigger event and submits a transfer request (TR) to the transfer controller.
- EDMA3 event and interrupt processing registers: Allows mapping of events to parameter sets, enable/disable events, enable/disable interrupt conditions, and clearing interrupts.
- Completion detection: The completion detect block detects completion of transfers by the EDMA3TC and/or slave peripherals. You can optionally use completion of transfers to chain trigger new transfers or to assert interrupts.
- Event queues: Event queues form the interface between the event detection logic and the transfer request submission logic.
- Memory protection registers: Memory protection registers define the accesses (privilege level and requestor(s)) that are allowed to access the DMA channel shadow region view(s) and regions of PaRAM.

Other functions include the following:

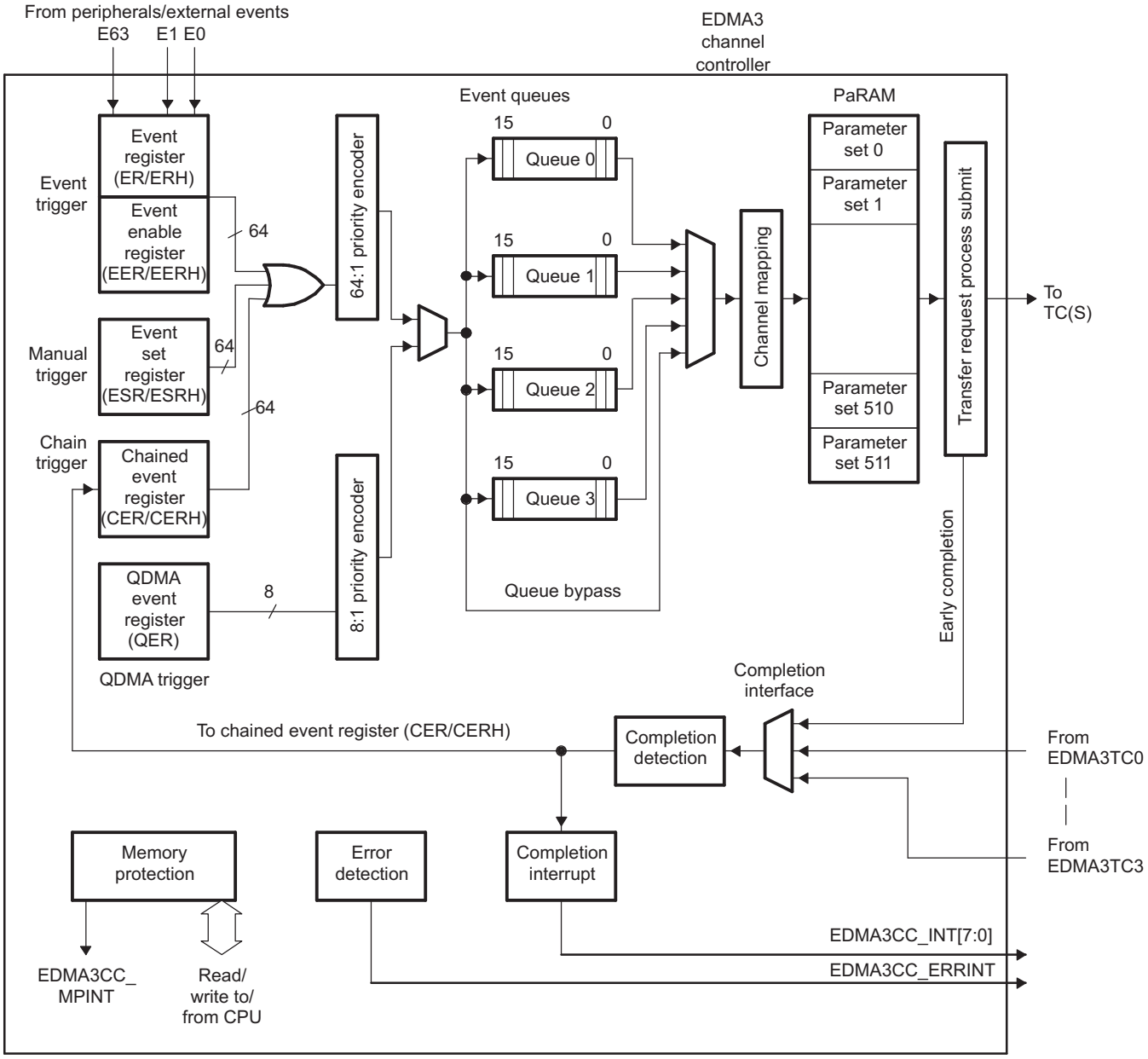
- Region registers: Region registers allow DMA resources (DMA channels and interrupts) to be assigned to unique regions that different EDMA3 programmers own (for example, ARM or DSP).
- Debug registers: Debug registers allow debug visibility by providing registers to read the queue status, controller status, and missed event status.

The EDMA3CC includes two channel types: DMA channels (64 channels) and QDMA channels (8 channels).

Each channel is associated with a given event queue/transfer controller and with a given PaRAM set. The main thing that differentiates a DMA channel from a QDMA channel is the method that the system uses to trigger transfers. See Section 8.3.4.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-2. EDMA3 Channel Controller (EDMA3CC) Block Diagram**




A trigger event is needed to initiate a transfer. A trigger event may be due to an external event, manual write to the event set register, or chained event for DMA channels. QDMA channels auto-trigger when a write to the trigger word that you program occurs on the associated PaRAM set. All such trigger events are logged into appropriate registers upon recognition.

Once a trigger event is recognized, the appropriate event gets queued in the EDMA3CC event queue. The assignment of each DMA/QDMA channel to an event queue is programmable. Each queue is 16 events deep; therefore, you can queue up to 16 events (on a single queue) in the EDMA3CC at a time. Additional pending events that are mapped to a full queue are queued when the event queue space becomes available. See Section 8.3.11.

If events on different channels are detected simultaneously, the events are queued based on a fixed priority arbitration scheme with the DMA channels being higher priority events than the QDMA channels. Among the two groups of channels, the lowest-numbered channel is the highest priority.


Copyright © 2011–2016, Texas Instruments Incorporated

Each event in the event queue is processed in FIFO order. When the head of the queue is reached, the PaRAM associated with that channel is read to determine the transfer details. The TR submission logic evaluates the validity of the TR and is responsible for submitting a valid transfer request (TR) to the appropriate EDMA3TC (based on the event queue to the EDMA3TC association, Q0 goes to TC0 , Q1 goes to TC1, Q2 goes to TC2, and Q3 goes to TC3). For more information, refer to Section 8.3.3.

The EDMA3TC receives the request and is responsible for data movement, as specified in the transfer request packet (TRP), other necessary tasks like buffering, and ensuring transfers are carried out in an optimal fashion wherever possible. For more information on EDMA3TC, refer to Section 8.3.1.3.

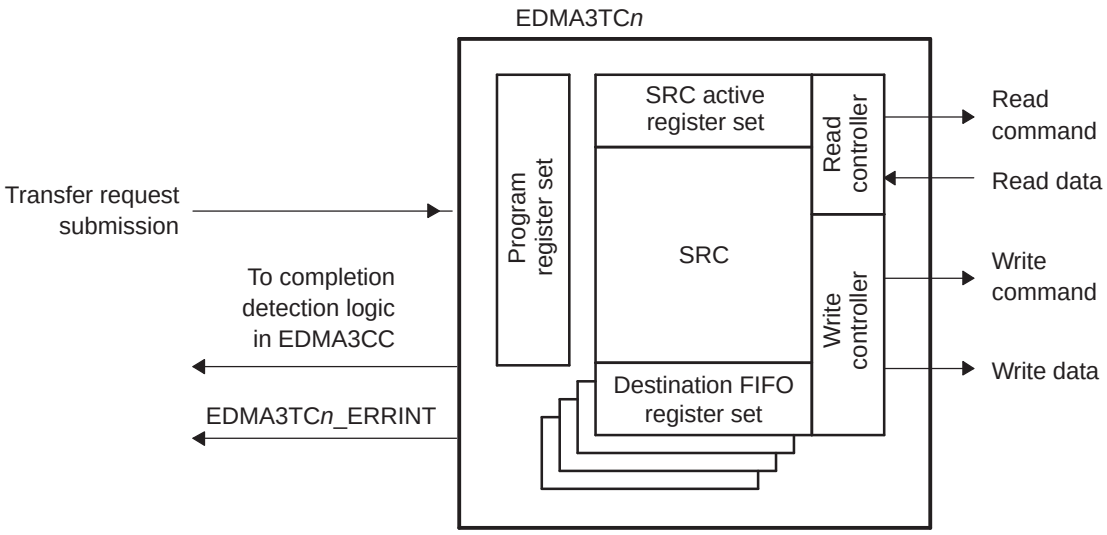
If you have decided to receive an interrupt or to chain to another channel on completion of the current transfer, the EDMA3TC signals completion to the EDMA3CC completion detection logic when the transfer is complete. You can alternately choose to trigger completion when a TR leaves the EDMA3CC boundary, rather than wait for all of the data transfers to complete. Based on the setting of the EDMA3CC interrupt registers, the completion interrupt generation logic is responsible for generating EDMA3CC completion interrupts to the CPU. For more information, refer to Section 8.3.5.

Additionally, the EDMA3CC also has an error detection logic that causes an error interrupt generation on various error conditions (like missed events, exceeding event queue thresholds, etc.). For more information on error interrupts, refer to Section 8.3.9.4.

### 8.3.1.3 EDMA3 Transfer Controller (EDMA3TC)

Section 8.3.9.4 shows a functional block diagram of the EDMA3 transfer controller (EDMA3TC).

**Figure 8-3. EDMA3 Transfer Controller (EDMA3TC) Block Diagram**




The main blocks of the EDMA3TC are:

- DMA program register set: The DMA program register set stores the transfer requests received from the EDMA3 channel controller (EDMA3CC).
- DMA source active register set: The DMA source active register set stores the context for the DMA transfer request currently in progress in the read controller.
- Read controller: The read controller issues read commands to the source address.
- Destination FIFO register set: The destination (DST) FIFO register set stores the context for the DMA transfer request(s) currently in progress in the write controller.
- Write controller: The write controller issues write commands/write data to the destination slave.
- Data FIFO: The data FIFO exists for holding temporary in-flight data.
- Completion interface: The completion interface sends completion codes to the EDMA3CC when a transfer completes, and generates interrupts and chained events (also, see Section 8.3.1.2 for more information on transfer completion reporting).


Copyright © 2011–2016, Texas Instruments Incorporated

When the EDMA3TC is idle and receives its first TR, DMA program register set receives the TR, where it transitions to the DMA source active set and the destination FIFO register set immediately. The second TR (if pending from EDMA3CC) is loaded into the DMA program set, ensuring it can start as soon as possible when the active transfer completes. As soon as the current active set is exhausted, the TR is loaded from the DMA program register set into the DMA source active register set as well as to the appropriate entry in the destination FIFO register set.

The read controller issues read commands governed by the rules of command fragmentation and optimization. These are issued only when the data FIFO has space available for the data read. When sufficient data is in the data FIFO, the write controller starts issuing a write command again following the rules for command fragmentation and optimization. For more information on command fragmentation and optimization, refer to Section 8.3.12.1.1.

Depending on the number of entries, the read controller can process up to two or four transfer requests ahead of the destination subject to the amount of free data FIFO.

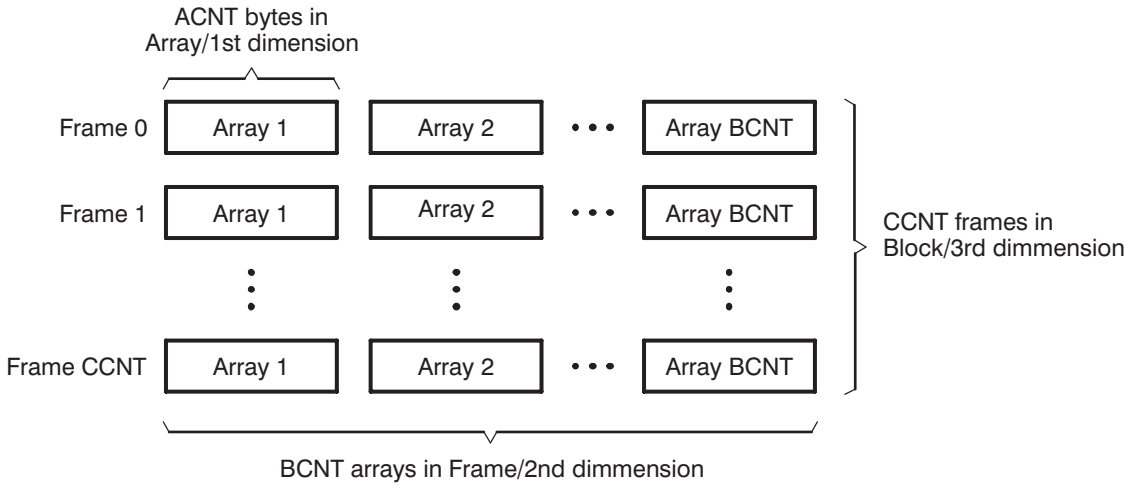
### 8.3.2 Types of EDMA3 Transfers

An EDMA3 transfer is always defined in terms of three dimensions. Figure 8-4 shows the three dimensions used by EDMA3 transfers. These three dimensions are defined as:

- 1st Dimension or Array (A): The 1st dimension in a transfer consists of ACNT contiguous bytes.
- 2nd Dimension or Frame (B): The 2nd dimension in a transfer consists of BCNT arrays of ACNT bytes. Each array transfer in the 2nd dimension is separated from each other by an index programmed using SRCBIDX or DSTBIDX.
- 3rd Dimension or Block (C): The 3rd dimension in a transfer consists of CCNT frames of BCNT arrays of ACNT bytes. Each transfer in the 3rd dimension is separated from the previous by an index programmed using SRCCIDX or DSTCIDX.

Note that the reference point for the index depends on the synchronization type. The amount of data transferred upon receipt of a trigger/synchronization event is controlled by the synchronization types (SYNCDIM bit in OPT). Of the three dimensions, only two synchronization types are supported: A-synchronized transfers and AB-synchronized transfers.

**Figure 8-4. Definition of ACNT, BCNT, and CCNT**





Copyright © 2011–2016, Texas Instruments Incorporated

### 8.3.2.1 A-Synchronized Transfers

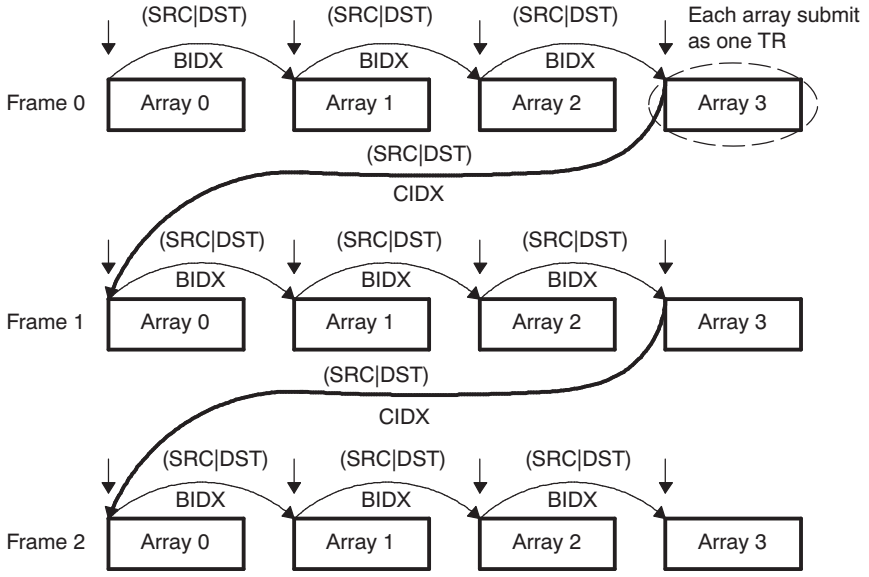
In an A-synchronized transfer, each EDMA3 sync event initiates the transfer of the 1st dimension of ACNT bytes, or one array of ACNT bytes. In other words, each event/TR packet conveys the transfer information for one array only. Thus, BCNT × CCNT events are needed to completely service a PaRAM set.

Arrays are always separated by SRCBIDX and DSTBIDX, as shown in Figure 8-5, where the start address of Array N is equal to the start address of Array N – 1 plus source (SRC) or destination (DST) BIDX.

Frames are always separated by SRCCIDX and DSTCIDX. For A-synchronized transfers, after the frame is exhausted, the address is updated by adding SRCCIDX/DSTCIDX to the beginning address of the last array in the frame. As in Figure 8-5, SRCCIDX/DSTCIDX is the difference between the start of Frame 0 Array 3 to the start of Frame 1 Array 0.

Figure 8-5 shows an A-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of n (ACNT) bytes. In this example, a total of 12 sync events (BCNT × CCNT) exhaust a PaRAM set. See Section 8.3.3.6 for details on parameter set updates.

**Figure 8-5. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.3.2.2 AB-Synchronized Transfers

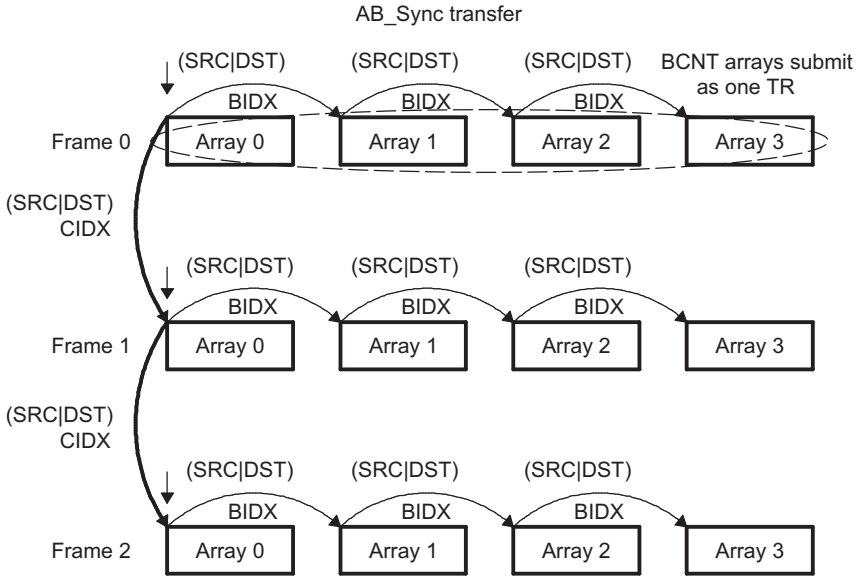
In a AB-synchronized transfer, each EDMA3 sync event initiates the transfer of 2 dimensions or one frame. In other words, each event/TR packet conveys information for one entire frame of BCNT arrays of ACNT bytes. Thus, CCNT events are needed to completely service a PaRAM set.

Arrays are always separated by SRCBIDX and DSTBIDX as shown in Figure 8-6. Frames are always separated by SRCCIDX and DSTCIDX.

Note that for AB-synchronized transfers, after a TR for the frame is submitted, the address update is to add SRCCIDX/DSTCIDX to the beginning address of the beginning array in the frame. This is different from A-synchronized transfers where the address is updated by adding SRCCIDX/DSTCIDX to the start address of the last array in the frame. See Section 8.3.3.6 for details on parameter set updates.

Figure 8-6 shows an AB-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of *n* (ACNT) bytes. In this example, a total of 3 sync events (CCNT) exhaust a PaRAM set; that is, a total of 3 transfers of 4 arrays each completes the transfer.

**Figure 8-6. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)**




**NOTE:** ABC-synchronized transfers are not directly supported. But can be logically achieved by chaining between multiple AB-synchronized transfers.

### *8.3.3 Parameter RAM (PaRAM)*

The EDMA3 controller is a RAM-based architecture. The transfer context (source/destination addresses, count, indexes, etc.) for DMA or QDMA channels is programmed in a parameter RAM table within EDMA3CC, referred to as PaRAM. The PaRAM table is segmented into multiple PaRAM sets. Each PaRAM set includes eight four-byte PaRAM set entries (32-bytes total per PaRAM set), which includes typical DMA transfer parameters such as source address, destination address, transfer counts, indexes, options, etc.

The PaRAM structure supports flexible ping-pong, circular buffering, channel chaining, and auto-reloading (linking).

The contents of the PaRAM include the following:

- 512 PaRAM sets
- 64 channels that are direct mapped and can be used as link or QDMA sets if not used for DMA channels
- 64 channels remain for link or QDMA sets


Copyright © 2011–2016, Texas Instruments Incorporated

By default, all channels map to PaRAM set to 0. These should be remapped before use. For more information, see Section 8.5.1.1.3 (DCHMAP registers) and Section 8.5.1.1.4 (QCHMAP registers).

**Table 8-1. EDMA3 Parameter RAM Contents**

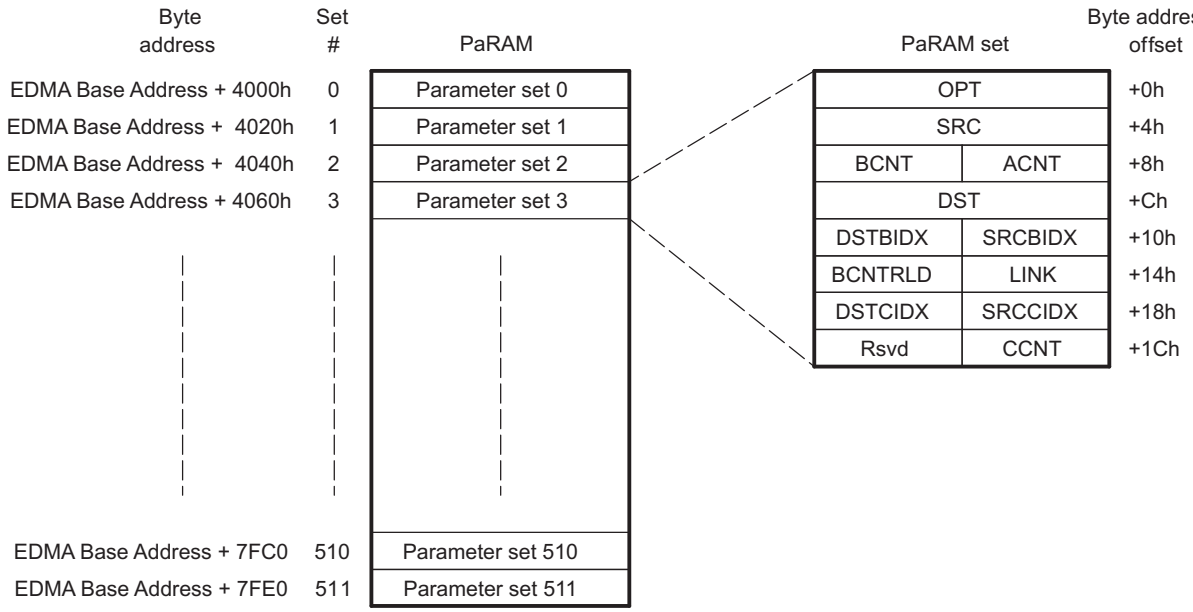
| PaRAM Set Number | Address[1] | Parameters[2] |
|---|---|---|
| 0 | EDMA Base Address + 4000h to EDMA Base Address + 401Fh | PaRAM set 0 |
| 1 | EDMA Base Address + 4020h to EDMA Base Address + 403Fh | PaRAM set 1 |
| 2 | EDMA Base Address + 4040h to EDMA Base Address + 405Fh | PaRAM set 2 |
| 3 | EDMA Base Address + 4060h to EDMA Base Address + 407Fh | PaRAM set 3 |
| 4 | EDMA Base Address + 4080h to EDMA Base Address + 409Fh | PaRAM set 4 |
| 5 | EDMA Base Address + 40A0h to EDMA Base Address + 40BFh | PaRAM set 5 |
| 6 | EDMA Base Address + 40C0h to EDMA Base Address + 40DFh | PaRAM set 6 |
| 7 | EDMA Base Address + 40E0h to EDMA Base Address + 40FFh | PaRAM set 7 |
| 8 | EDMA Base Address + 4100h to EDMA Base Address + 411Fh | PaRAM set 8 |
| 9 | EDMA Base Address + 4120h to EDMA Base Address + 413Fh | PaRAM set 9 |
| : | : | : |
| 63 | EDMA Base Address + 47E0h to EDMA Base Address + 47FFh | PaRAM set 63 |
| 64 | EDMA Base Address + 4800h to EDMA Base Address + 481Fh | PaRAM set 64 |
| 65 | EDMA Base Address + 4820h to EDMA Base Address + 483Fh | PaRAM set 65 |
| : | : | : |
| 510 | EDMA Base Address + 7FC0h to EDMA Base Address + 7FDFh | PaRAM set 510 |
| 511 | EDMA Base Address + 7FE0h to EDMA Base Address + 7FFFh | PaRAM set 511 |

(1) The EDMA base address can be obtained from the device-specific data manual. See EDMA TPCC Registers.
(2) The device has 8 QDMA channels that can be mapped to any parameter set number from 0 to 511.

### 8.3.3.1 PaRAM

Each parameter set of PaRAM is organized into eight 32-bit words or 32 bytes, as shown in Figure 8-7 and described in Table 8-2. Each PaRAM set consists of 16-bit and 32-bit parameters.

**Figure 8-7. PaRAM Set**





Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-2. EDMA3 Channel Parameter Description**

| Offset Address (bytes) | Acronym | Parameter | Description |
|---|---|---|---|
| 0h | OPT | Channel Options | Transfer configuration options |
| 4h | SRC | Channel Source Address | The byte address from which data is transferred |
| 8h(1) | ACNT | Count for 1st Dimension | Unsigned value specifying the number of contiguous bytes within an array (first dimension of the transfer). Valid values range from 1 to 65 535. |
| | BCNT | Count for 2nd Dimension | Unsigned value specifying the number of arrays in a frame, where an array is ACNT bytes. Valid values range from 1 to 65 535. |
| Ch | DST | Channel Destination Address | The byte address to which data is transferred |
| 10h(1) | SRCBIDX | Source BCNT Index | Signed value specifying the byte address offset between source arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767. |
| | DSTBIDX | Destination BCNT Index | Signed value specifying the byte address offset between destination arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767. |
| 14h(1) | LINK | Link Address | The PaRAM address containing the PaRAM set to be linked (copied from) when the current PaRAM set is exhausted. A value of FFFFh specifies a null link. |
| | BCNTRLD | BCNT Reload | The count value used to reload BCNT when BCNT decrements to 0 (TR is submitted for the last array in 2nd dimension). Only relevant in A-synchronized transfers. |
| 18h(1) | SRCCIDX | Source CCNT Index | Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767.<br>A-synchronized transfers: The byte address offset from the beginning of the last source array in a frame to the beginning of the first source array in the next frame.<br>AB-synchronized transfers: The byte address offset from the beginning of the first source array in a frame to the beginning of the first source array in the next frame. |
| | DSTCIDX | Destination CCNT index | Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767.<br>A-synchronized transfers: The byte address offset from the beginning of the last destination array in a frame to the beginning of the first destination array in the next frame.<br>AB-synchronized transfers: The byte address offset from the beginning of the first destination array in a frame to the beginning of the first destination array in the next frame. |
| 1Ch | CCNT | Count for 3rd Dimension | Unsigned value specifying the number of frames in a block, where a frame is BCNT arrays of ACNT bytes. Valid values range from 1 to 65 535. |
| | RSVD | Reserved | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |

(1) If OPT, SRC, or DST is the trigger word for a QDMA transfer then it is required to do a 32-bit access to that field. Furthermore, it is recommended to perform only 32-bit accesses on the parameter RAM for best code compatibility. For example, switching the endianness of the processor will swap addresses of the 16-bit fields, but 32-bit accesses avoid the issue entirely.


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.3.3.2 EDMA3 Channel PaRAM Set Entry Fields

#### *8.3.3.2.1 Channel Options Parameter (OPT)*

The channel options parameter (OPT) is shown in Figure 8-8 and described in Table 8-3.

**Figure 8-8. Channel Options Parameter (OPT)**

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |
| R-0 | R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TCC | TCCMODE | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

**Table 8-3. Channel Options Parameters (OPT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | PRIV | | Privilege level (supervisor versus user) for the host/CPU/DMA that programmed this PaRAM set. This value is set with the EDMA3 master's privilege value when any part of the PaRAM set is written. |
| | | 0 | User level privilege. |
| | | 1 | Supervisor level privilege. |
| 30-28 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 27-24 | PRIVID | 0-Fh | Privilege identification for the external host/CPU/DMA that programmed this PaRAM set. This value is set with the EDMA3 master's privilege identification value when any part of the PaRAM set is written. |
| 23 | ITCCHEN | | Intermediate transfer completion chaining enable. |
| | | 0 | Intermediate transfer complete chaining is disabled. |
| | | 1 | Intermediate transfer complete chaining is enabled. |
| | | | When enabled, the chained event register (CER/CERH) bit is set on every intermediate chained transfer completion (upon completion of every intermediate TR in the PaRAM set, except the final TR in the PaRAM set). The bit (position) set in CER or CERH is the TCC value specified. |
| 22 | TCCHEN | | Transfer complete chaining enable. |
| | | 0 | Transfer complete chaining is disabled. |
| | | 1 | Transfer complete chaining is enabled. |
| | | | When enabled, the chained event register (CER/CERH) bit is set on final chained transfer completion (upon completion of the final TR in the PaRAM set). The bit (position) set in CER or CERH is the TCC value specified. |
| 21 | ITCINTEN | | Intermediate transfer completion interrupt enable. |
| | | 0 | Intermediate transfer complete interrupt is disabled. |
| | | 1 | Intermediate transfer complete interrupt is enabled. |
| | | | When enabled, the interrupt pending register (IPR / IPRH) bit is set on every intermediate transfer completion (upon completion of every intermediate TR in the PaRAM set, except the final TR in the PaRAM set). The bit (position) set in IPR or IPRH is the TCC value specified. To generate a completion interrupt to the CPU, the corresponding IER [TCC] / IERH [TCC] bit must be set. |
| 20 | TCINTEN | | Transfer complete interrupt enable. |
| | | 0 | Transfer complete interrupt is disabled. |
| | | 1 | Transfer complete interrupt is enabled. |
| | | | When enabled, the interrupt pending register (IPR / IPRH) bit is set on transfer completion (upon completion of the final TR in the PaRAM set). The bit (position) set in IPR or IPRH is the TCC value specified. To generate a completion interrupt to the CPU, the corresponding IER[TCC] / IERH [TCC] bit must be set. |
| 19-18 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-3. Channel Options Parameters (OPT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 17-12 | TCC | 0-3Fh | Transfer complete code. This 6-bit code sets the relevant bit in the chaining enable register (CER [TCC] /CERH [TCC]) for chaining or in the interrupt pending register (IPR [TCC] / IPRH [TCC]) for interrupts. |
| 11 | TCCMODE | | Transfer complete code mode. Indicates the point at which a transfer is considered completed for chaining and interrupt generation. |
| | | 0 | Normal completion: A transfer is considered completed after the data has been transferred. |
| | | 1 | Early completion: A transfer is considered completed after the EDMA3CC submits a TR to the EDMA3TC. TC may still be transferring data when the interrupt/chain is triggered. |
| 10-8 | FWID | 0-7h | FIFO Width. Applies if either SAM or DAM is set to constant addressing mode. |
| | | 0 | FIFO width is 8-bit. |
| | | 1h | FIFO width is 16-bit. |
| | | 2h | FIFO width is 32-bit. |
| | | 3h | FIFO width is 64-bit. |
| | | 4h | FIFO width is 128-bit. |
| | | 5h | FIFO width is 256-bit. |
| | | 6h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-4 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3 | STATIC | | Static set. |
| | | 0 | Set is not static. The PaRAM set is updated or linked after a TR is submitted. A value of 0 should be used for DMA channels and for non-final transfers in a linked list of QDMA transfers. |
| | | 1 | Set is static. The PaRAM set is not updated or linked after a TR is submitted. A value of 1 should be used for isolated QDMA transfers or for the final transfer in a linked list of QDMA transfers. |
| 2 | SYNCDIM | | Transfer synchronization dimension. |
| | | 0 | A-synchronized. Each event triggers the transfer of a single array of ACNT bytes. |
| | | 1 | AB-synchronized. Each event triggers the transfer of BCNT arrays of ACNT bytes. |
| 1 | DAM | | Destination address mode. |
| | | 0 | Increment (INCR) mode. Destination addressing within an array increments. Destination is not a FIFO. |
| | | 1 | Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width. |
| 0 | SAM | | Source address mode. |
| | | 0 | Increment (INCR) mode. Source addressing within an array increments. Source is not a FIFO. |
| | | 1 | Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width. |

#### *8.3.3.2.2 Channel Source Address (SRC)*

The 32-bit source address parameter specifies the starting byte address of the source. For SAM in increment mode, there are no alignment restrictions imposed by EDMA3. For SAM in constant addressing mode, you must program the source address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). The EDMA3TC will signal an error, if this rule is violated. See Section 8.3.12.3 for additional details.

#### *8.3.3.2.3 Channel Destination Address (DST)*

The 32-bit destination address parameter specifies the starting byte address of the destination. For DAM in increment mode, there are no alignment restrictions imposed by EDMA3. For DAM in constant addressing mode, you must program the destination address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). The EDMA3TC will signal an error, if this rule is violated. See Section 8.3.12.3 for additional details.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.3.3.2.4 Count for 1st Dimension (ACNT)

ACNT represents the number of bytes within the 1st dimension of a transfer. ACNT is a 16-bit unsigned value with valid values between 0 and 65 535. Therefore, the maximum number of bytes in an array is 65 535 bytes (64K – 1 bytes). ACNT must be greater than or equal to 1 for a TR to be submitted to EDMA3TC. A transfer with ACNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in OPT.

See Section 8.3.3.5 and Section 8.3.5.3 for details on dummy/null completion conditions.

#### 8.3.3.2.5 Count for 2nd Dimension (BCNT)

BCNT is a 16-bit unsigned value that specifies the number of arrays of length ACNT. For normal operation, valid values for BCNT are between 1 and 65 535. Therefore, the maximum number of arrays in a frame is 65 535 (64K – 1 arrays). A transfer with BCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in OPT.

See Section 8.3.3.5 and Section 8.3.5.3 for details on dummy/null completion conditions.

#### 8.3.3.2.6 Count for 3rd Dimension (CCNT)

CCNT is a 16-bit unsigned value that specifies the number of frames in a block. Valid values for CCNT are between 1 and 65 535. Therefore, the maximum number of frames in a block is 65 535 (64K – 1 frames). A transfer with CCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in OPT.

A CCNT value of 0 is considered either a null or dummy transfer. See Section 8.3.3.5 and Section 8.3.5.3 for details on dummy/null completion conditions.

#### 8.3.3.2.7 BCNT Reload (BCNTRLD)

BCNTRLD is a 16-bit unsigned value used to reload the BCNT field once the last array in the 2nd dimension is transferred. This field is only used for A-synchronized transfers. In this case, the EDMA3CC decrements the BCNT value by 1 on each TR submission. When BCNT reaches 0, the EDMA3CC decrements CCNT and uses the BCNTRLD value to reinitialize the BCNT value.

For AB-synchronized transfers, the EDMA3CC submits the BCNT in the TR and the EDMA3TC decrements BCNT appropriately. For AB-synchronized transfers, BCNTRLD is not used.

#### 8.3.3.2.8 Source B Index (SRCBIDX)

SRCBIDX is a 16-bit signed value (2s complement) used for source address modification between each array in the 2nd dimension. Valid values for SRCBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the source array to the beginning of the next source array. It applies to both A-synchronized and AB-synchronized transfers. Some examples:

- SRCBIDX = 0000h (0): no address offset from the beginning of an array to the beginning of the next array. All arrays are fixed to the same beginning address.
- SRCBIDX = 0003h (+3): the address offset from the beginning of an array to the beginning of the next array in a frame is 3 bytes. For example, if the current array begins at address 1000h, the next array begins at 1003h.
- SRCBIDX = FFFFh (–1): the address offset from the beginning of an array to the beginning of the next array in a frame is –1 byte. For example, if the current array begins at address 5054h, the next array begins at 5053h.

#### 8.3.3.2.9 Destination B Index (DSTBIDX)

DSTBIDX is a 16-bit signed value (2s complement) used for destination address modification between each array in the 2nd dimension. Valid values for DSTBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the destination array to the beginning of the next destination array within the current frame. It applies to both A-synchronized and AB-synchronized transfers. See SRCBIDX for examples.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.3.3.2.10 Source C Index (SRCCIDX)

SRCCIDX is a 16-bit signed value (2s complement) used for source address modification in the 3rd dimension. Valid values for SRCCIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by SRC address) to the beginning of the first source array in the next frame. It applies to both A-synchronized and AB-synchronized transfers. Note that when SRCCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame (Figure 8-5), while the current array in an AB-synchronized transfer is the first array in the frame (Figure 8-6).

#### 8.3.3.2.11 Destination C Index (DSTCIDX)

DSTCIDX is a 16-bit signed value (2s complement) used for destination address modification in the 3rd dimension. Valid values are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by DST address) to the beginning of the first destination array TR in the next frame. It applies to both A-synchronized and AB-synchronized transfers. Note that when DSTCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame (Figure 8-5), while the current array in a AB-synchronized transfer is the first array in the frame (Figure 8-6).

#### 8.3.3.2.12 Link Address (LINK)

The EDMA3CC provides a mechanism, called linking, to reload the current PaRAM set upon its natural termination (that is, after the count fields are decremented to 0) with a new PaRAM set. The 16-bit parameter LINK specifies the byte address offset in the PaRAM from which the EDMA3CC loads/reloads the next PaRAM set during linking.

You must program the link address to point to a valid aligned 32-byte PaRAM set. The 5 LSBs of the LINK field should be cleared to 0.

The EDMA3CC ignores the upper 2 bits of the LINK entry, allowing the programmer the flexibility of programming the link address as either an absolute/literal byte address or use the PaRAM-base-relative offset address. Therefore, if you make use of the literal address with a range from 4000h to 7FFFh, it will be treated as a PaRAM-base-relative value of 0000h to 3FFFh.

You should make sure to program the LINK field correctly, so that link update is requested from a PaRAM address that falls in the range of the available PaRAM addresses on the device.

A LINK value of FFFFh is referred to as a NULL link that should cause the EDMA3CC to perform an internal write of 0 to all entries of the current PaRAM set, except for the LINK field that is set to FFFFh. Also, see Section 8.3.5 for details on terminating a transfer.

### 8.3.3.3 Null PaRAM Set

A null PaRAM set is defined as a PaRAM set where all count fields (ACNT, BCNT, and CCNT) are cleared to 0. If a PaRAM set associated with a channel is a NULL set, then when serviced by the EDMA3CC, the bit corresponding to the channel is set in the associated event missed register (EMR, EMRH, or QEMR). This bit remains set in the associated secondary event register (SER, SERH, or QSER). *This implies that any future events on the same channel are ignored by the EDMA3CC and you are required to clear the bit in SER, SERH, or QSER for the channel.* This is considered an error condition, since events are not expected on a channel that is configured as a null transfer. See Section 8.5.1.6.8 and Section 8.5.1.2.1 for more information on the SER and EMR registers, respectively.

### 8.3.3.4 Dummy PaRAM Set

A dummy PaRAM set is defined as a PaRAM set where at least one of the count fields (ACNT, BCNT, or CCNT) is cleared to 0 and at least one of the count fields is nonzero.

If a PaRAM set associated with a channel is a dummy set, then when serviced by the EDMA3CC, it will not set the bit corresponding to the channel (DMA/QDMA) in the event missed register (EMR, EMRH, or QEMR) and the secondary event register (SER, SERH, or QSER) bit gets cleared similar to a normal transfer. Future events on that channel are serviced. A dummy transfer is a legal transfer of 0 bytes. See Section 8.5.1.6.8 and Section 8.5.1.2.1 for more information on the SER and EMR registers, respectively.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.3.3.5 Dummy Versus Null Transfer Comparison

There are some differences in the way the EDMA3CC logic treats a dummy versus a null transfer request. A null transfer request is an error condition, but a dummy transfer is a legal transfer of 0 bytes. A null transfer causes an error bit (E*n*) in EMR to get set and the E*n* bit in SER remains set, essentially preventing any further transfers on that channel without clearing the associated error registers.

Table 8-4 summarizes the conditions and effects of null and dummy transfer requests.

**Table 8-4. Dummy and Null Transfer Request**

| Feature | Null TR | Dummy TR |
|---|---|---|
| EMR/EMRH/QEMR is set | Yes | No |
| SER/SERH/QSER remains set | Yes | No |
| Link update (STATIC = 0 in OPT) | Yes | Yes |
| QER is set | Yes | Yes |
| IPR/IPRH CER/CERH is set using early completion | Yes | Yes |

#### 8.3.3.6 Parameter Set Updates

When a TR is submitted for a given DMA/QDMA channel and its corresponding PaRAM set, the EDMA3CC is responsible for updating the PaRAM set in anticipation of the next trigger event. For events that are not final, this includes address and count updates; for final events, this includes the link update.

The specific PaRAM set entries that are updated depend on the channel's synchronization type (A-synchronized or B-synchronized) and the current state of the PaRAM set. A B-update refers to the decrementing of BCNT in the case of A-synchronized transfers after the submission of successive TRs. A C-update refers to the decrementing of CCNT in the case of A-synchronized transfers after BCNT TRs for ACNT byte transfers have submitted. For AB-synchronized transfers, a C-update refers to the decrementing of CCNT after submission of every transfer request.

See Table 8-5 for details and conditions on the parameter updates. A link update occurs when the PaRAM set is exhausted, as described in Section 8.3.3.7.

After the TR is read from the PaRAM (and is in process of being submitted to EDMA3TC), the following fields are updated if needed:

- A-synchronized: BCNT, CCNT, SRC, DST.
- AB-synchronized: CCNT, SRC, DST.

The following fields are not updated (except for during linking, where all fields are overwritten by the link PaRAM set):

- A-synchronized: ACNT, BCNTRLD, SRCBIDX, DSTBIDX, SRCCIDX, DSTCIDX, OPT, LINK.
- AB-synchronized: ACNT, BCNT, BCNTRLD, SRCBIDX, DSTBIDX, SRCCIDX, DSTCIDX, OPT, LINK.


Copyright © 2011–2016, Texas Instruments Incorporated

Note that PaRAM updates only pertain to the information that is needed to properly submit the next transfer request to the EDMA3TC. Updates that occur while data is moved within a transfer request are tracked within the transfer controller, and is detailed in Section 8.3.12. For A-synchronized transfers, the EDMA3CC always submits a TRP for ACNT bytes (BCNT = 1 and CCNT = 1). For AB-synchronized transfers, the EDMA3CC always submits a TRP for ACNT bytes of BCNT arrays (CCNT = 1). The EDMA3TC is responsible for updating source and destination addresses within the array based on ACNT and FWID (in OPT). For AB-synchronized transfers, the EDMA3TC is also responsible to update source and destination addresses between arrays based on SRCBIDX and DSTBIDX.

Table 8-5 shows the details of parameter updates that occur within EDMA3CC for A-synchronized and AB-synchronized transfers.

**Table 8-5. Parameter Updates in EDMA3CC (for Non-Null, Non-Dummy PaRAM Set)**

| | A-Synchronized Transfer | | | AB-Synchronized Transfer | | |
|---|---|---|---|---|---|---|
| | **B-Update** | **C-Update** | **Link Update** | **B-Update** | **C-Update** | **Link Update** |
| **Condition:** | **BCNT > 1** | **BCNT == 1 && CCNT > 1** | **BCNT == 1 && CCNT == 1** | **N/A** | **CCNT > 1** | **CCNT == 1** |
| SRC | += SRCBIDX | += SRCCIDX | = Link.SRC | in EDMA3TC | += SRCCIDX | = Link.SRC |
| DST | += DSTBIDX | += DSTCIDX | = Link.DST | in EDMA3TC | += DSTCIDX | = Link.DST |
| ACNT | None | None | = Link.ACNT | None | None | = Link.ACNT |
| BCNT | –= 1 | = BCNTRLD | = Link.BCNT | in EDMA3TC | N/A | = Link.BCNT |
| CCNT | None | –= 1 | = Link.CCNT | in EDMA3TC | –=1 | = Link.CCNT |
| SRCBIDX | None | None | = Link.SRCBIDX | in EDMA3TC | None | = Link.SRCBIDX |
| DSTBIDX | None | None | = Link.DSTBIDX | None | None | = Link.DSTBIDX |
| SRCCIDX | None | None | = Link.SRCBIDX | in EDMA3TC | None | = Link.SRCBIDX |
| DSTCIDX | None | None | = Link.DSTBIDX | None | None | = Link.DSTBIDX |
| LINK | None | None | = Link.LINK | None | None | = Link.LINK |
| BCNTRLD | None | None | = Link.BCNTRLD | None | None | = Link.BCNTRLD |
| OPT[1] | None | None | = LINK.OPT | None | None | = LINK.OPT |

(1) In all cases, no updates occur if OPT.STATIC == 1 for the current PaRAM set.

**NOTE:** The EDMA3CC includes no special hardware to detect when an indexed address update calculation overflows/underflows. The address update will wrap across boundaries as programmed by the user. You should ensure that no transfer is allowed to cross internal port boundaries between peripherals. A single TR must target a single source/destination slave endpoint.


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.3.3.7 Linking Transfers

The EDMA3CC provides a mechanism known as linking, which allows the entire PaRAM set to be reloaded from a location within the PaRAM memory map (for both DMA and QDMA channels). Linking is especially useful for maintaining ping-pong buffers, circular buffering, and repetitive/continuous transfers with no CPU intervention. Upon completion of a transfer, the current transfer parameters are reloaded with the parameter set pointed that the 16-bit link address field of the current parameter set points to. Linking only occurs when the STATIC bit in OPT is cleared.

**NOTE:** You should always link a transfer (EDMA3 or QDMA) to another useful transfer. If you must terminate a transfer, then you should link the transfer to a NULL parameter set. See Section 8.3.3.3.

The link update occurs after the current PaRAM set event parameters have been exhausted. An event's parameters are exhausted when the EDMA3 channel controller has submitted all of the transfers that are associated with the PaRAM set.

A link update occurs for null and dummy transfers depending on the state of the STATIC bit in OPT and the LINK field. In both cases (null or dummy), if the value of LINK is FFFFh, then a null PaRAM set (with all 0s and LINK set to FFFFh) is written to the current PaRAM set. Similarly, if LINK is set to a value other than FFFFh, then the appropriate PaRAM location that LINK points to is copied to the current PaRAM set.

Once the channel completion conditions are met for an event, the transfer parameters that are located at the link address are loaded into the current DMA or QDMA channel's associated parameter set. This indicates that the EDMA3CC reads the entire set (eight words) from the PaRAM set specified by LINK and writes all eight words to the PaRAM set that is associated with the current channel. Figure 8-9 shows an example of a linked transfer.

Any PaRAM set in the PaRAM can be used as a link/reload parameter set. The PaRAM sets associated with peripheral synchronization events (see Section 8.3.6) should only be used for linking if the corresponding events are disabled.

If a PaRAM set location is defined as a QDMA channel PaRAM set (by QCHMAP*n*), then copying the link PaRAM set into the current QDMA channel PaRAM set is recognized as a trigger event. It is latched in QER because a write to the trigger word was performed. You can use this feature to create a linked list of transfers using a single QDMA channel and multiple PaRAM sets. See Section 8.3.4.2.

Linking to itself replicates the behavior of auto-initialization, thus facilitating the use of circular buffering and repetitive transfers. After an EDMA3 channel exhausts its current PaRAM set, it reloads all of the parameter set entries from another PaRAM set, which is initialized with values that are identical to the original PaRAM set. Figure 8-10 shows an example of a linked to self transfer. Here, the PaRAM set 511 has the link field pointing to the address of parameter set 511 (linked to self).

**NOTE:** If the STATIC bit in OPT is set for a PaRAM set, then link updates are not performed.

### 8.3.3.8 Constant Addressing Mode Transfers/Alignment Issues

If either SAM or DAM is set (constant addressing mode), then the source or destination address must be aligned to a 256-bit aligned address, respectively, and the corresponding BIDX should be an even multiple of 32 bytes (256 bits). The EDMA3CC does not recognize errors here, but the EDMA3TC asserts an error if this is not true. See Section 8.3.12.3.

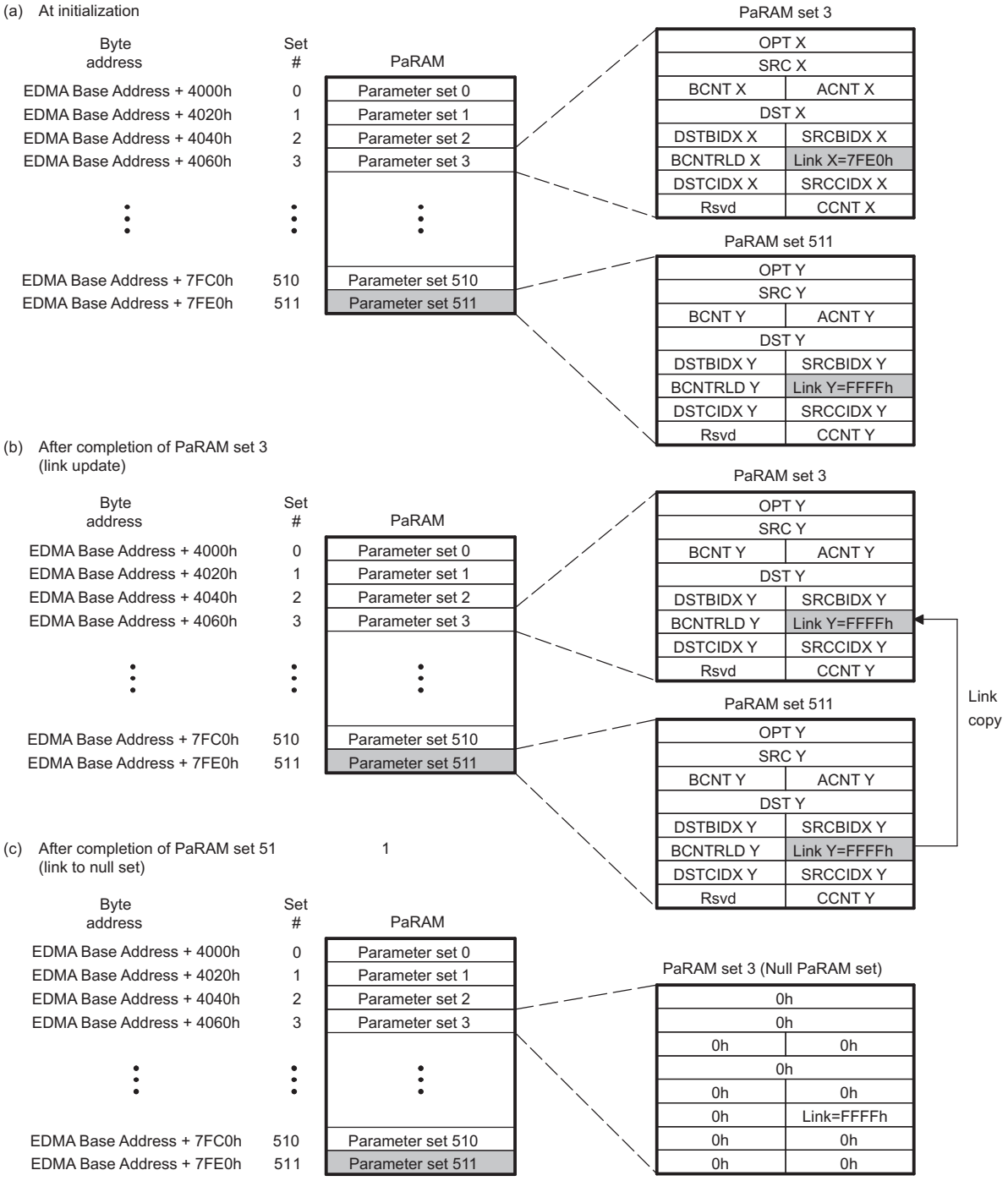
**NOTE:** The constant addressing (CONST) mode has limited applicability. The EDMA3 should be configured for the constant addressing mode (SAM/DAM = 1) only if the transfer source or destination (on-chip memory, off-chip memory controllers, slave peripherals) support the constant addressing mode. See the device-specific data manual and/or peripheral user`s guide to verify if the constant addressing mode is supported. If the constant addressing mode is not supported, the similar logical transfer can be achieved using the increment (INCR) mode (SAM/DAM =0) by appropriately programming the count and indices values.


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.3.3.9 Element Size

The EDMA3 controller does not use element-size and element-indexing. Instead, all transfers are defined in terms of all three dimensions: ACNT, BCNT, and CCNT. An element-indexed transfer is logically achieved by programming ACNT to the size of the element and BCNT to the number of elements that need to be transferred. For example, if you have 16-bit audio data and 256 audio samples that must be transferred to a serial port, you can only do this by programming the ACNT = 2 (2 bytes) and BCNT = 256.

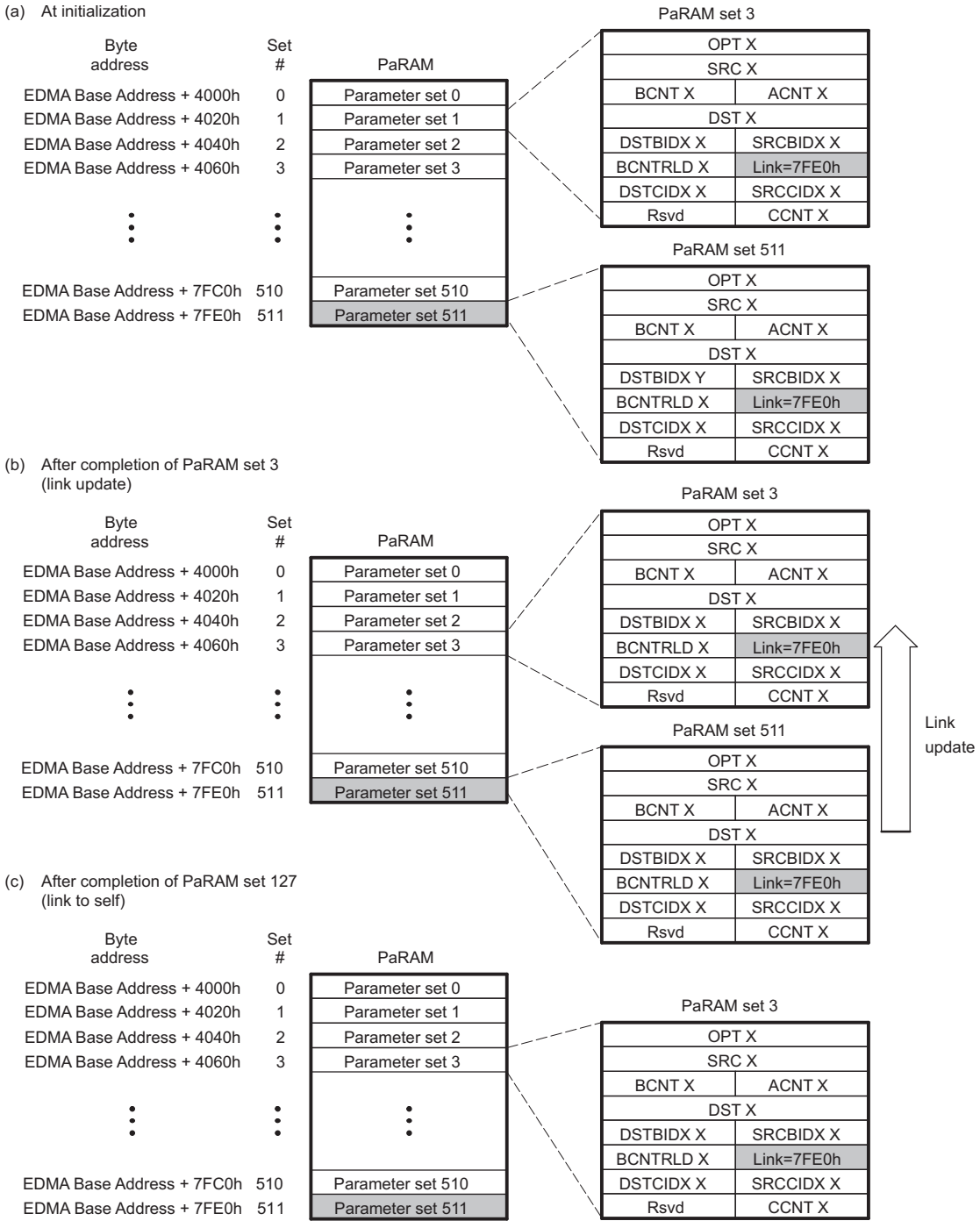
**Figure 8-9. Linked Transfer**






Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-10. Link-to-Self Transfer**




### 8.3.4 Initiating a DMA Transfer

There are multiple ways to initiate a programmed data transfer using the EDMA3 channel controller. Transfers on DMA channels are initiated by three sources.

They are listed as follows:

- **Event-triggered transfer request** (this is the more typical usage of EDMA3): A peripheral, system, or externally-generated event triggers a transfer request.
- **Manually-triggered transfer request:**The CPU manually triggers a transfer by writing a 1 to the


Copyright © 2011–2016, Texas Instruments Incorporated

corresponding bit in the event set register (ESR/ESRH).

- **Chain-triggered transfer request:** A transfer is triggered on the completion of another transfer or sub-transfer.

Transfers on QDMA channels are initiated by two sources. They are as follows:

- **Auto-triggered transfer request:** Writing to the programmed trigger word triggers a transfer.
- **Link-triggered transfer requests:** Writing to the trigger word triggers the transfer when linking occurs.

#### 8.3.4.1 DMA Channel

##### *8.3.4.1.1 Event-Triggered Transfer Request*

When an event is asserted from a peripheral or device pins, it gets latched in the corresponding bit of the event register (ER.E*n* = 1). For peripheral event to DMA event mapping, see the device-specific data manual. If the corresponding event in the event enable register (EER) is enabled (EER.E*n* = 1), then the EDMA3CC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

If the PaRAM set is valid (not a NULL set), then a transfer request packet (TRP) is submitted to the EDMA3TC and the E*n* bit in ER is cleared. At this point, a new event can be safely received by the EDMA3CC.

If the PaRAM set associated with the channel is a NULL set (see Section 8.3.3.3), then no transfer request (TR) is submitted and the corresponding E*n* bit in ER is cleared and simultaneously the corresponding channel bit is set in the event miss register (EMR.E*n* = 1) to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include cleaning the event missed error before re-triggering the DMA channel.

When an event is received, the corresponding event bit in the event register is set (ER.E*n* = 1), regardless of the state of EER.E*n*. If the event is disabled when an external event is received (ER.E*n* = 1 and EER.E*n* = 0), the ER.E*n* bit remains set. If the event is subsequently enabled (EER.E*n* = 1), then the pending event is processed by the EDMA3CC and the TR is processed/submitted, after which the ER.E*n* bit is cleared.

If an event is being processed (prioritized or is in the event queue) and another sync event is received for the same channel prior to the original being cleared (ER.E*n* != 0), then the second event is registered as a missed event in the corresponding bit of the event missed register (EMR.E*n* = 1).

The EDMA section in the device-specific data manual gives the table of the synchronization events associated with each of the programmable DMA channels in the device. It also provides the mapping of events to the EDMA event crossbar. See the device-specific data manual to determine the event to channel mapping.

##### *8.3.4.1.2 Manually-Triggered Transfer Request*

The CPU or any EDMA programmer initiates a DMA transfer by writing to the event set register (ESR). Writing a 1 to an event bit in the ESR results in the event being prioritized/queued in the appropriate event queue, regardless of the state of the EER.E*n* bit. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA3TC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see Section 8.3.3.3), then no transfer request (TR) is submitted and the corresponding E*n* bit in ER is cleared and simultaneously the corresponding channel bit is set in the event miss register (EMR.E*n* = 1) to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include clearing the event missed error before re-triggering the DMA channel.

If an event is being processed (prioritized or is in the event queue) and the same channel is manually set by a write to the corresponding channel bit of the event set register (ESR.E*n* = 1) prior to the original being cleared (ESR.E*n* = 0), then the second event is registered as a missed event in the corresponding bit of the event missed register (EMR.E*n* = 1).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.3.4.1.3 Chain-Triggered Transfer Request

Chaining is a mechanism by which the completion of one transfer automatically sets the event for another channel. When a chained completion code is detected, the value of which is dictated by the transfer completion code (TCC[5:0] in OPT of the PaRAM set associated with the channel), it results in the corresponding bit in the chained event register (CER) to be set (CER.E[TCC] = 1).

Once a bit is set in CER, the EDMA3CC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA3TC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see Section 8.3.3.3), then no transfer request (TR) is submitted and the corresponding E*n* bit in CER is cleared and simultaneously the corresponding channel bit is set in the event miss register (EMR.E*n* = 1) to indicate that the event was discarded due to a null TR being serviced. In this case, the error condition must be cleared by you before the DMA channel can be re-triggered. Good programming practices might include clearing the event missed error before re-triggering the DMA channel.

If a chaining event is being processed (prioritized or queued) and another chained event is received for the same channel prior to the original being cleared (CER.E*n* != 0), then the second chained event is registered as a missed event in the corresponding channel bit of the event missed register (EMR.E*n* = 1).

**NOTE:** Chained event registers, event registers, and event set registers operate independently. An event (E*n*) can be triggered by any of the trigger sources (event-triggered, manually-triggered, or chain-triggered).

### 8.3.4.2 QDMA Channels

#### 8.3.4.2.1 Auto-triggered and Link-Triggered Transfer Request

QDMA-based transfer requests are issued when a QDMA event gets latched in the QDMA event register (QER.E*n* = 1). A bit corresponding to a QDMA channel is set in the QDMA event register (QER) when the following occurs:

- A CPU (or any EDMA3 programmer) write occurs to a PaRAM address that is defined as a QDMA channel trigger word (programmed in the QDMA channel mapping register (QCHMAP*n*)) for the particular QDMA channel and the QDMA channel is enabled via the QDMA event enable register (QEER.E*n* = 1).
- EDMA3CC performs a link update on a PaRAM set address that is configured as a QDMA channel (matches QCHMAP*n* settings) and the corresponding channel is enabled via the QDMA event enable register (QEER.E*n* = 1).

Once a bit is set in QER, the EDMA3CC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA3TC and the channel can be triggered again.

If a bit is already set in QER (QER.E*n* = 1) and a second QDMA event for the same QDMA channel occurs prior to the original being cleared, the second QDMA event gets captured in the QDMA event miss register (QEMR.E*n* = 1).

### 8.3.4.3 Comparison Between DMA and QDMA Channels

The primary difference between DMA and QDMA channels is the event/channel synchronization. QDMA events are either auto-triggered or link triggered. Auto-triggering allows QDMA channels to be triggered by CPU(s) with a minimum number of linear writes to PaRAM. Link triggering allows a linked list of transfers to be executed, using a single QDMA PaRAM set and multiple link PaRAM sets.


Copyright © 2011–2016, Texas Instruments Incorporated

A QDMA transfer is triggered when a CPU (or other EDMA3 programmer) writes to the trigger word of the QDMA channel parameter set (auto-triggered) or when the EDMA3CC performs a link update on a PaRAM set that has been mapped to a QDMA channel (link triggered). Note that for CPU triggered (manually triggered) DMA channels, in addition to writing to the PaRAM set, it is required to write to the event set register (ESR) to kick-off the transfer.

QDMA channels are typically for cases where a single event will accomplish a complete transfer since the CPU (or EDMA3 programmer) must reprogram some portion of the QDMA PaRAM set in order to re-trigger the channel. In other words, QDMA transfers are programmed with BCNT = CCNT = 1 for A-synchronized transfers, and CCNT = 1 for AB-synchronized transfers.

Additionally, since linking is also supported (if STATIC = 0 in OPT) for QDMA transfers, it allows you to initiate a linked list of QDMAs, so when EDMA3CC copies over a link PaRAM set (including the write to the trigger word), the current PaRAM set mapped to the QDMA channel will automatically be recognized as a valid QDMA event and initiate another set of transfers as specified by the linked set.

### 8.3.5 Completion of a DMA Transfer

A parameter set for a given channel is complete when the required number of transfer requests is submitted (based on receiving the number of synchronization events). The expected number of TRs for a non-null/non-dummy transfer is shown in Table 8-6 for both synchronization types along with state of the PaRAM set prior to the final TR being submitted. When the counts (BCNT and/or CCNT) are this value, the next TR results in a:

- Final chaining or interrupt codes to be sent by the transfer controllers (instead of intermediate).
- Link updates (linking to either null or another valid link set).

**Table 8-6. Expected Number of Transfers for Non-Null Transfer**

| Sync Mode | Counts at time 0 | Total # Transfers | Counts prior to final TR |
|---|---|---|---|
| A-synchronized | ACNT<br>BCNT<br>CCNT | (BCNT × CCNT ) TRs of ACNT bytes each | BCNT == 1 && CCNT == 1 |
| AB-synchronized | ACNT<br>BCNT<br>CCNT | CCNT TRs for ACNT × BCNT bytes each | CCNT == 1 |

You must program the PaRAM OPT field with a specific transfer completion code (TCC) along with the other OPT fields (TCCHEN, TCINTEN, ITCCHEN, and ITCINTEN bits) to indicate whether the completion code is to be used for generating a chained event or/and for generating an interrupt upon completion of a transfer.

The specific TCC value (6-bit binary value) programmed dictates which of the 64-bits in the chain event register (CER[TCC]) and/or interrupt pending register (IPR[TCC]) is set.

You can also selectively program whether the transfer controller sends back completion codes on completion of the final transfer request (TR) of a parameter set (TCCHEN or TCINTEN), for all but the final transfer request (TR) of a parameter set (ITCCHEN or ITCINTEN), or for all TRs of a parameter set (both). See Section 8.3.8 for details on chaining (intermediate/final chaining) and Section 8.3.9 for details on intermediate/final interrupt completion.

A completion detection interface exists between the EDMA3 channel controller and transfer controller(s). This interface sends back information from the transfer controller to the channel controller to indicate that a specific transfer is completed. Completion of a transfer is used for generating chained events and/or generating interrupts to the CPU(s).

All DMA/QDMA PaRAM sets must also specify a link address value. For repetitive transfers such as ping-pong buffers, the link address value should point to another predefined PaRAM set. Alternatively, a non-repetitive transfer should set the link address value to the null link value. The null link value is defined as FFFFh. See Section 8.3.3.7 for more details.


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** Any incoming events that are mapped to a null PaRAM set results in an error condition. The error condition should be cleared before the corresponding channel is used again. See Section 8.3.3.5.

There are three ways the EDMA3CC gets updated/informed about a transfer completion: normal completion, early completion, and dummy/null completion. This applies to both chained events and completion interrupt generation.

#### 8.3.5.1 Normal Completion

In normal completion mode (TCCMODE = 0 in OPT), the transfer or sub-transfer is considered to be complete when the EDMA3 channel controller receives the completion codes from the EDMA3 transfer controller. In this mode, the completion code to the channel controller is posted by the transfer controller after it receives a signal from the destination peripheral. Normal completion is typically used to generate an interrupt to inform the CPU that a set of data is ready for processing.

#### 8.3.5.2 Early Completion

In early completion mode (TCCMODE = 1 in OPT), the transfer is considered to be complete when the EDMA3 channel controller submits the transfer request (TR) to the EDMA3 transfer controller. In this mode, the channel controller generates the completion code internally. Early completion is typically useful for chaining, as it allows subsequent transfers to be chained-triggered while the previous transfer is still in progress within the transfer controller, maximizing the overall throughput of the set of the transfers.

#### 8.3.5.3 Dummy or Null Completion

This is a variation of early completion. Dummy or null completion is associated with a dummy set (Section 8.3.3.4) or null set (Section 8.3.3.3). In both cases, the EDMA3 channel controller does not submit the associated transfer request to the EDMA3 transfer controller(s). However, if the set (dummy/null) has the OPT field programmed to return completion code (intermediate/final interrupt/chaining completion), then it will set the appropriate bits in the interrupt pending registers (IPR/IPRH) or chained event register (CER/CERH). The internal early completion path is used by the channel controller to return the completion codes internally (that is, EDMA3CC generates the completion code).

### *8.3.6 Event, Channel, and PaRAM Mapping*

Several of the 64 DMA channels are tied to a specific hardware event, thus allowing events from device peripherals or external hardware (via the EDMA_EVT[3:0] pins) to trigger transfers. A DMA channel typically requests a data transfer when it receives its event (apart from manually-triggered, chain-triggered, and other transfers). The amount of data transferred per synchronization event depends on the channel's configuration (ACNT, BCNT, CCNT, etc.) and the synchronization type (A-synchronized or AB-synchronized).

The association of an event to a channel is fixed within the EDMA Channel Controller, that is, each DMA channel has one specific event associated with it. At the SoC level, the EDMA event crossbar can be used to select which SoC level events (of which there are more than 64) are mapped to the 64 input events to the EDMA Channel Controller. The default mapping and event crossbar mapping are defined in section *EDMA Channel Synchronization Events* of the *Chip Level Resources* chapter. The event crossbar mapping is controlled by the EDMA3CC_EVTMUX_m_n registers (refer to Control Module chapter for their description).

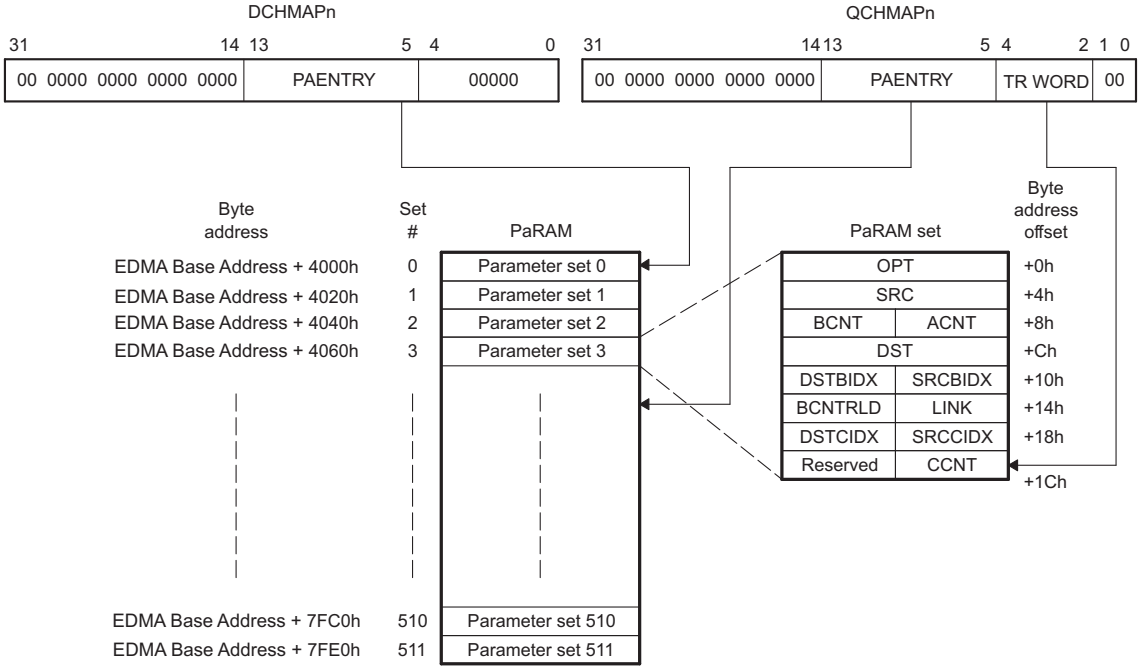
In an application, if a channel does not use the associated synchronization event or if it does not have an associated synchronization event (unused), that channel can be used for manually-triggered or chained-triggered transfers, for linking/reloading, or as a QDMA channel.


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.3.6.1 DMA Channel to PaRAM Mapping

The mapping between the DMA channel numbers and the PaRAM sets is programmable (see Table 8-1). The DMA channel mapping registers (DCHMAPn) in the EDMA3CC provide programmability that allows the DMA channels to be mapped to any of the PaRAM sets in the PaRAM memory map. Figure 8-11 illustrates the use of DCHMAP. There is one DCHMAP register per channel.

**Figure 8-11. DMA Channel and QDMA Channel to PaRAM Mapping**



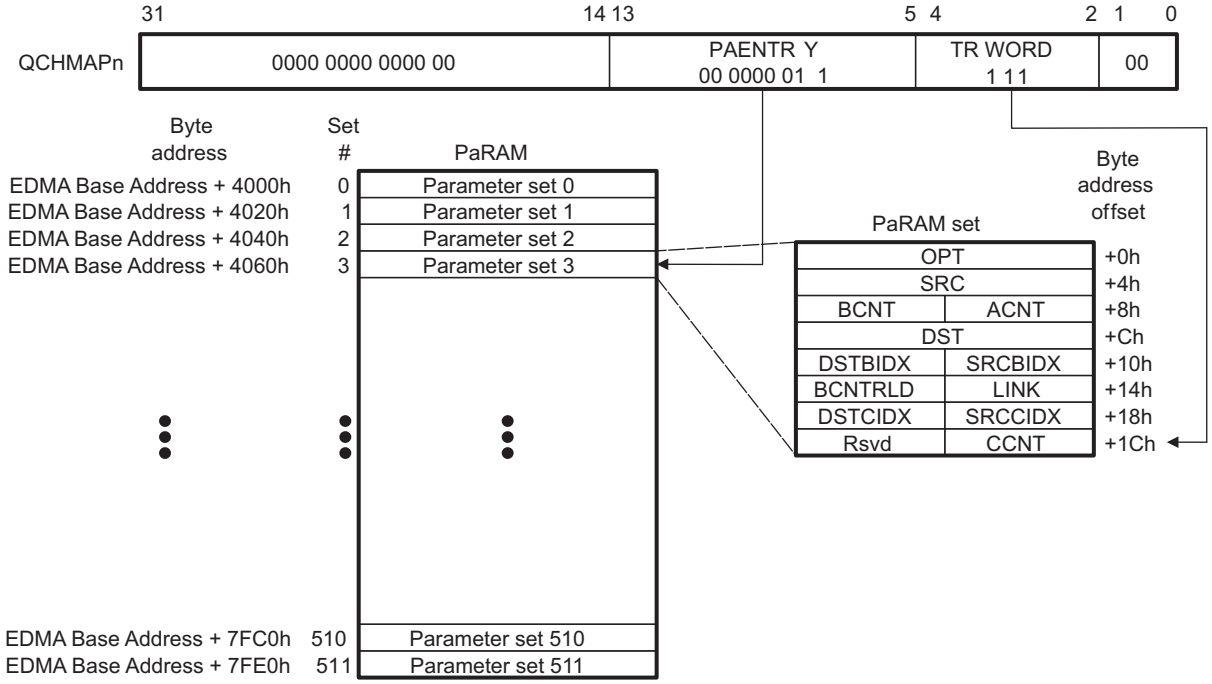

### 8.3.6.2 QDMA Channel to PaRAM Mapping

The mapping between the QDMA channels and the PaRAM sets is programmable. The QDMA channel mapping register (QCHMAP) in the EDMA3CC allows you to map the QDMA channels to any of the PaRAM sets in the PaRAM memory map. Figure 8-12 illustrates the use of QCHMAP.

Additionally, QCHMAP allows you to program the trigger word in the PaRAM set for the QDMA channel. A trigger word is one of the eight words in the PaRAM set. For a QDMA transfer to occur, a valid TR synchronization event for EDMA3CC is a write to the trigger word in the PaRAM set pointed to by QCHMAP for a particular QDMA channel. By default, QDMA channels are mapped to PaRAM set 0. You must appropriately re-map PaRAM set 0 before you use it.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-12. QDMA Channel to PaRAM Mapping**




## 8.3.7 EDMA3 Channel Controller Regions

The EDMA3 channel controller divides its address space into eight regions. Individual channel resources are assigned to a specific region, where each region is typically assigned to a specific EDMA programmer.

You can design the application software to use regions or to ignore them altogether. You can use active memory protection in conjunction with regions so that only a specific EDMA programmer (for example, privilege identification) or privilege level (for example, user vs. supervisor) is allowed access to a given region, and thus to a given DMA or QDMA channel. This allows robust system-level DMA code where each EDMA programmer only modifies the state of the assigned resources. Memory protection is described in Section 8.3.10.

### 8.3.7.1 Region Overview

The EDMA3 channel controller memory-mapped registers are divided in three main categories:

1. Global registers
2. Global region channel registers
3. Shadow region channel registers

The global registers are located at a single/fixed location in the EDMA3CC memory map. These registers control EDMA3 resource mapping and provide debug visibility and error tracking information. See the device-specific data manual for the EDMA3CC memory map.

The channel registers (including DMA, QDMA, and interrupt registers) are accessible via the global channel region address range, or in the shadow *n* channel region address range(s). For example, the event enable register (EER) is visible at the global address of EDMA Base Address + 1020h or region addresses of EDMA Base Address + 2020h for region 0, EDMA Base Address + 2220h for region 1, … EDMA Base Address + 2E20h for region 7.

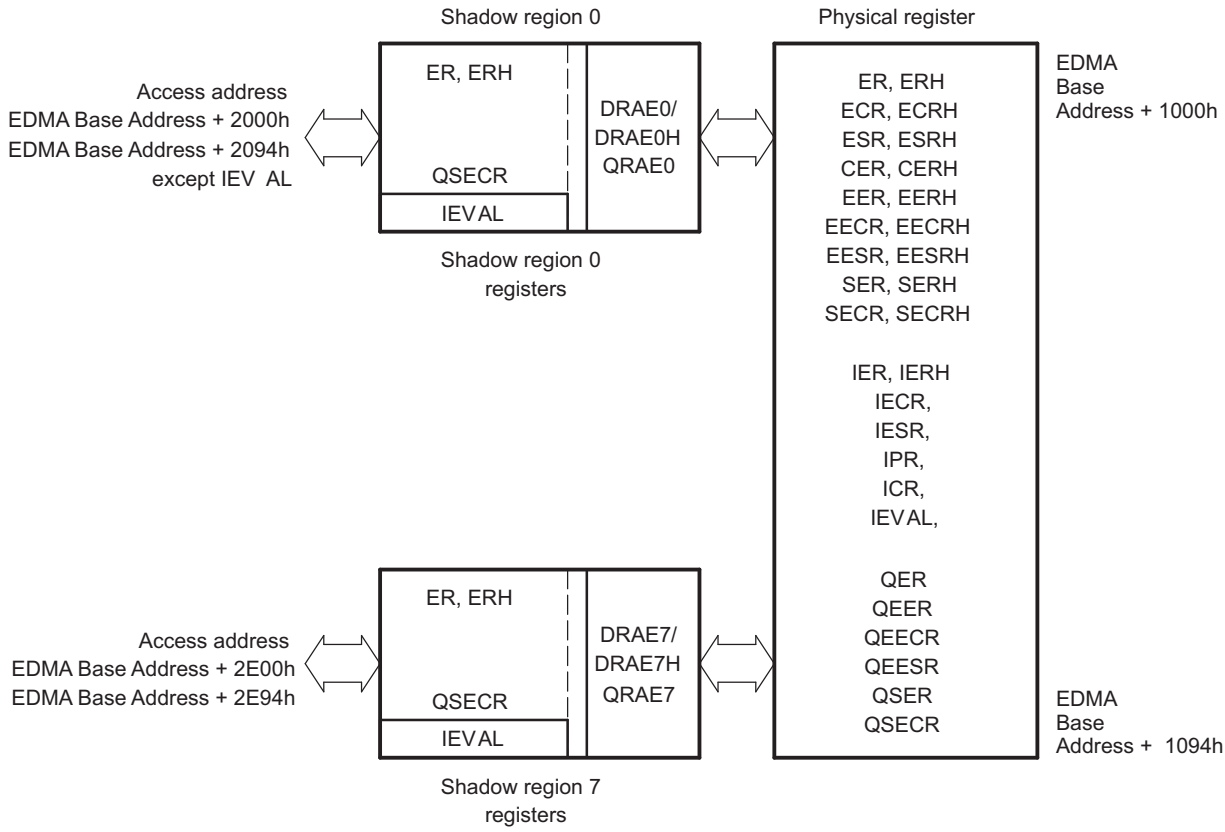
The DMA region access enable registers (DRAE*m*) and the QDMA region access enable registers (QRAE*n*) control the underlying control register bits that are accessible via the shadow region address space (except for IEVAL*n*). Table 8-7 lists the registers in the shadow region memory map. See the EDMA3CC memory map (Table 8-19) for the complete global and shadow region memory maps. Figure 8-13 illustrates the conceptual view of the regions.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-7. Shadow Region Registers**

| DRAE*m* | DRAEH*m* | QRAE*n* |
|---|---|---|
| ER | ERH | QER |
| ECR | ECRH | QEER |
| ESR | ESRH | QEECR |
| CER | CERH | QEESR |
| EER | EERH | |
| EECR | EECRH | |
| EESR | EESRH | |
| SER | SERH | |
| SECR | SECRH | |
| IER | IERH | |
| IECR | IECRH | |
| IESR | IESRH | |
| IPR | IPRH | |
| ICR | ICRH | |
| **Register not affected by DRAE\DRAEH** | | |
| IEVAL | | |

**Figure 8-13. Shadow Region Registers**




### 8.3.7.2 Channel Controller Regions

There are eight EDMA3 shadow regions (and associated memory maps). Associated with each shadow region are a set of registers defining which channels and interrupt completion codes belong to that region. These registers are user-programmed per region to assign ownership of the DMA/QDMA channels to a region.


Copyright © 2011–2016, Texas Instruments Incorporated

- DRAE*m* and DRAEH*m*: One register pair exists for each of the shadow regions. The number of bits in each register pair matches the number of DMA channels (64 DMA channels). These registers need to be programmed to assign ownership of DMA channels and interrupt (or TCC codes) to the respective region. Accesses to DMA and interrupt registers via the shadow region address view are filtered through the DRAE/DRAEH pair. A value of 1 in the corresponding DRAE(H) bit implies that the corresponding DMA/interrupt channel is accessible; a value of 0 in the corresponding DRAE(H) bit forces writes to be discarded and returns a value of 0 for reads.
- QRAE*n*: One register exists for every region. The number of bits in each register matches the number of QDMA channels (4 QDMA channels). These registers must be programmed to assign ownership of QDMA channels to the respective region. To enable a channel in a shadow region using shadow region 0 QEER, the respective bit in QRAE must be set or writing into QEESR will not have the desired effect.
- MPPA*n* and MPPAG: One register exists for every region. This register defines the privilege level, requestor, and types of accesses allowed to a region's memory-mapped registers.

It is typical for an application to have a unique assignment of QDMA/DMA channels (and, therefore, a given bit position) to a given region.

The use of shadow regions allows for restricted access to EDMA3 resources (DMA channels, QDMA channels, TCC, interrupts) by tasks in a system by setting or clearing bits in the DRAE/ORAE registers. If exclusive access to any given channel / TCC code is required for a region, then only that region's DRAE/ORAE should have the associated bit set.

***Example 8-1. Resource Pool Division Across Two Regions***

This example illustrates a judicious resource pool division across two regions, assuming region 0 must be allocated 16 DMA channels (0-15) and 1 QDMA channel (0) and 32 TCC codes (0-15 and 48-63). Region 1 needs to be allocated 16 DMA channels (16-32) and the remaining 7 QDMA channels (1-7) and TCC codes (16-47). DRAE should be equal to the OR of the bits that are required for the DMA channels and the TCC codes:

```
Region 0: DRAEH, DRAE = 0xFFFF0000, 0x0000FFFF QRAE = 0x0000001
Region 1: DRAEH, DRAE = 0x0000FFFF, 0xFFFF0000 QRAE = 0x00000FE
```

#### 8.3.7.3 Region Interrupts

In addition to the EDMA3CC global completion interrupt, there is an additional completion interrupt line that is associated with every shadow region. Along with the interrupt enable register (IER), DRAE acts as a secondary interrupt enable for the respective shadow region interrupts. See Section 8.3.9 for more information.

### *8.3.8 Chaining EDMA3 Channels*

The channel chaining capability for the EDMA3 allows the completion of an EDMA3 channel transfer to trigger another EDMA3 channel transfer. The purpose is to allow you the ability to chain several events through one event occurrence.

Chaining is different from linking (Section 8.3.3.7). The EDMA3 link feature reloads the current channel parameter set with the linked parameter set. The EDMA3 chaining feature does not modify or update any channel parameter set; it provides a synchronization event to the chained channel (see Section 8.3.4.1.3 for chain-triggered transfer requests).

Chaining is achieved at either final transfer completion or intermediate transfer completion, or both, of the current channel. Consider a channel *m* (DMA/QDMA) required to chain to channel *n*. Channel number *n* (0-63) needs to be programmed into the TCC bit of channel *m* channel options parameter (OPT) set.

- If final transfer completion chaining (TCCHEN = 1 in OPT) is enabled, the chain-triggered event occurs after the submission of the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).


Copyright © 2011–2016, Texas Instruments Incorporated

- If intermediate transfer completion chaining (ITCCHEN = 1 in OPT) is enabled, the chain-triggered event occurs after every transfer request, except the last of channel *m* is either submitted or completed (depending on early or normal completion).
- If both final and intermediate transfer completion chaining (TCCHEN = 1 and ITCCHEN = 1 in OPT) are enabled, then the chain-trigger event occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 8-8 illustrates the number of chain event triggers occurring in different synchronized scenarios. Consider channel 31 programmed with ACNT = 3, BCNT = 4, CCNT = 5, and TCC = 30.

**Table 8-8. Chain Event Triggers**

| | (Number of chained event triggers on channel 30) | |
|---|---|---|
| **Options** | **A-Synchronized** | **AB-Synchronized** |
| TCCHEN = 1, ITCCHEN = 0 | 1 (Owing to the last TR) | 1 (Owing to the last TR) |
| TCCHEN = 0, ITCCHEN = 1 | 19 (Owing to all but the last TR) | 4 (Owing to all but the last TR) |
| TCCHEN = 1, ITCCHEN = 1 | 20 (Owing to a total of 20 TRs) | 5 (Owing to a total of 5 TRs) |

### 8.3.9 EDMA3 Interrupts

The EDMA3 interrupts are divided into 2 categories: transfer completion interrupts and error interrupts.

There are nine region interrupts, eight shadow regions and one global region. The transfer completion interrupts are listed in Table 8-9. The transfer completion interrupts and the error interrupts from the transfer controllers are all routed to the DSP and ARM interrupt controllers.

**Table 8-9. EDMA3 Transfer Completion Interrupts**

| Name | Description |
|---|---|
| EDMA3CC_INT0 | EDMA3CC Transfer Completion Interrupt Shadow Region 0 |
| EDMA3CC_INT1 | EDMA3CC Transfer Completion Interrupt Shadow Region 1 |
| EDMA3CC_INT2 | EDMA3CC Transfer Completion Interrupt Shadow Region 2 |
| EDMA3CC_INT3 | EDMA3CC Transfer Completion Interrupt Shadow Region 3 |
| EDMA3CC_INT4 | EDMA3CC Transfer Completion Interrupt Shadow Region 4 |
| EDMA3CC_INT5 | EDMA3CC Transfer Completion Interrupt Shadow Region 5 |
| EDMA3CC_INT6 | EDMA3CC Transfer Completion Interrupt Shadow Region 6 |
| EDMA3CC_INT7 | EDMA3CC Transfer Completion Interrupt Shadow Region 7 |

**Table 8-10. EDMA3 Error Interrupts**

| Name | Description |
|---|---|
| EDMA3CC_ERRINT | EDMA3CC Error Interrupt |
| EDMA3CC_MPINT | EDMA3CC Memory Protection Interrupt |
| EDMA3TC0_ERRINT | TC0 Error Interrupt |
| EDMA3TC1_ERRINT | TC1 Error Interrupt |
| EDMA3TC2_ERRINT | TC2 Error Interrupt |
| EDMA3TC3_ERRINT | TC3 Error Interrupt |

#### 8.3.9.1 Transfer Completion Interrupts

The EDMA3CC is responsible for generating transfer completion interrupts to the CPU(s) (and other EDMA3 masters). The EDMA3 generates a single completion interrupt per shadow region, as well as one for the global region on behalf of all 64 channels. The various control registers and bit fields facilitate EDMA3 interrupt generation.


Copyright © 2011–2016, Texas Instruments Incorporated

The software architecture should either use the global interrupt or the shadow interrupts, but not both.

The transfer completion code (TCC) value is directly mapped to the bits of the interrupt pending register (IPR/IPRH). For example, if TCC = 10 0001b, IPRH[1] is set after transfer completion, and results in interrupt generation to the CPU(s) if the completion interrupt is enabled for the CPU. See Section 8.3.9.1.1 for details on enabling EDMA3 transfer completion interrupts.

When a completion code is returned (as a result of early or normal completions), the corresponding bit in IPR/IPRH is set if transfer completion interrupt (final/intermediate) is enabled in the channel options parameter (OPT) for a PaRAM set associated with the transfer.

**Table 8-11. Transfer Complete Code (TCC) to EDMA3CC Interrupt Mapping**

| TCC Bits in OPT (TCINTEN/ITCINTEN = 1) | IPR Bit Set | TCC Bits in OPT (TCINTEN/ITCINTEN = 1) | IPRH Bit Set(1) |
|---|---|---|---|
| 0 | IPR0 | 20h | IPR32/IPRH0 |
| 1 | IPR1 | 21h | IPR33/IPRH1 |
| 2h | IPR2 | 22h | IPR34/IPRH2 |
| 3h | IPR3 | 23h | IPR35/IPRH3 |
| 4h | IPR4 | 24h | IPR36/IPRH4 |
| ... | ... | ... | ... |
| 1Eh | IPR30 | 3Eh | IPR62/IPRH30 |
| 1Fh | IPR31 | 3Fh | IPR63/IPRH31 |

(1) Bit fields IPR[32-63] correspond to bits 0 to 31 in IPRH, respectively.

You can program the transfer completion code (TCC) to any value for a DMA/QDMA channel. A direct relation between the channel number and the transfer completion code value does not need to exist. This allows multiple channels having the same transfer completion code value to cause a CPU to execute the same interrupt service routine (ISR) for different channels.

If the channel is used in the context of a shadow region and you intend for the shadow region interrupt to be asserted, then ensure that the bit corresponding to the TCC code is enabled in IER/IERH and in the corresponding shadow region's DMA region access registers (DRAE/DRAEH).

You can enable Interrupt generation at either final transfer completion or intermediate transfer completion, or both. Consider channel *m* as an example.

- If the final transfer interrupt (TCCINT = 1 in OPT) is enabled, the interrupt occurs after the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).
- If the intermediate transfer interrupt (ITCCINT = 1 in OPT) is enabled, the interrupt occurs after every transfer request, except the last TR of channel *m* is either submitted or completed (depending on early or normal completion).
- If both final and intermediate transfer completion interrupts (TCCINT = 1, and ITCCINT = 1 in OPT) are enabled, then the interrupt occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 8-12 shows the number of interrupts that occur in different synchronized scenarios. Consider channel 31, programmed with ACNT = 3, BCNT = 4, CCNT = 5, and TCC = 30.

**Table 8-12. Number of Interrupts**

| Options | A-Synchronized | AB-Synchronized |
|---|---|---|
| TCINTEN = 1, ITCINTEN = 0 | 1 (Last TR) | 1 (Last TR) |
| TCINTEN = 0, ITCINTEN = 1 | 19 (All but the last TR) | 4 (All but the last TR) |
| TCINTEN = 1, ITCINTEN = 1 | 20 (All TRs) | 5 (All TRs) |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.3.9.1.1 Enabling Transfer Completion Interrupts

For the EDMA3 channel controller to assert a transfer completion to the external environment, the interrupts must be enabled in the EDMA3CC. This is in addition to setting up the TCINTEN and ITCINTEN bits in OPT of the associated PaRAM set.

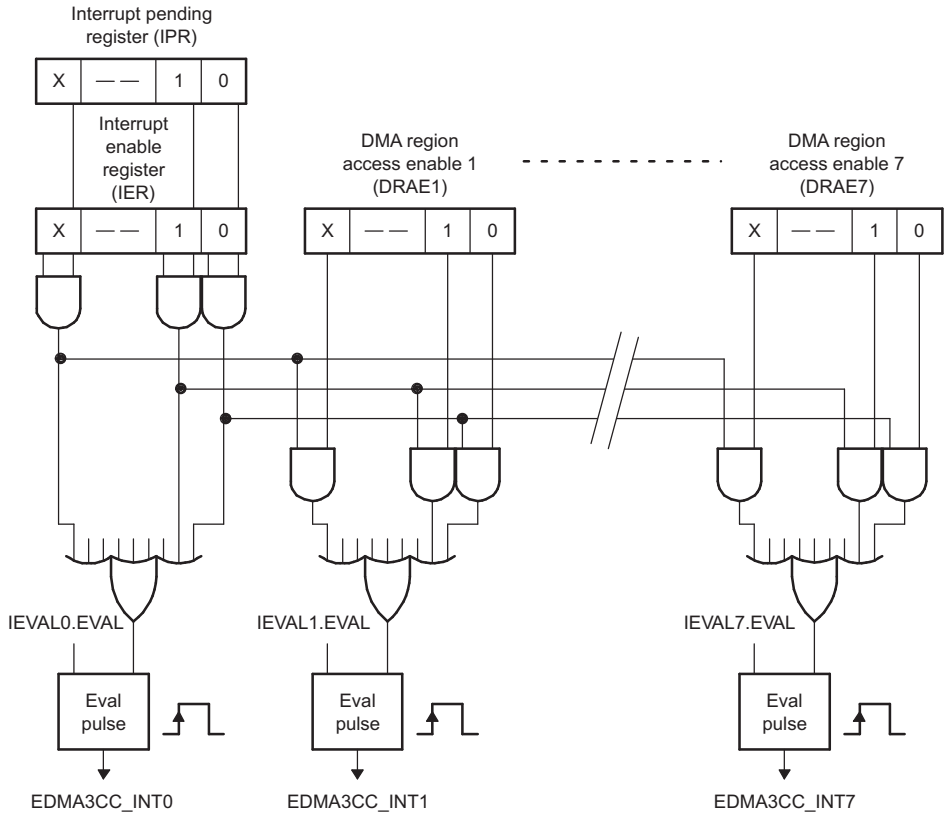
The EDMA3 channel controller has interrupt enable registers (IER/IERH) and each bit location in IER/IERH serves as a primary enable for the corresponding interrupt pending registers (IPR/IPRH).

All of the interrupt registers (IER, IESR, IECR, and IPR) are either manipulated from the global DMA channel region, or by the DMA channel shadow regions. The shadow regions provide a view to the same set of physical registers that are in the global region.

The EDMA3 channel controller has a hierarchical completion interrupt scheme that uses a single set of interrupt pending registers (IPR/IPRH) and single set of interrupt enable registers (IER/IERH). The programmable DMA region access enable registers (DRAE/DRAEH) provides a second level of interrupt masking. The global region interrupt output is gated based on the enable mask that is provided by IER/IERH. see Figure 8-14

The region interrupt outputs are gated by IER and the specific DRAE/DRAEH associated with the region. See Figure 8-14.

**Figure 8-14. Interrupt Diagram**




For the EDMA3CC to generate the transfer completion interrupts that are associated with each shadow region, the following conditions must be true:

- EDMA3CC_INT0: (IPR.E0 & IER.E0 & DRAE0.E0) | (IPR.E1 & IER.E1 & DRAE0.E1) | …|(IPRH.E63 & IERH.E63 & DRAHE0.E63)
- EDMA3CC_INT1: (IPR.E0 & IER.E0 & DRAE1.E0) | (IPR.E1 & IER.E1 & DRAE1.E1) | …| (IPRH.E63 & IERH.E63 & DRAHE1.E63)
- EDMA3CC_INT2 : (IPR.E0 & IER.E0 & DRAE2.E0) | (IPR.E1 & IER.E1 & DRAE2.E1) | …|(IPRH.E63 & IERH.E63 & DRAHE2.E63)....


Copyright © 2011–2016, Texas Instruments Incorporated

- Up to EDMA3CC_INT7 : (IPR.E0 & IER.E0 & DRAE7.E0) | (IPR.E1 & IER.E1 & DRAE7.E1) | …|(IPRH.E63 & IERH.E63 & DRAEH7.E63)

**NOTE:** The DRAE/DRAEH for all regions are expected to be set up at system initialization and to remain static for an extended period of time. The interrupt enable registers should be used for dynamic enable/disable of individual interrupts.

Because there is no relation between the TCC value and the DMA/QDMA channel, it is possible, for example, for DMA channel 0 to have the OPT.TCC = 63 in its associated PaRAM set. This would mean that if a transfer completion interrupt is enabled (OPT.TCINTEN or OPT.ITCINTEN is set), then based on the TCC value, IPRH.E63 is set up on completion. For proper channel operations and interrupt generation using the shadow region map, you must program the DRAE/DRAEH that is associated with the shadow region to have read/write access to both bit 0 (corresponding to channel 0) and bit 63 (corresponding to IPRH bit that is set upon completion).

#### *8.3.9.1.2 Clearing Transfer Completion Interrupts*

Transfer completion interrupts that are latched to the interrupt pending registers (IPR/IPRH) are cleared by writing a 1 to the corresponding bit in the interrupt pending clear register (ICR/ICRH). For example, a write of 1 to ICR.E0 clears a pending interrupt in IPR.E0.

If an incoming transfer completion code (TCC) gets latched to a bit in IPR/IPRH, then additional bits that get set due to a subsequent transfer completion will not result in asserting the EDMA3CC completion interrupt. In order for the completion interrupt to be pulsed, the required transition is from a state where no enabled interrupts are set to a state where at least one enabled interrupt is set.

### 8.3.9.2 EDMA3 Interrupt Servicing

Upon completion of a transfer (early or normal completion), the EDMA3 channel controller sets the appropriate bit in the interrupt pending registers (IPR/IPRH), as the transfer completion codes specify. If the completion interrupts are appropriately enabled, then the CPU enters the interrupt service routine (ISR) when the completion interrupt is asserted.

After servicing the interrupt, the ISR should clear the corresponding bit in IPR/IPRH, thereby enabling recognition of future interrupts. The EDMA3CC will only assert additional completion interrupts when all IPR/IPRH bits clear.

When one interrupt is serviced many other transfer completions may result in additional bits being set in IPR/IPRH, thereby resulting in additional interrupts. Each of the bits in IPR/IPRH may need different types of service; therefore, the ISR may check all pending interrupts and continue until all of the posted interrupts are serviced appropriately.

Examples of pseudo code for a CPU interrupt service routine for an EDMA3CC completion interrupt are shown in Example 8-2 and Example 8-3.

The ISR routine in Example 8-2 is more exhaustive and incurs a higher latency.

***Example 8-2. Interrupt Servicing***

The pseudo code:

1. Reads the interrupt pending register (IPR/IPRH).
2. Performs the operations needed.
3. Writes to the interrupt pending clear register (ICR/ICRH) to clear the corresponding IPR/IPRH bit(s).
4. Reads IPR/IPRH again:
   (a) If IPR/IPRH is not equal to 0, repeat from step 2 (implies occurrence of new event between step 2 to step 4).
   (b) If IPR/IPRH is equal to 0, this should assure you that all of the enabled interrupts are inactive.


Copyright © 2011–2016, Texas Instruments Incorporated

***Example 8-2. Interrupt Servicing (continued)***

**NOTE:** An event may occur during step 4 while the IPR/IPRH bits are read as 0 and the application is still in the interrupt service routine. If this happens, a new interrupt is recorded in the device interrupt controller and a new interrupt generates as soon as the application exits in the interrupt service routine.

Example 8-3 is less rigorous, with less burden on the software in polling for set interrupt bits, but can occasionally cause a race condition as mentioned above.

***Example 8-3. Interrupt Servicing***

If you want to leave any enabled and pending (possibly lower priority) interrupts; you must force the interrupt logic to reassert the interrupt pulse by setting the EVAL bit in the interrupt evaluation register (IEVAL).

The pseudo code is as follows:

1. Enters ISR.
2. Reads IPR/IPRH.
3. For the condition that is set in IPR/IPRH that you want to service, do the following:
   (a) Service interrupt as the application requires.
   (b) Clear the bit for serviced conditions (others may still be set, and other transfers may have resulted in returning the TCC to EDMA3CC after step 2).
4. Reads IPR/IPRH prior to exiting the ISR:
   (a) If IPR/IPRH is equal to 0, then exit the ISR.
   (b) If IPR/IPRH is not equal to 0, then set IEVAL so that upon exit of ISR, a new interrupt triggers if any enabled interrupts are still pending.

#### 8.3.9.3 Interrupt Evaluation Operations

The EDMA3CC has interrupt evaluate registers (IEVAL) that exist in the global region and in each shadow region. The registers in the shadow region are the only registers in the DMA channel shadow region memory map that are not affected by the settings for the DMA region access enable registers (DRAE/DRAEH). Writing a 1 to the EVAL bit in the registers that are associated with a particular shadow region results in pulsing the associated region interrupt (global or shadow), if any enabled interrupt (via IER/IERH) is still pending (IPR/IPRH). This register assures that the CPU does not miss the interrupts (or the EDMA3 master associated with the shadow region) if the software architecture chooses not to use all interrupts. See Example 8-3 for the use of IEVAL in the EDMA3 interrupt service routine (ISR).

Similarly an error evaluation register (EEVAL) exists in the global region. Writing a 1 to the EVAL bit in EEVAL causes the pulsing of the error interrupt if any pending errors are in EMR/EMRH, QEMR, or CCERR. See Section 8.3.9.4 for additional information regarding error interrupts.

**NOTE:** While using IEVAL for shadow region completion interrupts, you should make sure that the IEVAL operated upon is from that particular shadow region memory map.


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.3.9.4 Error Interrupts

The EDMA3CC error registers provide the capability to differentiate error conditions (event missed, threshold exceed, etc.). Additionally, setting the error bits in these registers results in asserting the EDMA3CC error interrupt. If the EDMA3CC error interrupt is enabled in the device interrupt controller(s), then it allows the CPU(s) to handle the error conditions.

The EDMA3CC has a single error interrupt (EDMA3CC_ERRINT) that is asserted for all EDMA3CC error conditions. There are four conditions that cause the error interrupt to pulse:

- DMA missed events: for all 64 DMA channels. DMA missed events are latched in the event missed registers (EMR/EMRH).
- QDMA missed events: for all 8 QDMA channels. QDMA missed events are latched in the QDMA event missed register (QEMR).
- Threshold exceed: for all event queues. These are latched in EDMA3CC error register (CCERR).
- TCC error: for outstanding transfer requests that are expected to return completion code (TCCHEN or TCINTEN bit in OPT is set to 1) exceeding the maximum limit of 63. This is also latched in the EDMA3CC error register (CCERR).

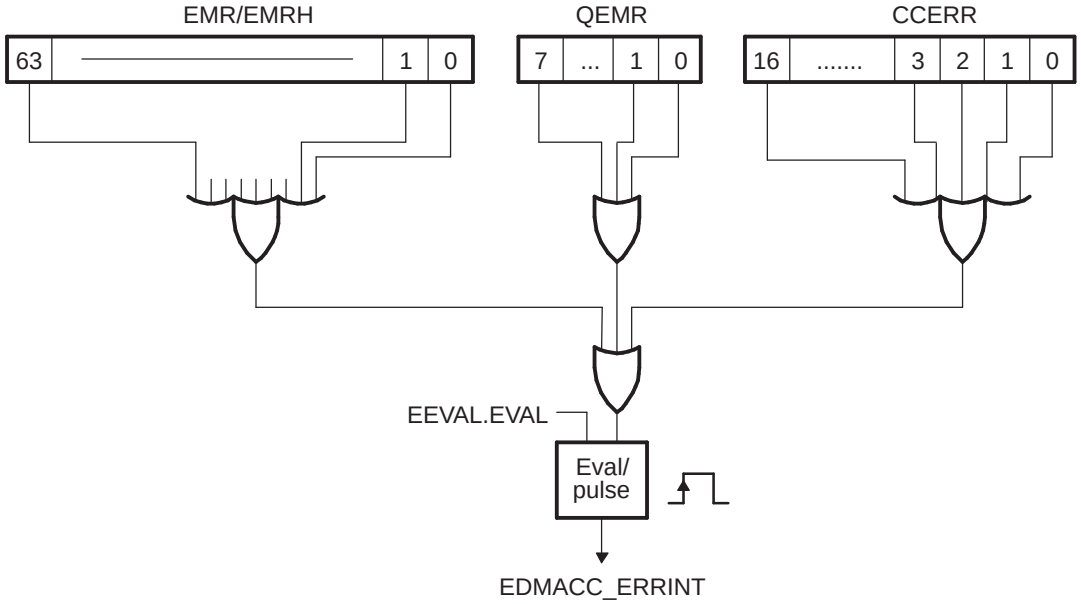
Figure 8-15 illustrates the EDMA3CC error interrupt generation operation.

If any of the bits are set in the error registers due to any error condition, the EDMA3CC_ERRINT is always asserted, as there are no enables for masking these error events. Similar to transfer completion interrupts (EDMA3CC_INT), the error interrupt also only pulses when the error interrupt condition transitions from no errors being set to at least one error being set. If additional error events are latched prior to the original error bits clearing, the EDMA3CC does not generate additional interrupt pulses.

To reduce the burden on the software, there is an error evaluate register (EEVAL) that allows re-evaluation of pending set error events/bits, similar to the interrupt evaluate register (IEVAL). You can use this so that the CPU(s) does not miss any error events.

**NOTE:** It is good practice to enable the error interrupt in the device interrupt controller and to associate an interrupt service routine with it to address the various error conditions appropriately. Doing so puts less burden on the software (polling for error status); additionally, it provides a good debug mechanism for unexpected error conditions.

**Figure 8-15. Error Interrupt Operation**





Copyright © 2011–2016, Texas Instruments Incorporated

### 8.3.10 Memory Protection

The EDMA3 channel controller supports two kinds of memory protection: active and proxy.

#### 8.3.10.1 Active Memory Protection

Active memory protection is a feature that allows or prevents read and write accesses (by any EDMA3 programmer) to the EDMA3CC registers (based on permission characteristics that you program). Active memory protection is achieved by a set of memory protection permissions attribute (MPPA) registers.

The EDMA3CC register map is divided into three categories:

- a global region.
- a global channel region.
- eight shadow regions.

Each shadow region consists of the respective shadow region registers and the associated PaRAM. For more detailed information regarding the contents of a shadow region, refer to section Table 8-7.

Each of the eight shadow regions has an associated MPPA register (MPPA*n*) that defines the specific requestor(s) and types of requests that are allowed to the regions resources.

The global channel region is also protected with a memory-mapped register (MPPAG). The MPPAG applies to the global region and to the global channel region, except the other MPPA registers themselves. For more detailed information on the list of the registers in each region, refer to the register memory-map in section Table 8-14.

See Section 8.5.1.5.4 for the bit field descriptions of MPPA*n*. The MPPA*n* have a certain set of access rules.

Table 8-13 shows the accesses that are allowed or not allowed to the MPPAG and MPPA*n*. The active memory protection uses the PRIV and PRIVID attributes of the EDMA programmer. The PRIV is the privilege level (i.e., user vs. supervisor). The PRIVID refers to a privilege ID with a number that is associated with an EDMA3 programmer. See the device-specific data manual for the PRIVIDs that are associated with potential EDMA3 programmers.

**Table 8-13. Allowed Accesses**

| Access | Supervisor | User |
|---|---|---|
| Read | Yes | Yes |
| Write | Yes | No |

Table 8-14 describes the MPPA register mapping for the shadow regions (which includes shadow region registers and PaRAM addresses).

The region-based MPPA registers are used to protect accesses to the DMA shadow regions and the associated region PaRAM. Because there are eight regions, there are eight MPPA region registers (MPPA[0-7]).

**Table 8-14. MPPA Registers to Region Assignment**

| Register | Registers Protect | Address Range | PaRAM Protect[1] | Address Range |
|---|---|---|---|---|
| MPPAG | Global Range | 0000h-1FFCh | N/A | N/A |
| MPPA0 | DMA Shadow 0 | 2000h-21FCh | 1st octant | 4000h-47FCh |
| MPPA1 | DMA Shadow 1 | 2200h-23FCh | 2nd octant | 4800h-4FFCh |
| MPPA2 | DMA Shadow 2 | 2400h-25FCh | 3rd octant | 5000h-57FCh |
| MPPA3 | DMA Shadow 3 | 2600h-27FCh | 4th octant | 5800h-5FFCh |
| MPPA4 | DMA Shadow 4 | 2800h-29FCh | 5th octant | 6000h-67FCh |
| MPPA5 | DMA Shadow 5 | 2A00h-2BFCh | 6th octant | 6800h-6FFCh |
| MPPA6 | DMA Shadow 6 | 2C00h-2DFCh | 7th octant | 7000h-77FCh |
| MPPA7 | DMA Shadow 7 | 2E00h-2FFCh | 8th octant | 7800h-7FFCh |

(1) The PARAM region is divided into 8 regions referred to as an octant.



Copyright © 2011–2016, Texas Instruments Incorporated

Example Access denied.

Write access to shadow region 7's event enable set register (EESR):

1. The original value of the event enable register (EER) at address offset 0x1020 is 0x0.
2. The MPPA[7] is set to prevent user level accesses (UW = 0, UR = 0), but it allows supervisor level accesses (SW = 1, SR = 1) with a privilege ID of 0. (AID0 = 1).
3. An EDMA3 programmer with a privilege ID of 0 attempts to perform a user-level write of a value of 0xFF00FF00 to shadow region 7's event enable set register (EESR) at address offset 0x2E30. Note that the EER is a read-only register and the only way that you can write to it is by writing to the EESR. Also remember that there is only one physical register for EER, EESR, etc. and that the shadow regions only provide to the same physical set.
4. Since the MPPA[7] has UW = 0, though the privilege ID of the write access is set to 0, the access is not allowed and the EER is not written to.

**Table 8-15. Example Access Denied**

| Register | Value | Description |
|---|---|---|
| EER<br>(offset 0x1020) | 0x0000 0000 | Value in EER to begin with. |
| EESR<br>(offset 0x2E30) | 0xFF00 FF00<br>↓ | Value attempted to be written to shadow region 7's EESR.<br>This is done by an EDMA3 programmer with a privilege level of User and Privilege ID of 0. |
| MPPA[7]<br>(offset 0x082C) | 0x0000 04B0 | Memory Protection Filter AID0 = 1, UW = 0, UR = 0, SW = 1, SR = 1. |
| | X | Access Denied |
| EER<br>(offset 0x1020) | 0x0000 0000 | Final value of EER |

Example Access Allowed

Write access to shadow region 7's event enable set register (EESR):

1. The original value of the event enable register (EER) at address offset 0x1020 is 0x0.
2. The MPPA[7] is set to allow user-level accesses (UW = 1, UR = 1) and supervisor-level accesses (SW = 1, SR = 1) with a privilege ID of 0. (AID0 = 1).
3. An EDMA3 programmer with a privilege ID of 0, attempts to perform a user-level write of a value of 0xABCD0123 to shadow region 7's event enable set register (EESR) at address offset 0x2E30. Note that the EER is a read-only register and the only way that you can write to it is by writing to the EESR. Also remember that there is only one physical register for EER, EESR, etc. and that the shadow regions only provide to the same physical set.
4. Since the MPPA[7] has UW = 1 and AID0 = 1, the user-level write access is allowed.
5. Remember that accesses to shadow region registers are masked by their respective DRAE register. In this example, the DRAE[7] is set of 0x9FF00FC2.
6. The value finally written to EER is 0x8BC00102.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-16. Example Access Allowed**

| Register | Value | Description |
|---|---|---|
| EER (offset 0x1020) | 0x0000 0000 | Value in EER to begin with. |
| EESR (offset 0x2E30) | 0xFF00 FF00 | Value attempted to be written to shadow region 7's EESR. This is done by an EDMA3 programmer with a privilege level of User and Privilege ID of 0. |
| MPPA[7] (offset 0x082C) | 0x0000 04B3 | Memory Protection Filter AID = 1, UW = 1, UR = 1, SW = 1, SR = 1. |
| | √ ↓ | Access allowed. |
| DRAE[7] (offset 0x0378) | 0x9FF0 0FC2 ↓ | DMA Region Access Enable Filter |
| EESR (offset 0x2E30) | 0x8BC0 0102 ↓ | Value written to shadow region 7's EESR. This is done by an EDMA3 programmer with a privilege level of User and a Privilege ID of 0. |
| EER (offset 0x1020) | ↓ 0xBC0 0102 | Final value of EER. |

### 8.3.10.2 Proxy Memory Protection

Proxy memory protection allows an EDMA3 transfer programmed by a given EDMA3 programmer to have its permissions travel with the transfer through the EDMA3TC. The permissions travel along with the read transactions to the source and the write transactions to the destination endpoints. The PRIV bit and PRIVID bit in the channel options parameter (OPT) is set with the EDMA3 programmer's PRIV value and PRIVID values, respectively, when any part of the PaRAM set is written.

The PRIV is the privilege level (i.e., user vs. supervisor). The PRIVID refers to a privilege ID with a number that is associated with an EDMA3 programmer.

See the data manual for the PRIVIDs that are associated with potential EDMA3 programmers.

These options are part of the TR that are submitted to the transfer controller. The transfer controller uses the above values on their respective read and write command bus so that the target endpoints can perform memory protection checks based on these values.

Consider a parameter set that is programmed by a CPU in user privilege level for a simple transfer with the source buffer on an L2 page and the destination buffer on an L1D page. The PRIV is 0 for user-level and the CPU has a PRIVID of 0.

The PaRAM set is shown in Figure 8-16.


Copyright © 2011–2016, Texas Instruments Incorporated

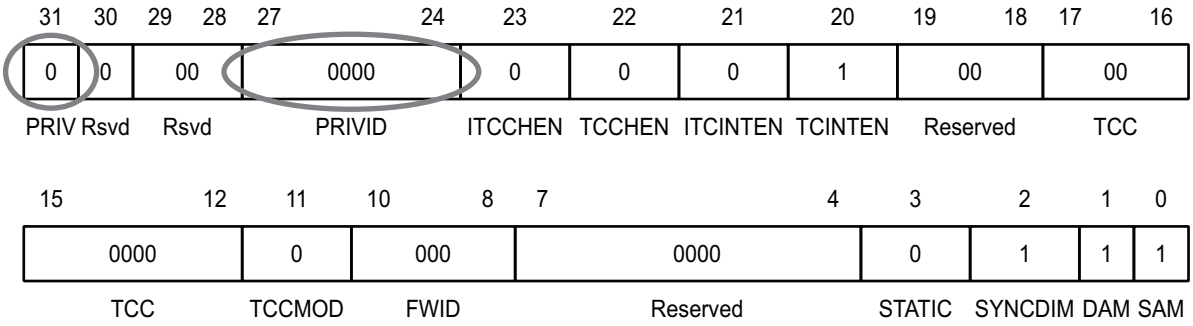
**Figure 8-16. PaRAM Set Content for Proxy Memory Protection Example**

*(a) EDMA3 Parameters*

| Parameter Contents | | Parameter | |
|---|---|---|---|
| 0010 0007h | | Channel Options Parameter (OPT) | |
| 009F 0000h | | Channel Source Address (SRC) | |
| 0001h | 0004h | Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| 00F0 7800h | | Channel Destination Address (DST) | |
| 0001h | 0001h | Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| 0000h | FFFFh | BCNT Reload (BCNTRLD) | Link Address (LINK) |
| 0001h | 1000h | Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| 0000h | 0001h | Reserved | Count for 3rd Dimension (CCNT) |

**Figure 8-17. Channel Options Parameter (OPT) Example**

(b) Channel Options Parameter (OPT) Content




The PRIV and PRIVID information travels along with the read and write requests that are issued to the source and destination memories.

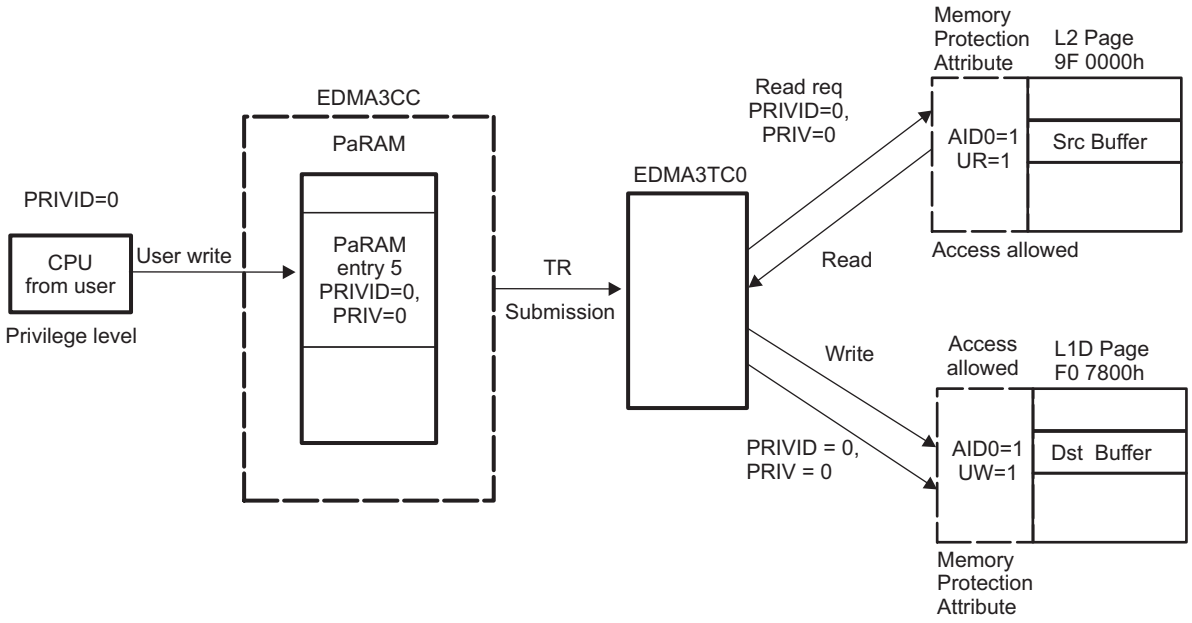
For example, if the access attributes that are associated with the L2 page with the source buffer only allow supervisor read, write accesses (SR,SW), the user-level read request above is refused. Similarly, if the access attributes that are associated with the L1D page with the destination buffer only allow supervisor read and write accesses (SR, SW), the user-level write request above is refused. For the transfer to succeed, the source and destination pages should have user-read and user-write permissions, respectively, along with allowing accesses from a PRIVID 0. For more information regarding how to set memory protection attributes for pages of memory in L2/L1D, please refer to the *TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide* (SPRU732).

Because the programmers privilege level and privilege identification travel with the read and write requests, EDMA3 acts as a proxy.

Figure 8-18 illustrates the propagation of PRIV and PRIVID at the boundaries of all the interacting entities (CPU, EDMA3CC, EDMA3TC, and slave memories).


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-18. Proxy Memory Protection Example**




## 8.3.11 Event Queue(s)

Event queues are a part of the EDMA3 channel controller. Event queues form the interface between the event detection logic in the EDMA3CC and the transfer request (TR) submission logic of the EDMA3CC. Each queue is 16 entries deep; thus, each event queue can queue a maximum of 16 events. If there are more than 16 events, then the events that cannot find a place in the event queue remain set in the associated event register and the CPU does not stall.

There are four event queues for the device: Queue0, Queue1, Queue2, and Queue3. Events in Queue0 result in submission of its associated transfer requests (TRs) to TC0. Similarly, transfer requests that are associated with events in Queue3 are submitted to TC3.

An event that wins prioritization against other DMA and/or QDMA pending events is placed at the tail of the appropriate event queue. Each event queue is serviced in FIFO order. Once the event reaches the head of its queue and the corresponding transfer controller is ready to receive another TR, the event is de-queued and the PaRAM set corresponding to the de-queued event is processed and submitted as a transfer request packet (TRP) to the associated EDMA3 transfer controller.

Queue0 has highest priority and Queue3 has the lowest priority, if Queue0 and Queue1 both have at least one event entry and if both TC0 and TC1 can accept transfer requests, then the event in Queue0 is de-queued first and its associated PaRAM set is processed and submitted as a transfer request (TR) to TC0.

See Section 8.3.11.4 for system-level performance considerations. All of the event entries in all of the event queues are software readable (not writeable) by accessing the event entry registers (Q0E0, Q0E1,…Q1E15, etc.). Each event entry register characterizes the queued event in terms of the type of event (manual, event, chained or auto-triggered) and the event number. See Section 8.5.1.4.1 for a description of the bit fields in the queue event entry registers.

### 8.3.11.1 DMA/QDMA Channel to Event Queue Mapping

Each of the 64 DMA channels and eight QDMA channels are programmed independently to map to a specific queue, using the DMA queue number register (DMAQNUM) and the QDMA queue number register (QDMANUM). The mapping of DMA/QDMA channels is critical to achieving the desired performance level for the EDMA and most importantly, in meeting real-time deadlines. See Section 8.3.11.4.


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** If an event is ready to be queued and both the event queue and the EDMA3 transfer controller that is associated to the event queue are empty, then the event bypasses the event queue, and moves the PaRAM processing logic, and eventually to the transfer request submission logic for submission to the EDMA3TC. In this case, the event is not logged in the event queue status registers.

#### 8.3.11.2 Queue RAM Debug Visibility

There are four event queues and each queue has 16 entries. These 16 entries are managed in a circular FIFO manner. There is a queue status register (QSTAT) associated with each queue. These along with all of the 16 entries per queue can be read via registers QSTAT*n* and QxEy, respectively.

These registers provide user visibility and may be helpful while debugging real-time issues (typically post-mortem), involving multiple events and event sources. The event queue entry register (Q*x*E*y*) uniquely identifies the specific event type (event-triggered, manually-triggered, chain-triggered, and QDMA events) along with the event number (for all DMA/QDMA event channels) that are in the queue or have been de-queued (passed through the queue).

Each of the 16 entries in the event queue are read using the EDMA3CC memory-mapped register. By reading the event queue, you see the history of the last 16 TRs that have been processed by the EDMA3 on a given queue. This provides user/software visibility and is helpful for debugging real-time issues (typically post-mortem), involving multiple events and event sources.

The queue status register (QSTAT*n*) includes fields for the start pointer (STRTPTR) which provides the offset to the head entry of an event. It also includes a field called NUMVAL that provides the total number of valid entries residing in the event queue at a given instance of time. The STRTPTR may be used to index appropriately into the 16 event entries. NUMVAL number of entries starting from STRTPTR are indicative of events still queued in the respective queue. The remaining entry may be read to determine what's already de-queued and submitted to the associated transfer controller.

#### 8.3.11.3 Queue Resource Tracking

The EDMA3CC event queue includes watermarking/threshold logic that allows you to keep track of maximum usage of all event queues. This is useful for debugging real-time deadline violations that may result from head-of-line blocking on a given EDMA3 event queue.

You can program the maximum number of events that can queue up in an event queue by programming the threshold value (between 0 to 15) in the queue watermark threshold A register (QWMTHRA). The maximum queue usage is recorded actively in the watermark (WM) field of the queue status register (QSTAT*n*) that keeps getting updated based on a comparison of number of valid entries, which is also visible in the NUMVAL bit in QSTAT*n* and the maximum number of entries (WM bit in QSTAT*n*).

If the queue usage is exceeded, this status is visible in the EDMA3CC registers: the QTHRXCD*n* bit in the channel controller error register (CCERR) and the THRXCD bit in QSTAT*n*, where *n* stands for the event queue number. Any bits that are set in CCERR also generate an EDMA3CC error interrupt.

#### 8.3.11.4 Performance Considerations

The main system bus infrastructure (L3) (see the device-specific data manual) arbitrates bus requests from all of the masters (TCs, CPU(S), PCIe and other bus masters) to the shared slave resources (peripherals and memories).

The priorities of transfer requests (read and write commands) from the EDMA3 transfer controllers with respect to other masters within the system crossbar are programmed using the Control Module INIT_PRIORITY_n and INITIATOR_PRIO_m registers. The QUEPRI register has no affect.

Therefore, the priority of unloading queues has a secondary affect compared to the priority of the transfers as they are executed by the EDMA3TC (dictated by the priority set using the Control Module INIT_PRIORITY_n and INITIATOR_PRIO_m registers).


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.3.12 EDMA3 Transfer Controller (EDMA3TC)

The EDMA3 channel controller is the user-interface of the EDMA3 and the EDMA3 transfer controller (EDMA3TC) is the data movement engine of the EDMA3. The EDMA3CC submits transfer requests (TR) to the EDMA3TC and the EDMA3TC performs the data transfers dictated by the TR; thus, the EDMA3TC is a slave to the EDMA3CC.

#### 8.3.12.1 Architecture Details

##### 8.3.12.1.1 Command Fragmentation

The TC read and write controllers in conjunction with the source and destination register sets are responsible for issuing optimally-sized reads and writes to the slave endpoints. An optimally-sized command is defined by the transfer controller default burst size (DBS), which is defined in Section 8.3.12.5.

The EDMA3TC attempts to issue the largest possible command size as limited by the DBS value or the ACNT/BCNT value of the TR. EDMA3TC obeys the following rules:

- The read/write controllers always issue commands less than or equal to the DBS value.
- The first command of a 1D transfer command always aligns the address of subsequent commands to the DBS value.

Table 8-17 lists the TR segmentation rules that are followed by the EDMA3TC. In summary, if the ACNT value is larger than the DBS value, then the EDMA3TC breaks the ACNT array into DBS-sized commands to the source/destination addresses. Each BCNT number of arrays are then serviced in succession.

For BCNT arrays of ACNT bytes (that is, a 2D transfer), if the ACNT value is less than or equal to the DBS value, then the TR may be optimized into a 1D-transfer in order to maximize efficiency. The optimization takes place if the EDMA3TC recognizes that the 2D-transfer is organized as a single dimension (ACNT == BIDX) and the ACNT value is a power of 2.

Table 8-17 lists conditions in which the optimizations are performed.

**Table 8-17. Read/Write Command Optimization Rules**

| ACNT ≤ DBS | ACNT is power of 2 | BIDX = ACNT | BCNT ≤ 1023 | SAM/DAM = Increment | Description |
|---|---|---|---|---|---|
| Yes | Yes | Yes | Yes | Yes | Optimized |
| No | x | x | x | x | Not Optimized |
| x | No | x | x | x | Not Optimized |
| x | x | No | x | x | Not Optimized |
| x | x | x | No | x | Not Optimized |
| x | x | x | x | No | Not Optimized |

##### 8.3.12.1.2 TR Pipelining

TR pipelining refers to the ability of the source active set to proceed ahead of the destination active set. Essentially, the reads for a given TR may already be in progress while the writes of a previous TR may not have completed.

The number of outstanding TRs is limited by the number of destination FIFO register entries, which is 4.

TR pipelining is useful for maintaining throughput on back-to-back small TRs. It minimizes the startup overhead because reads start in the background of a previous TR writes.


Copyright © 2011–2016, Texas Instruments Incorporated

***Example 8-4. Command Fragmentation (DBS = 64)***

The pseudo code:

1. ACNT = 8, BCNT = 8, SRCBIDX = 8, DSTBIDX = 10, SRCADDR = 64, DSTADDR = 191

   Read Controller: This is optimized from a 2D-transfer to a 1D-transfer such that the read side is equivalent to ACNT = 64, BCNT = 1.

   Cmd0 = 64 byte

   Write Controller: Because DSTBIDX != ACNT, it is not optimized.

   Cmd0 = 8 byte, Cmd1 = 8 byte, Cmd2 = 8 byte, Cmd3 = 8 byte, Cmd4 = 8 byte, Cmd5 = 8 byte, Cmd6 = 8 byte, Cmd7 = 8 byte.

2. ACNT=128, BCNT = 1,SRCADDR = 63, DSTADDR = 513

   Read Controller: Read address is not aligned.

   Cmd0 = 1 byte, (now the SRCADDR is aligned to 64 for the next command)

   Cmd1 = 64 bytes

   Cmd2 = 63 bytes

   Write Controller: The write address is also not aligned.

   Cmd0 = 63 bytes, (now the DSTADDR is aligned to 64 for the next command)

   Cmd1 = 64 bytes

   Cmd2 = 1 byte

#### *8.3.12.1.3 Performance Tuning*

By default, reads are as issued as fast as possible. In some cases, the reads issued by the EDMA3TC could fill the available command buffering for a slave, delaying other (potentially higher priority) masters from successfully submitting commands to that slave. The rate at which read commands are issued by the EDMA3TC is controlled by the RDRATE register. The RDRATE register defines the number of cycles that the EDMA3TC read controller waits before issuing subsequent commands for a given TR, thus minimizing the chance of the EDMA3TC consuming all available slave resources. The RDRATE value should be set to a relatively small value if the transfer controller is targeted for high priority transfers and to a higher value if the transfer controller is targeted for low priority transfers.

In contrast, the Write Interface does not have any performance turning knobs because writes always have an interval between commands as write commands are submitted along with the associated write data.

### 8.3.12.2 Memory Protection

The transfer controller plays an important role in handling proxy memory protection. There are two access properties associated with a transfer: for instance, the privilege id (system-wide identification assigned to a master) of the master initiating the transfer, and the privilege level (user versus supervisor) used to program the transfer. This information is maintained in the PaRAM set when it is programmed in the channel controller. When a TR is submitted to the transfer controller, this information is made available to the EDMA3TC and used by the EDMA3TC while issuing read and write commands. The read or write commands have the same privilege identification, and privilege level as that programmed in the EDMA3 transfer in the channel controller.

### 8.3.12.3 Error Generation

Errors are generated if enabled under three conditions:

- EDMA3TC detection of an error signaled by the source or destination address.
- Attempt to read or write to an invalid address in the configuration memory map.
- Detection of a constant addressing mode TR violating the constant addressing mode transfer rules (the source/destination addresses and source/destination indexes must be aligned to 32 bytes).


Copyright © 2011–2016, Texas Instruments Incorporated

Either or all error types may be disabled. If an error bit is set and enabled, the error interrupt for the concerned transfer controller is pulsed.

#### 8.3.12.4 Debug Features

The DMA program register set, DMA source active register set, and the destination FIFO register set are used to derive a brief history of TRs serviced through the transfer controller.

Additionally, the EDMA3TC status register (TCSTAT) has dedicated bit fields to indicate the ongoing activity within different parts of the transfer controller:

- The SRCACTV bit indicates whether the source active set is active.
- The DSTACTV bit indicates the number of TRs resident in the destination register active set at a given instance.
- The PROGBUSY bit indicates whether a valid TR is present in the DMA program set.

If the TRs are in progression, caution must be used and you must realize that there is a chance that the values read from the EDMA3TC status registers will be inconsistent since the EDMA3TC may change the values of these registers due to ongoing activities.

It is recommended that you ensure no additional submission of TRs to the EDMA3TC in order to facilitate ease of debug.

##### *8.3.12.4.1 Destination FIFO Register Pointer*

The destination FIFO register pointer is implemented as a circular buffer with the start pointer being DFSTRTPTR and a buffer depth of usually 2 or 4. The EDMA3TC maintains two important status details in TCSTAT that may be used during advanced debugging, if necessary. The DFSTRTPTR is a start pointer, that is, the index to the head of the destination FIFO register. The DSTACTV is a counter for the number of valid (occupied) entries. These registers may be used to get a brief history of transfers.

Examples of some register field values and their interpretation:

- DFSTRTPTR = 0 and DSTACTV = 0 implies that no TRs are stored in the destination FIFO register.
- DFSTRTPTR = 1 and DSTACTV = 2h implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 1 and the second pending TR is read from the destination FIFO register entry 2.
- DFSTRTPTR = 3h and DSTACTV = 2h implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 3 and the second pending TR is read from the destination FIFO register entry 0.

#### 8.3.12.5 EDMA3TC Configuration

Table 8-18 provides the configuration of the individual EDMA3 transfer controllers present on the device. The DBS for each transfer controller is configurable using the TPTC_CFG register in the chip configuration module.

**Table 8-18. EDMA3 Transfer Controller Configurations**

| Name | TC0 | TC1 | TC2 | TC3 |
|---|---|---|---|---|
| FIFOSIZE | 1024 bytes | 1024 bytes | 1024 bytes | 1024 bytes |
| BUSWIDTH | 16 bytes | 16 bytes | 16 bytes | 16 bytes |
| DSTREGDEPTH | 4 entries | 4 entries | 4 entries | 4 entries |
| DBS | Configurable | Configurable | Configurable | Configurable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.3.13 Event Dataflow

This section summarizes the data flow of a single event, from the time the event is latched to the channel controller to the time the transfer completion code is returned. The following steps list the sequence of EDMA3CC activity:

1. Event is asserted from an external source (peripheral or external interrupt). This also is similar for a manually-triggered, chained-triggered, or QDMA-triggered event. The event is latched into the ER.E*n*/ERH.E*n* (or CER.E*n*/CERH.E*n*, ESR.E*n* /ESRH.E*n*, QER.E*n*) bit.
2. Once an event is prioritized and queued into the appropriate event queue, the SER.E*n*\SERH.E*n* (or QSER.E*n*) bit is set to inform the event prioritization/processing logic to disregard this event since it is already in the queue. Alternatively, if the transfer controller and the event queue are empty, then the event bypasses the queue.
3. The EDMA3CC processing and the submission logic evaluates the appropriate PaRAM set and determines whether it is a non-null and non-dummy transfer request (TR).
4. The EDMA3CC clears the ER.E*n*/ERH.E*n* (or CER.E*n*/CERH.E*n*, ESR.E*n*/ESRH.E*n*, QER.E*n*) bit and the SER.E*n*/SERH.E*n* bit as soon as it determines the TR is non-null. In the case of a null set, the SER.E*n*/SERH.E*n* bit remains set. It submits the non-null/non-dummy TR to the associated transfer controller. If the TR was programmed for early completion, the EDMA3CC immediately sets the interrupt pending register (IPR.I[TCC]/IPRH.I[TCC]-32).
5. If the TR was programmed for normal completion, the EDMA3CC sets the interrupt pending register (IPR.I[TCC]/IPRH.I[TCC]) when the EDMA3TC informs the EDMA3CC about completion of the transfer (returns transfer completion codes).
6. The EDMA3CC programs the associated EDMA3TC*n*'s Program Register Set with the TR.
7. The TR is then passed to the Source Active set and the DST FIFO Register Set, if both the register sets are available.
8. The Read Controller processes the TR by issuing read commands to the source slave endpoint. The Read Data lands in the Data FIFO of the EDMA3TC*n*.
9. As soon as sufficient data is available, the Write Controller begins processing the TR by issuing write commands to the destination slave endpoint.
10. This continues until the TR completes and the EDMA3TC*n* then signals completion status to the EDMA3CC.

### 8.3.14 EDMA3 Prioritization

The EDMA3 controller has many implementation rules to deal with concurrent events/channels, transfers, etc. The following subsections detail various arbitration details whenever there might be occurrence of concurrent activity. Figure 8-19 shows the different places EDMA3 priorities come into play.

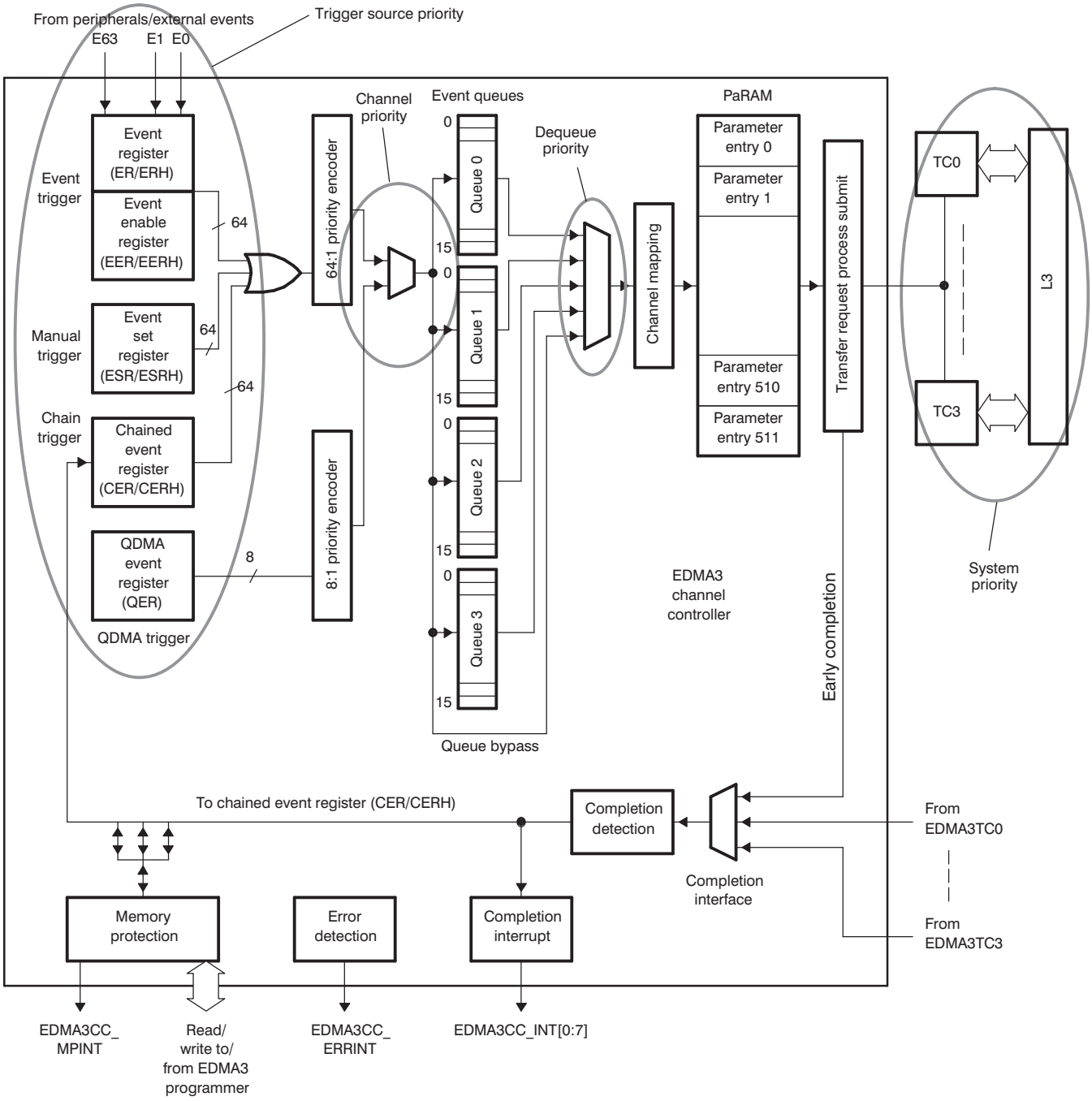
#### 8.3.14.1 Channel Priority

The DMA event registers (ER and ERH) capture up to 64 events; likewise, the QDMA event register (QER) captures QDMA events for all QDMA channels; therefore, it is possible for events to occur simultaneously on the DMA/QDMA event inputs. For events arriving simultaneously, the event associated with the lowest channel number is prioritized for submission to the event queues (for DMA events, channel 0 has the highest priority and channel 63 has the lowest priority; similarly, for QDMA events, channel 0 has the highest priority and channel 7 has the lowest priority). This mechanism only sorts simultaneous events for submission to the event queues.

If a DMA and QDMA event occurs simultaneously, the DMA event always has prioritization against the QDMA event for submission to the event queues.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-19. EDMA3 Prioritization**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.3.14.2 Trigger Source Priority

If a DMA channel is associated with more than one trigger source (event trigger, manual trigger, and chain trigger), and if multiple events are set simultaneously for the same channel (ER.E*n* = 1, ESR.E*n* = 1 , CER.E*n* = 1) , then the EDMA3CC always services these events in the following priority order: event trigger (via ER) is higher priority than chain trigger (via CER) and chain trigger is higher priority than manual trigger (via ESR).

This implies that if for channel 0, both ER.E0 = 1 and CER.E0 = 1 at the same time, then the ER.E0 event is always queued before the CER.E0 event.

#### 8.3.14.3 Dequeue Priority

The priority of the associated transfer request (TR) is further mitigated by which event queue is being used for event submission (dictated by DMAQNUM and QDMAQNUM). For submission of a TR to the transfer request, events need to be de-queued from the event queues. Queue 0 has the highest dequeue priority and queue 3 the lowest.

### *8.3.15 EDMA3 Operating Frequency (Clock Control)*

The EDMA3 channel controller and transfer controller are clocked from SYSCLK4. The EDMA3 system runs at the L3 clock frequency.

### *8.3.16 Reset Considerations*

A hardware reset resets the EDMA3 (EDMA3CC and EDMA3TC) and the EDMA3 configuration registers. The PaRAM memory contents are undefined after device reset and you should not rely on parameters to be reset to a known state. The PaRAM entry must be initialized to a desired value before it is used.

### *8.3.17 Power Management*

The EDMA3 (EDMA3CC and EDMA3TC) can be placed in reduced-power modes to conserve power during periods of low activity. The power management of the peripheral is controlled by the power reset clock management (PRCM). The PRCM acts as a master controller for power management for all peripherals on the device.

The EDMA3 controller can be idled on receiving a clock stop request from the PRCM. The requests to EDMA3CC and EDMA3TC are separate. In general, it should be verified that there are no pending activities in the EDMA3 controller

### *8.3.18 Emulation Considerations*

During debug when using the emulator, the CPU(s) may be halted on an execute packet boundary for single-stepping, benchmarking, profiling, or other debug purposes. During an emulation halt, the EDMA3 channel controller and transfer controller operations continue. Events continue to be latched and processed and transfer requests continue to be submitted and serviced.

Since EDMA3 is involved in servicing multiple master and slave peripherals, it is not feasible to have an independent behavior of the EDMA3 for emulation halts. EDMA3 functionality would be coupled with the peripherals it is servicing, which might have different behavior during emulation halts. For example, if a McASP is halted during an emulation access (FREE = 0 and SOFT = 0 or 1 in McASP registers), the McASP stops generating the McASP receive or transmit events (REVT or XEVT) to the EDMA. From the point of view of the McASP, the EDMA3 is suspended, but other peripherals (for example, a timer) still assert events and will be serviced by the EDMA.


Copyright © 2011–2016, Texas Instruments Incorporated

## 8.4 EDMA Transfer Examples

The EDMA3 channel controller performs a variety of transfers depending on the parameter configuration. The following sections provide a description and PaRAM configuration for some typical use case scenarios.

### *8.4.1 Block Move Example*

The most basic transfer performed by the EDMA3 is a block move. During device operation it is often necessary to transfer a block of data from one location to another, usually between on-chip and off-chip memory.

In this example, a section of data is to be copied from external memory to internal L2 SRAM as shown in Figure 8-20. Figure 8-21 shows the parameters for this transfer.

The source address for the transfer is set to the start of the data block in external memory, and the destination address is set to the start of the data block in L2. If the data block is less than 64K bytes, the PaRAM configuration shown in Figure 8-21 holds true with the synchronization type set to A-synchronized and indexes cleared to 0. If the amount of data is greater than 64K bytes, BCNT and the B-indexes need to be set appropriately with the synchronization type set to AB-synchronized. The STATIC bit in OPT is set to prevent linking.

This transfer example may also be set up using QDMA. For successive transfer submissions, of a similar nature, the number of cycles used to submit the transfer are fewer depending on the number of changing transfer parameters. You may program the QDMA trigger word to be the highest numbered offset in the PaRAM set that undergoes change.

**Figure 8-20. Block Move Example**

Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-21. Block Move Example PaRAM Configuration**

*(a) EDMA Parameters*

| Parameter Contents | |
|---|---|
| 0010 0008h | |
| Channel Source Address (SRC) | |
| 0001h | 0100h |
| Channel Destination Address (DST) | |
| 0000h | 0000h |
| 0000h | FFFFh |
| 0000h | 0000h |
| 0000h | 0001h |

| Parameter | |
|---|---|
| Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | |
| Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | |
| Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| BCNT Reload (BCNTRLD) | Link Address (LINK) |
| Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| Reserved | Count for 3rd Dimension (CCNT) |

*(b) Channel Options Parameter (OPT) Content*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 0 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0000 | 0 | 000 | 0000 | 1 | 0 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.4.2 Subframe Extraction Example

The EDMA3 can efficiently extract a small frame of data from a larger frame of data. By performing a 2D-to-1D transfer, the EDMA3 retrieves a portion of data for the CPU to process. In this example, a 640 × 480-pixel frame of video data is stored in external memory. Each pixel is represented by a 16-bit halfword. The CPU extracts a 16 × 12-pixel subframe of the image for processing. To facilitate more efficient processing time by the CPU, the EDMA3 places the subframe in internal L2 SRAM. Figure 8-22 shows the transfer of a subframe from external memory to L2. Figure 8-23 shows the parameters for this transfer.

The same PaRAM entry options are used for QDMA channels, as well as DMA channels. The STATIC bit in OPT is set to prevent linking. For successive transfers, only changed parameters need to be programmed before triggering the channel.

**Figure 8-22. Subframe Extraction Example**

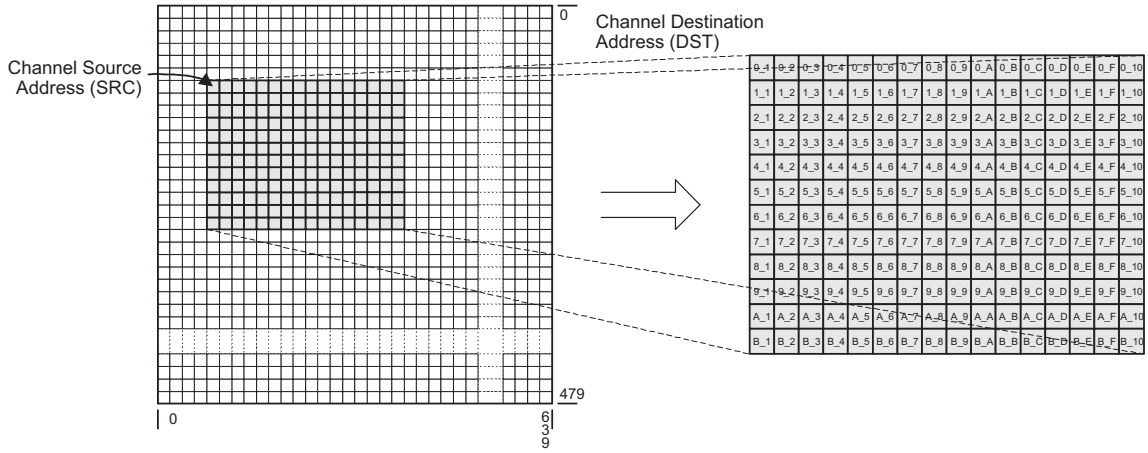



**Figure 8-23. Subframe Extraction Example PaRAM Configuration**

*(a) EDMA Parameters*

| Parameter Contents | |
|---|---|
| 0010 000Ch | |
| Channel Source Address (SRC) | |
| 000Ch | 0020h |
| Channel Destination Address (DST) | |
| 0020h | 0500h |
| 0000h | FFFFh |
| 0000h | 0000h |
| 0000h | 0001h |

| Parameter | |
|---|---|
| Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | |
| Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | |
| Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| BCNT Reload (BCNTRLD) | Link Address (LINK) |
| Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| Reserved | Count for 3rd Dimension (CCNT) |

*(b) Channel Options Parameter (OPT) Content*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 0 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0000 | 0 | 000 | 0000 | 1 | 1 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.4.3 Data Sorting Example

Many applications require the use of multiple data arrays; it is often desirable to have the arrays arranged such that the first elements of each array are adjacent, the second elements are adjacent, and so on. Often this is not how the data is presented to the device. Either data is transferred via a peripheral with the data arrays arriving one after the other or the arrays are located in memory with each array occupying a portion of contiguous memory spaces. For these instances, the EDMA3 can reorganize the data into the desired format. Figure 8-24 shows the data sorting.

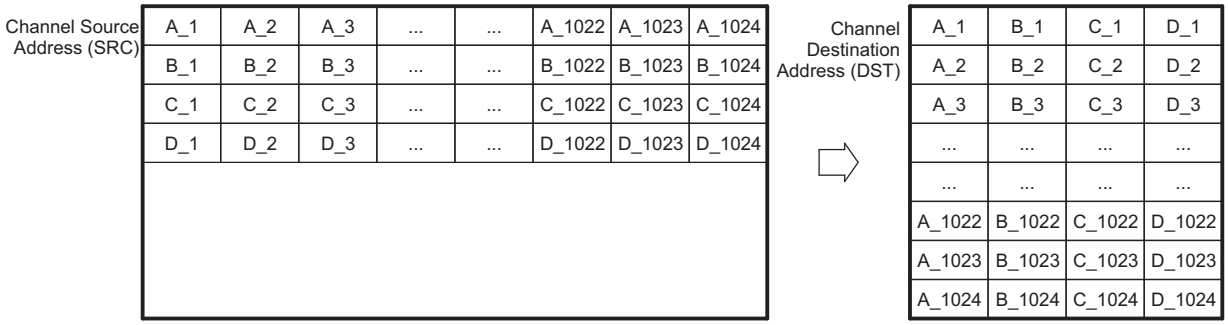
To determine the parameter set values, the following need to be considered:

- ACNT - Program this to be the size in bytes of an element.
- BCNT - Program this to be the number of elements in a frame.
- CCNT - Program this to be the number of frames.
- SRCBIDX - Program this to be the size of the element or ACNT.
- DSTBIDX - CCNT × ACNT
- SRCCDX - ACNT × BCNT
- DSTCIDX - ACNT

The synchronization type needs to be AB-synchronized and the STATIC bit is 0 to allow updates to the parameter set. It is advised to use normal EDMA3 channels for sorting.

It is not possible to sort this with a single trigger event. Instead, the channel can be programmed to be chained to itself. After BCNT elements get sorted, intermediate chaining could be used to trigger the channel again causing the transfer of the next BCNT elements and so on. Figure 8-25 shows the parameter set programming for this transfer, assuming channel 0 and an element size of 4 bytes.

**Figure 8-24. Data Sorting Example**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-25. Data Sorting Example PaRAM Configuration**

*(a) EDMA Parameters*

| Parameter Contents | |
|---|---|
| 0090 0004h | |
| Channel Source Address (SRC) | |
| 0400h | 0004h |
| Channel Destination Address (DST) | |
| 0010h | 0001h |
| 0000h | FFFFh |
| 0001h | 1000h |
| 0000h | 0004h |

| Parameter | |
|---|---|
| Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | |
| Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | |
| Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| BCNT Reload (BCNTRLD) | Link Address (LINK) |
| Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| Reserved | Count for 3rd Dimension (CCNT) |

*(b) Channel Options Parameter (OPT) Content*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 1 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0000 | 0 | 000 | 0000 | 0 | 1 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |



Copyright © 2011–2016, Texas Instruments Incorporated

### 8.4.4 Peripheral Servicing Example

The EDMA3 channel controller also services peripherals in the background of CPU operation, without requiring any CPU intervention. Through proper initialization of the EDMA3 channels, they can be configured to continuously service on-chip and off-chip peripherals throughout the device operation. Each event available to the EDMA3 has its own dedicated channel, and all channels operate simultaneously. The only requirements are to use the proper channel for a particular transfer and to enable the channel event in the event enable register (EER). When programming an EDMA3 channel to service a peripheral, it is necessary to know how data is to be presented to the processor. Data is always provided with some kind of synchronization event as either one element per event (non-bursting) or multiple elements per event (bursting).

#### 8.4.4.1 Non-bursting Peripherals

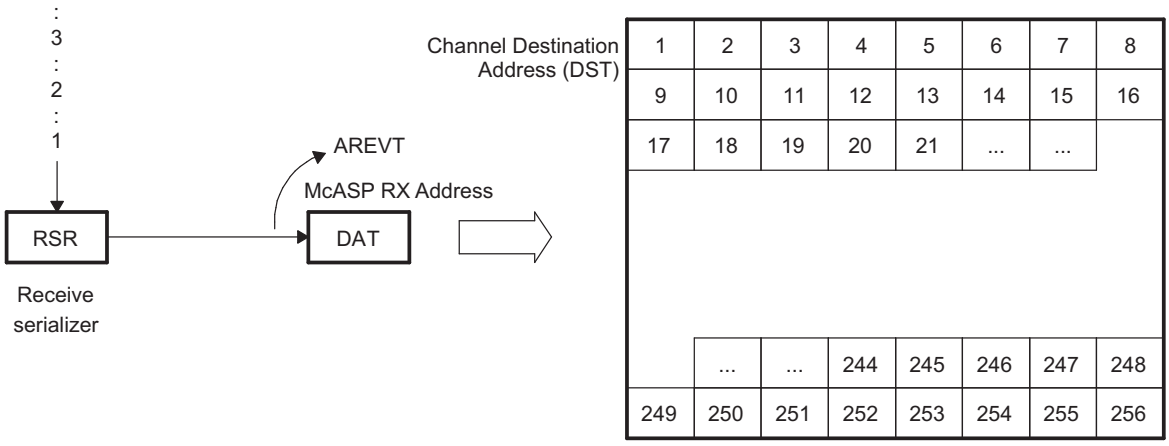
Non-bursting peripherals include the on-chip multichannel audio serial port (McASP) and many external devices, such as codecs. Regardless of the peripheral, the EDMA3 channel configuration is the same.

The McASP transmit and receive data streams are treated independently by the EDMA3. The transmit and receive data streams can have completely different counts, data sizes, and formats. Figure 8-26 shows servicing incoming McASP data.

To transfer the incoming data stream to its proper location in DDR memory, the EDMA3 channel must be set up for a 1D-to-1D transfer with A-synchronization. Because an event (AREVT) is generated for every word as it arrives, it is necessary to have the EDMA3 issue the transfer request for each element individually. Figure 8-27 shows the parameters for this transfer. The source address of the EDMA3 channel is set to the data port address(DAT) for McASP, and the destination address is set to the start of the data block in DDR. Because the address of serializer buffer is fixed, the source B index is cleared to 0 (no modification) and the destination B index is set to 01b (increment).

Based on the premise that serial data is typically a high priority, the EDMA3 channel should be programmed to be on queue 0.

**Figure 8-26. Servicing Incoming McASP Data Example**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-27. Servicing Incoming McASP Data Example PaRAM Configuration**

*(a) EDMA Parameters*

| Parameter Contents | |
|---|---|
| 0010 0000h | |
| McASP RX Address | |
| 0100h | 0001h |
| Channel Destination Address (DST) | |
| 0001h | 0000h |
| 0000h | FFFFh |
| 0000h | 0000h |
| 0000h | 0004h |

| Parameter | |
|---|---|
| Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | |
| Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | |
| Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| BCNT Reload (BCNTRLD) | Link Address (LINK) |
| Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| Reserved | Count for 3rd Dimension (CCNT) |

*(b) Channel Options Parameter (OPT) Content*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 0 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0000 | 0 | 000 | 0000 | 0 | 0 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |



Copyright © 2011–2016, Texas Instruments Incorporated

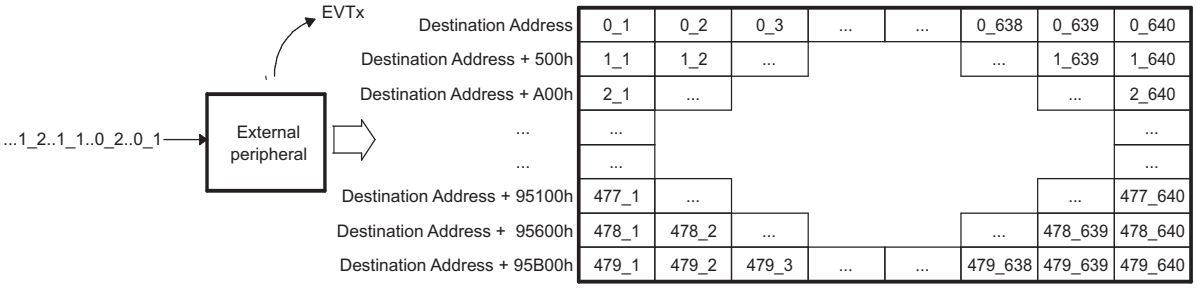
### 8.4.4.2 Bursting Peripherals

Higher bandwidth applications require that multiple data elements be presented to the processor core for every synchronization event. This frame of data can either be from multiple sources that are working simultaneously or from a single high-throughput peripheral that streams data to/from the processor.

In this example, a port is receiving a video frame from a camera and presenting it to the DSP one array at a time. The video image is 640 × 480 pixels, with each pixel represented by a 16-bit element. The image is to be stored in external memory. Figure 8-28 shows this example.

To transfer data from an external peripheral to an external buffer one array at a time based on EVT*n* , channel *n* must be configured. Due to the nature of the data (a video frame made up of arrays of pixels) the destination is essentially a 2D entity. Figure 8-29 shows the parameters to service the incoming data with a 1D-to-2D transfer using AB-synchronization. The source address is set to the location of the video framer peripheral, and the destination address is set to the start of the data buffer. Because the input address is static, the SRCBIDX is 0 (no modification to the source address). The destination is made up of arrays of contiguous, linear elements; therefore, the DSTBIDX is set to pixel size, 2 bytes. ANCT is equal to the pixel size, 2 bytes. BCNT is set to the number of pixels in an array, 640. CCNT is equal to the total number of arrays in the block, 480. SRCCIDX is 0 because the source address undergoes no increment. The DSTCIDX is equal to the difference between the starting addresses of each array. Because a pixel is 16 bits (2 bytes), DSTCIDX is equal to 640 × 2.

**Figure 8-28. Servicing Peripheral Burst Example**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-29. Servicing Peripheral Burst Example PaRAM Configuration**

*(a) EDMA Parameters*

| Parameter Contents | Parameter |
|---|---|
| 0010 0004h | Channel Options Parameter (OPT) |
| Channel Source Address | Channel Source Address (SRC) |
| 0280h \| 0002h | Count for 2nd Dimension (BCNT) \| Count for 1st Dimension (ACNT) |
| Channel Destination Address | Channel Destination Address (DST) |
| 0002h \| 0000h | Destination BCNT Index (DSTBIDX) \| Source BCNT Index (SRCBIDX) |
| 0000h \| FFFFh | BCNT Reload (BCNTRLD) \| Link Address (LINK) |
| 0500h \| 0000h | Destination CCNT Index (DSTCIDX) \| Source CCNT Index (SRCCIDX) |
| 0000h \| 01E0h | Reserved \| Count for 3rd Dimension (CCNT) |

*(b) Channel Options Parameter (OPT) Content*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 0 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0000 | 0 | 000 | 0000 | 0 | 1 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.4.4.3 Continuous Operation

Configuring an EDMA3 channel to receive a single frame of data is useful, and is applicable to some systems. A majority of the time, however, data is going to be continuously transmitted and received throughout the entire operation of the processor. In this case, it is necessary to implement some form of linking such that the EDMA3 channels continuously reload the necessary parameter sets. In this example, McASP is configured to transmit and receive data on a T1 array. To simplify the example, only two channels are active for both transmit and receive data streams. Each channel receives packets of 128 elements. The packets are transferred from the serial port to internal memory and from internal memory to the serial port, as shown Figure 8-30.

The McASP generates AREVT for every element received and generates AXEVT for every element transmitted. To service the data streams, them DMA channels associated with the McASP (channels 12 and 15) must be setup for 1D-to-1D transfers with A-synchronization.

Figure 8-31 shows the parameter entries for the channel for these transfers. To service the McASP continuously throughout DSP operation, the channels must be linked to a duplicate PaRAM set in the PaRAM. After all frames have been transferred, the EDMA3 channels reload and continue. Figure 8-32 shows the reload parameters for the channel.

#### *8.4.4.3.1 Receive Channel*

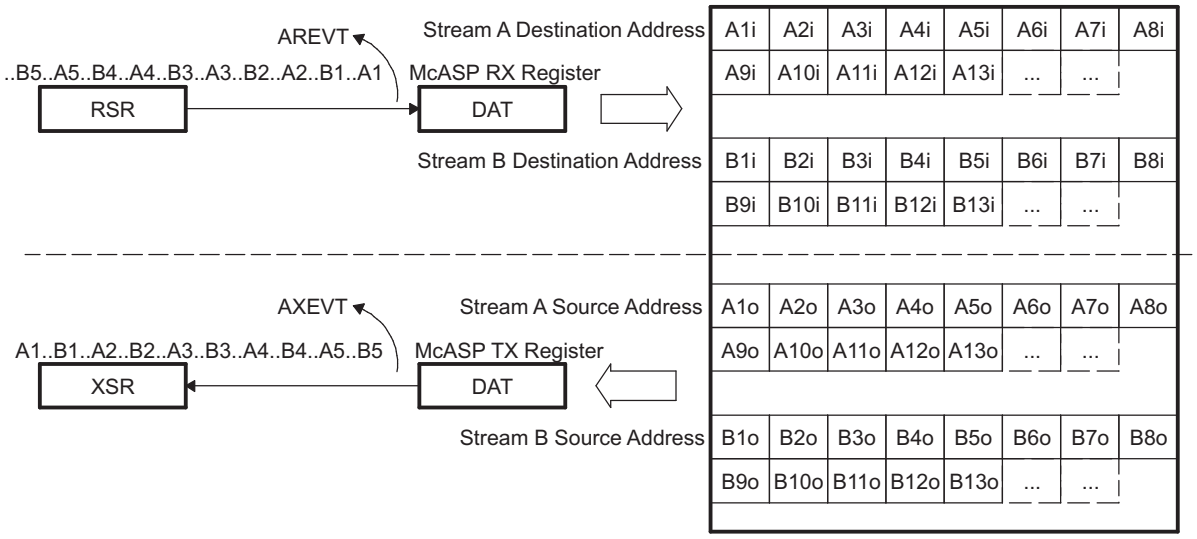
EDMA3 channel 15 services the incoming data stream of McASP. The source address is set to that of the receive serializer buffer, and the destination address is set to the first element of the data block. Because there are two data channels being serviced, A and B, they are to be located separately within the L2 SRAM.

To facilitate continuous operation, a copy of the PaRAM set for the channel is placed in PaRAM set 64. The LINK option is set and the link address is provided in the PaRAM set. Upon exhausting the channel 15 parameter set, the parameters located at the link address are loaded into the channel 15 parameter set and operation continues. This function continues throughout device operation until halted by the CPU.

#### *8.4.4.3.2 Transmit Channel*

EDMA3 channel 12 services the outgoing data stream of McASP. In this case the destination address needs no update, hence, the parameter set changes accordingly. Linking is also used to allow continuous operation by the EDMA3 channel, with duplicate PaRAM set entries at PaRAM set 65.

**Figure 8-30. Servicing Continuous McASP Data Example**





Copyright © 2011–2016, Texas Instruments Incorporated

## Figure 8-31. Servicing Continuous McASP Data Example PaRAM Configuration

*(a) EDMA Parameters for Receive Channel (PaRAM Set 15) being Linked to PaRAM Set 64*

| Parameter Contents | |
|---|---|
| 0010 0000h | |
| McASP RX Register | |
| 0080h | 0001h |
| Channel Destination Address (DST) | |
| 0001h | 0000h |
| 0080h | 4800h |
| 0000h | 0000h |
| 0000h | FFFFh |

| Parameter | |
|---|---|
| Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | |
| Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | |
| Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| BCNT Reload (BCNTRLD) | Link Address (LINK) |
| Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| Reserved | Count for 3rd Dimension (CCNT) |

*(b) Channel Options Parameter (OPT) Content for Receive Channel (PaRAM Set 15)*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 0 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0000 | 0 | 000 | 0000 | 0 | 0 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |

*(c) EDMA Parameters for Transmit Channel (PaRAM Set 12) being Linked to PaRAM Set 65*

| Parameter Contents | |
|---|---|
| 0010 1000h | |
| Channel Source Address (SRC) | |
| 0080h | 0001h |
| McASP TX Register | |
| 0000h | 0001h |
| 0080h | 4860h |
| 0000h | 0000h |
| 0000h | FFFFh |

| Parameter | |
|---|---|
| Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | |
| Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | |
| Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| BCNT Reload (BCNTRLD) | Link Address (LINK) |
| Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| Reserved | Count for 3rd Dimension (CCNT) |

*(d) Channel Options Parameter (OPT) Content for Transmit Channel (PaRAM Set 12)*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 0 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0001 | 0 | 000 | 0000 | 0 | 0 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-32. Servicing Continuous McASP Data Example Reload PaRAM Configuration**

*(a) EDMA Reload Parameters (PaRAM Set 64) for Receive Channel*

| Parameter Contents | |
|---|---|
| 0010 0000h | |
| McASP RX Register | |
| 0080h | 0001h |
| Channel Destination Address (DST) | |
| 0001h | 0000h |
| 0080h | 4800h |
| 0000h | 0000h |
| 0000h | FFFFh |

| Parameter | |
|---|---|
| Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | |
| Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | |
| Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| BCNT Reload (BCNTRLD) | Link Address (LINK) |
| Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| Reserved | Count for 3rd Dimension (CCNT) |

*(b) Channel Options Parameter (OPT) Content for Receive Channel (PaRAM Set 64)*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 0 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0000 | 0 | 000 | 0000 | 0 | 0 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |

*(c) EDMA Reload Parameters (PaRAM Set 65) for Transmit Channel*

| Parameter Contents | |
|---|---|
| 0010 1000h | |
| Channel Source Address (SRC) | |
| 0080h | 0001h |
| McASP TX Register | |
| 0000h | 0001h |
| 0080h | 4860h |
| 0000h | 0000h |
| 0000h | FFFFh |

| Parameter | |
|---|---|
| Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | |
| Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | |
| Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| BCNT Reload (BCNTRLD) | Link Address (LINK) |
| Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| Reserved | Count for 3rd Dimension (CCNT) |

*(d) Channel Options Parameter (OPT) Content for Transmit Channel (PaRAM Set 65)*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 0 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0001 | 0 | 000 | 0000 | 0 | 0 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.4.4.4 Ping-Pong Buffering

Although the previous configuration allows the EDMA3 to service a peripheral continuously, it presents a number of restrictions to the CPU. Because the input and output buffers are continuously being filled/emptied, the CPU must match the pace of the EDMA3 very closely to process the data. The EDMA3 receive data must always be placed in memory before the CPU accesses it, and the CPU must provide the output data before the EDMA3 transfers it. Though not impossible, this is an unnecessary challenge. It is particularly difficult in a 2-level cache scheme.

Ping-pong buffering is a simple technique that allows the CPU activity to be distanced from the EDMA3 activity. This means that there are multiple (usually two) sets of data buffers for all incoming and outgoing data streams. While the EDMA3 transfers the data into and out of the ping buffers, the CPU manipulates the data in the pong buffers. When both CPU and EDMA3 activity completes, they switch. The EDMA3 then writes over the old input data and transfers the new output data. Figure 8-33 shows the ping-pong scheme for this example.

To change the continuous operation example, such that a ping-pong buffering scheme is used, the EDMA3 channels need only a moderate change. Instead of one parameter set, there are two; one for transferring data to/from the ping buffers and one for transferring data to/from the pong buffers. As soon as one transfer completes, the channel loads the PaRAM set for the other and the data transfers continue. Figure 8-34 shows the EDMA3 channel configuration required.

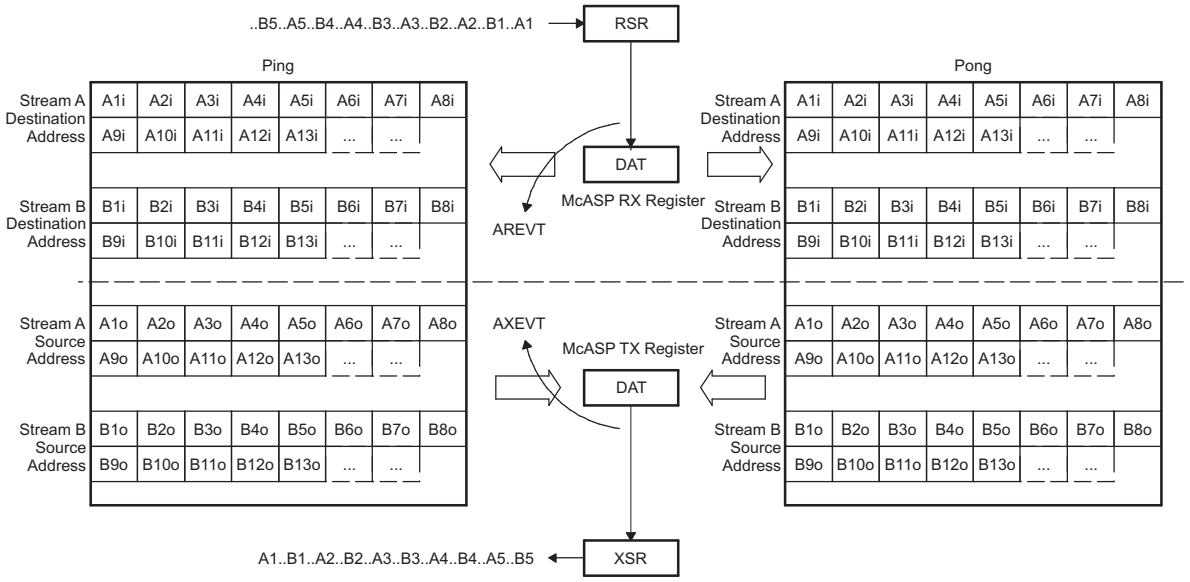
Each channel has two parameter sets, ping and pong. The EDMA3 channel is initially loaded with the ping parameters (Figure 8-34). The link address for the ping set is set to the PaRAM offset of the pong parameter set (Figure 8-35). The link address for the pong set is set to the PaRAM offset of the ping parameter set (Figure 8-36). The channel options, count values, and index values are all identical between the ping and pong parameters for each channel. The only differences are the link address provided and the address of the data buffer.

##### *8.4.4.4.1 Synchronization with the CPU*

To utilize the ping-pong buffering technique, the system must signal the CPU when to begin to access the new data set. After the CPU finishes processing an input buffer (ping), it waits for the EDMA3 to complete before switching to the alternate (pong) buffer. In this example, both channels provide their channel numbers as their report word and set the TCINTEN bit to generate an interrupt after completion. When channel 15 fills an input buffer, the E15 bit in the interrupt pending register (IPR) is set; when channel 12 empties an output buffer, the E12 bit in IPR is set. The CPU must manually clear these bits. With the channel parameters set, the CPU polls IPR to determine when to switch. The EDMA3 and CPU could alternatively be configured such that the channel completion interrupts the CPU. By doing this, the CPU could service a background task while waiting for the EDMA3 to complete.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-33. Ping-Pong Buffering for McASP Data Example**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-34. Ping-Pong Buffering for McASP Example PaRAM Configuration**

*(a) EDMA Parameters for Channel 15 (Using PaRAM Set 15 Linked to Pong Set 64)*

| Parameter Contents | |
|---|---|
| 0010 D000h | |
| McASP RX Register | |
| 0080h | 0001h |
| Channel Destination Address (DST) | |
| 0001h | 0000h |
| 0080h | 4800h |
| 0000h | 0000h |
| 0000h | 0001h |

| Parameter | |
|---|---|
| Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | |
| Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | |
| Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| BCNT Reload (BCNTRLD) | Link Address (LINK) |
| Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| Reserved | Count for 3rd Dimension (CCNT) |

*(b) Channel Options Parameter (OPT) Content for Channel 15*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 0 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 1101 | 0 | 000 | 0000 | 0 | 0 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |

*(c) EDMA Parameters for Channel 12 (Using PaRAM Set 12 Linked to Pong Set 65)*

| Parameter Contents | |
|---|---|
| 0010 C000h | |
| Channel Source Address (SRC) | |
| 0080h | 0001h |
| McASP TX Register | |
| 0000h | 0001h |
| 0080h | 4840h |
| 0000h | 0000h |
| 0000h | 0001h |

| Parameter | |
|---|---|
| Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | |
| Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | |
| Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| BCNT Reload (BCNTRLD) | Link Address (LINK) |
| Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| Reserved | Count for 3rd Dimension (CCNT) |

*(d) Channel Options Parameter (OPT) Content for Channel 12*

| 31 | 30 28 | 27 24 | 23 | 22 | 21 | 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 0000 | 0 | 0 | 0 | 1 | 00 | 00 |
| PRIV | Reserved | PRIVID | ITCCHEN | TCCHEN | ITCINTEN | TCINTEN | Reserved | TCC |

| 15 12 | 11 | 10 8 | 7 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 1100 | 0 | 000 | 0000 | 0 | 0 | 0 | 0 |
| TCC | TCCMOD | FWID | Reserved | STATIC | SYNCDIM | DAM | SAM |


Copyright © 2011–2016, Texas Instruments Incorporated

## Figure 8-35. Ping-Pong Buffering for McASP Example Pong PaRAM Configuration

*(a) EDMA Pong Parameters for Channel 15 at Set 64 Linked to Set 65*

| Parameter Contents | | Parameter | |
|---|---|---|---|
| 0010 D000h | | Channel Options Parameter (OPT) | |
| McASP RX Register | | Channel Source Address (SRC) | |
| 0080h | 0001h | Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | | Channel Destination Address (DST) | |
| 0001h | 0000h | Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| 0080h | 4820h | BCNT Reload (BCNTRLD) | Link Address (LINK) |
| 0000h | 0000h | Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| 0000h | 0001h | Reserved | Count for 3rd Dimension (CCNT) |

*(b) EDMA Pong Parameters for Channel 12 at Set 66 Linked to Set 67*

| Parameter Contents | | Parameter | |
|---|---|---|---|
| 0010 C000h | | Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | | Channel Source Address (SRC) | |
| 0080h | 0001h | Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| McASP TX Register | | Channel Destination Address (DST) | |
| 0000h | 0001h | Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| 0080h | 4860h | BCNT Reload (BCNTRLD) | Link Address (LINK) |
| 0000h | 0000h | Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| 0000h | 0001h | Reserved | Count for 3rd Dimension (CCNT) |

## Figure 8-36. Ping-Pong Buffering for McASP Example Ping PaRAM Configuration

*(a) EDMA Ping Parameters for Channel 15 at Set 65 Linked to Set 64*

| Parameter Contents | | Parameter | |
|---|---|---|---|
| 0010 D000h | | Channel Options Parameter (OPT) | |
| McASP RX Register | | Channel Source Address (SRC) | |
| 0080h | 0001h | Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| Channel Destination Address (DST) | | Channel Destination Address (DST) | |
| 0001h | 0000h | Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| 0080h | 4800h | BCNT Reload (BCNTRLD) | Link Address (LINK) |
| 0000h | 0000h | Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| 0000h | 0001h | Reserved | Count for 3rd Dimension (CCNT) |

*(b) EDMA Ping Parameters for Channel 12 at Set 67 Linked to Set 66*

| Parameter Contents | | Parameter | |
|---|---|---|---|
| 0010 C000h | | Channel Options Parameter (OPT) | |
| Channel Source Address (SRC) | | Channel Source Address (SRC) | |
| 0080h | 0001h | Count for 2nd Dimension (BCNT) | Count for 1st Dimension (ACNT) |
| McASP TX Register | | Channel Destination Address (DST) | |
| 0000h | 0001h | Destination BCNT Index (DSTBIDX) | Source BCNT Index (SRCBIDX) |
| 0080h | 4840h | BCNT Reload (BCNTRLD) | Link Address (LINK) |
| 0000h | 0000h | Destination CCNT Index (DSTCIDX) | Source CCNT Index (SRCCIDX) |
| 0000h | 0001h | Reserved | Count for 3rd Dimension (CCNT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.4.4.5 Transfer Chaining Examples

The following examples explain the intermediate transfer complete chaining function.

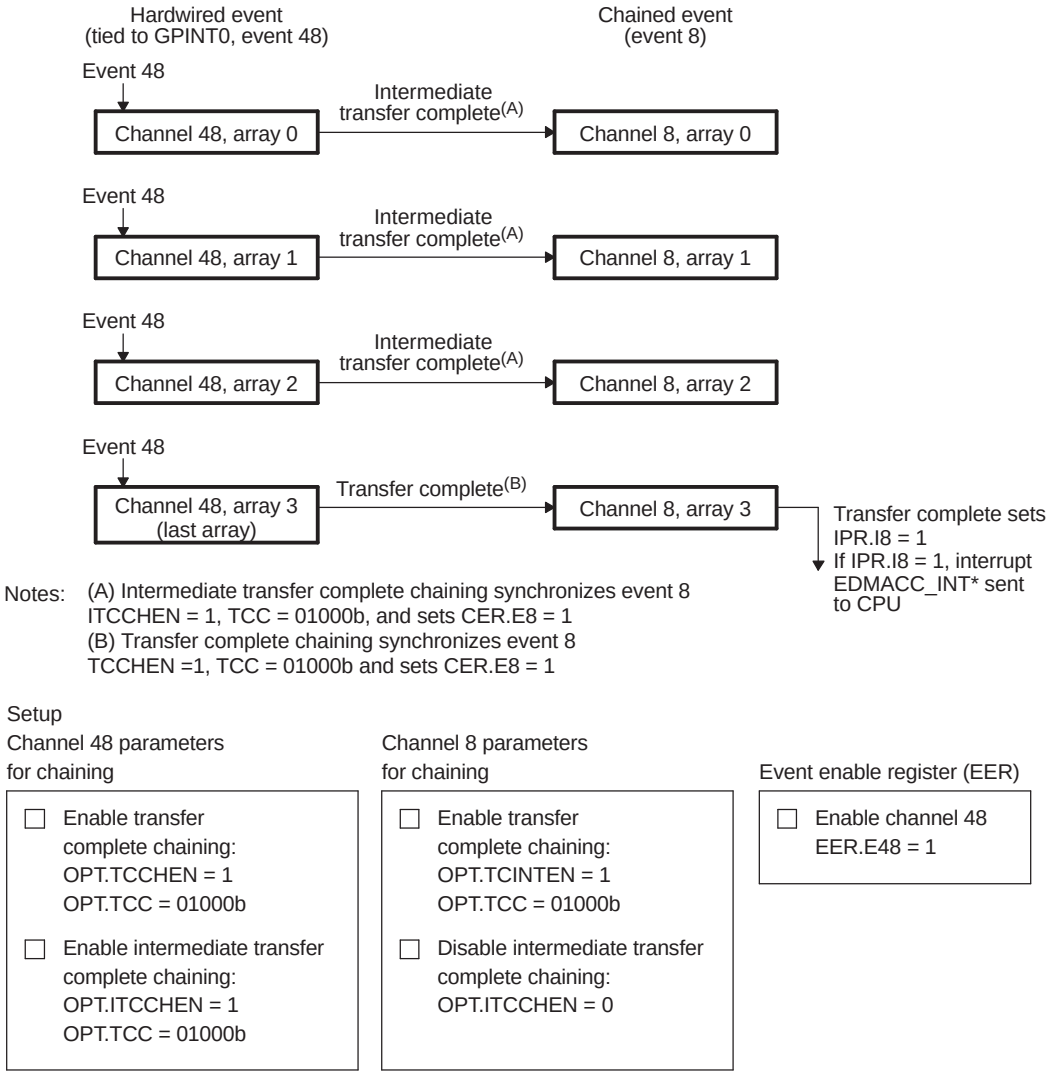
#### *8.4.4.5.1 Servicing Input/Output FIFOs with a Single Event*

Many systems require the use of a pair of external FIFOs that must be serviced at the same rate. One FIFO buffers data input, and the other buffers data output. The EDMA3 channels that service these FIFOs can be set up for AB-synchronized transfers. While each FIFO is serviced with a different set of parameters, both can be signaled from a single event. For example, an external interrupt pin can be tied to the status flags of one of the FIFOs. When this event arrives, the EDMA3 needs to perform servicing for both the input and output streams. Without the intermediate transfer complete chaining feature this would require two events, and thus two external interrupt pins. The intermediate transfer complete chaining feature allows the use of a single external event (for example, a GPIO event). Figure 8-37 shows the EDMA3 setup and illustration for this example.

A GPIO event (in this case, GPINT0) triggers an array transfer. Upon completion of each intermediate array transfer of channel 48, intermediate transfer complete chaining sets the E8 bit (specified by TCC of 8) in the chained event register (CER) and provides a synchronization event to channel 8. Upon completion of the last array transfer of channel 48, transfer complete chaining—not intermediate transfer complete chaining—sets the E8 bit in CER (specified by TCCMODE:TCC) and provides a synchronization event to channel 8. The completion of channel 8 sets the I8 bit (specified by TCCMODE:TCC) in the interrupt pending register (IPR), which can generate an interrupt to the CPU, if the I8 bit in the interrupt enable register (IER) is set.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-37. Intermediate Transfer Completion Chaining Example**

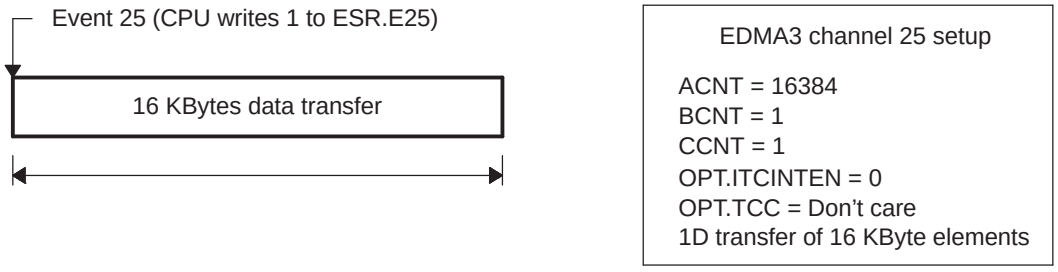



#### 8.4.4.5.2 Breaking Up Large Transfers with Intermediate Chaining

Another feature of intermediate transfer chaining (ITCCHEN) is for breaking up large transfers. A large transfer may lock out other transfers of the same priority level for the duration of the transfer. For example, a large transfer on queue 0 from the internal memory to the external memory using the EMIF may starve other EDMA3 transfers on the same queue. In addition, this large high-priority transfer may prevent the EMIF for a long duration to service other lower priority transfers. When a large transfer is considered to be high priority, it should be split into multiple smaller transfers. Figure 8-38 shows the EDMA3 setup and illustration of an example single large block transfer.

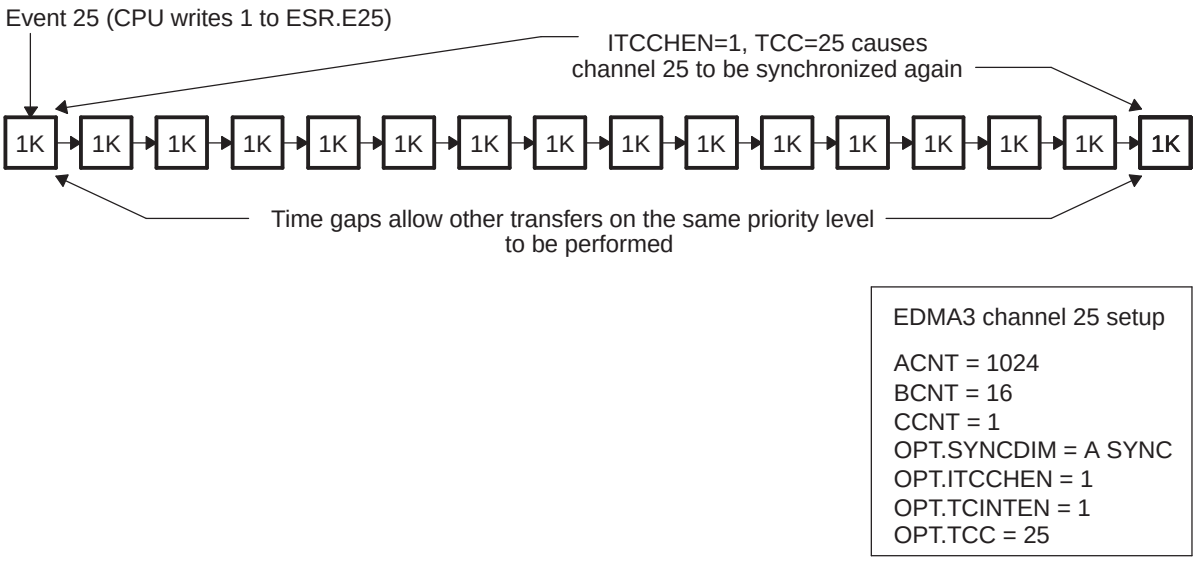

Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-38. Single Large Block Transfer Example**




The intermediate transfer chaining enable (ITCCHEN) provides a method to break up a large transfer into smaller transfers. For example, to move a single large block of memory (16K bytes), the EDMA3 performs an A-synchronized transfer. The element count is set to a reasonable value, where reasonable derives from the amount of time it would take to move this smaller amount of data. Assume 1 Kbyte is a reasonable small transfer in this example. The EDMA3 is set up to transfer 16 arrays of 1 Kbyte elements, for a total of 16K byte elements. The TCC field in the channel options parameter (OPT) is set to the same value as the channel number and ITCCHEN are set. In this example, EDMA3 channel 25 is used and TCC is also set to 25. The TCINTEN may also be set to trigger interrupt 25 when the last 1 Kbyte array is transferred. The CPU starts the EDMA3 transfer by writing to the appropriate bit of the event set register (ESR.E25). The EDMA3 transfers the first 1 Kbyte array. Upon completion of the first array, intermediate transfer complete code chaining generates a synchronization event to channel 25, a value specified by the TCC field. This intermediate transfer completion chaining event causes EDMA3 channel 25 to transfer the next 1 Kbyte array. This process continues until the transfer parameters are exhausted, at which point the EDMA3 has completed the 16K byte transfer. This method breaks up a large transfer into smaller packets, thus providing natural time slices in the transfer such that other events may be processed. Figure 8-39 shows the EDMA3 setup and illustration of the broken up smaller packet transfers.

**Figure 8-39. Smaller Packet Data Transfers Example**





Copyright © 2011–2016, Texas Instruments Incorporated

## 8.5 EDMA3 Registers

### 8.5.1 EDMA3 Channel Controller Registers

Table 8-19 lists the memory-mapped registers for the EDMA3 channel controller (EDMACC). All other register offset addresses not listed in Table 8-19 should be considered as reserved locations and the register contents should not be modified. See your device-specific data manual for the register memory maps.

**Table 8-19. EDMACC Registers**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 00h | PID | Peripheral Identification Register | Section 8.5.1.1.1 |
| 04h | CCCFG | EDMA3CC Configuration Register | Section 8.5.1.1.2 |
| 0100h-01FCh | DCHMAP0-63 | DMA Channel 0-63 Mapping Registers | Section 8.5.1.1.3 |
| 0200h | QCHMAP0 | QDMA Channel 0 Mapping Register | Section 8.5.1.1.4 |
| 0204h | QCHMAP1 | QDMA Channel 1 Mapping Register | Section 8.5.1.1.4 |
| 0208h | QCHMAP2 | QDMA Channel 2 Mapping Register | Section 8.5.1.1.4 |
| 020Ch | QCHMAP3 | QDMA Channel 3 Mapping Register | Section 8.5.1.1.4 |
| 0210h | QCHMAP4 | QDMA Channel 4 Mapping Register | Section 8.5.1.1.4 |
| 0214h | QCHMAP5 | QDMA Channel 5 Mapping Register | Section 8.5.1.1.4 |
| 0218h | QCHMAP6 | QDMA Channel 6 Mapping Register | Section 8.5.1.1.4 |
| 021Ch | QCHMAP7 | QDMA Channel 7 Mapping Register | Section 8.5.1.1.4 |
| 0240h | DMAQNUM0 | DMA Queue Number Register 0 | Section 8.5.1.1.5 |
| 0244h | DMAQNUM1 | DMA Queue Number Register 1 | Section 8.5.1.1.5 |
| 0248h | DMAQNUM2 | DMA Queue Number Register 2 | Section 8.5.1.1.5 |
| 024Ch | DMAQNUM3 | DMA Queue Number Register 3 | Section 8.5.1.1.5 |
| 0250h | DMAQNUM4 | DMA Queue Number Register 4 | Section 8.5.1.1.5 |
| 0254h | DMAQNUM5 | DMA Queue Number Register 5 | Section 8.5.1.1.5 |
| 0258h | DMAQNUM6 | DMA Queue Number Register 6 | Section 8.5.1.1.5 |
| 025Ch | DMAQNUM7 | DMA Queue Number Register 7 | Section 8.5.1.1.5 |
| 0260h | QDMAQNUM | QDMA Queue Number Register | Section 8.5.1.1.6 |
| 0300h | EMR | Event Missed Register | Section 8.5.1.2.1 |
| 0304h | EMRH | Event Missed Register High | Section 8.5.1.2.1 |
| 0308h | EMCR | Event Missed Clear Register | Section 8.5.1.2.2 |
| 030Ch | EMCRH | Event Missed Clear Register High | Section 8.5.1.2.2 |
| 0310h | QEMR | QDMA Event Missed Register | Section 8.5.1.2.3 |
| 0314h | QEMCR | QDMA Event Missed Clear Register | Section 8.5.1.2.4 |
| 0318h | CCERR | EDMA3CC Error Register | Section 8.5.1.2.5 |
| 031Ch | CCERRCLR | EDMA3CC Error Clear Register | Section 8.5.1.2.6 |
| 0320h | EEVAL | Error Evaluate Register | Section 8.5.1.2.7 |
| 0340h | DRAE0 | DMA Region Access Enable Register for Region 0 | Section 8.5.1.3.1 |
| 0344h | DRAEH0 | DMA Region Access Enable Register High for Region 0 | Section 8.5.1.3.1 |
| 0348h | DRAE1 | DMA Region Access Enable Register for Region 1 | Section 8.5.1.3.1 |
| 034Ch | DRAEH1 | DMA Region Access Enable Register High for Region 1 | Section 8.5.1.3.1 |
| 0350h | DRAE2 | DMA Region Access Enable Register for Region 2 | Section 8.5.1.3.1 |
| 0354h | DRAEH2 | DMA Region Access Enable Register High for Region 2 | Section 8.5.1.3.1 |
| 0358h | DRAE3 | DMA Region Access Enable Register for Region 3 | Section 8.5.1.3.1 |
| 035Ch | DRAEH3 | DMA Region Access Enable Register High for Region 3 | Section 8.5.1.3.1 |
| 0360h | DRAE4 | DMA Region Access Enable Register for Region 4 | Section 8.5.1.3.1 |
| 0364h | DRAEH4 | DMA Region Access Enable Register High for Region 4 | Section 8.5.1.3.1 |
| 0368h | DRAE5 | DMA Region Access Enable Register for Region 5 | Section 8.5.1.3.1 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-19. EDMACC Registers (continued)**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 036Ch | DRAEH5 | DMA Region Access Enable Register High for Region 5 | Section 8.5.1.3.1 |
| 0370h | DRAE6 | DMA Region Access Enable Register for Region 6 | Section 8.5.1.3.1 |
| 0374h | DRAEH6 | DMA Region Access Enable Register High for Region 6 | Section 8.5.1.3.1 |
| 0378h | DRAE7 | DMA Region Access Enable Register for Region 7 | Section 8.5.1.3.1 |
| 037Ch | DRAEH7 | DMA Region Access Enable Register High for Region 7 | Section 8.5.1.3.1 |
| 0380h-039Ch | QRAE0-7 | QDMA Region Access Enable Registers for Region 0-7 | Section 8.5.1.3.2 |
| 0400h-04FCh | Q0E0-Q3E15 | Event Queue Entry Registers Q0E0-Q3E15 | Section 8.5.1.4.1 |
| 0600h-060Ch | QSTAT0-3 | Queue Status Registers 0-3 | Section 8.5.1.4.2 |
| 0620h | QWMTHRA | Queue Watermark Threshold A Register | Section 8.5.1.4.3 |
| 0640h | CCSTAT | EDMA3CC Status Register | Section 8.5.1.4.4 |
| 0800h | MPFAR | Memory Protection Fault Address Register | Section 8.5.1.5.1 |
| 0804h | MPFSR | Memory Protection Fault Status Register | Section 8.5.1.5.2 |
| 0808h | MPFCR | Memory Protection Fault Command Register | Section 8.5.1.5.3 |
| 080Ch | MPPAG | Memory Protection Page Attribute Register Global | Section 8.5.1.5.4 |
| 0810h-082Ch | MPPA0-7 | Memory Protection Page Attribute Registers 0-7 | Section 8.5.1.5.4 |
| 1000h | ER | Event Register | Section 8.5.1.6.1 |
| 1004h | ERH | Event Register High | Section 8.5.1.6.1 |
| 1008h | ECR | Event Clear Register | Section 8.5.1.6.2 |
| 100Ch | ECRH | Event Clear Register High | Section 8.5.1.6.2 |
| 1010h | ESR | Event Set Register | Section 8.5.1.6.3 |
| 1014h | ESRH | Event Set Register High | Section 8.5.1.6.3 |
| 1018h | CER | Chained Event Register | Section 8.5.1.6.4 |
| 101Ch | CERH | Chained Event Register High | Section 8.5.1.6.4 |
| 1020h | EER | Event Enable Register | Section 8.5.1.6.5 |
| 1024h | EERH | Event Enable Register High | Section 8.5.1.6.5 |
| 1028h | EECR | Event Enable Clear Register | Section 8.5.1.6.6 |
| 102Ch | EECRH | Event Enable Clear Register High | Section 8.5.1.6.6 |
| 1030h | EESR | Event Enable Set Register | Section 8.5.1.6.7 |
| 1034h | EESRH | Event Enable Set Register High | Section 8.5.1.6.7 |
| 1038h | SER | Secondary Event Register | Section 8.5.1.6.8 |
| 103Ch | SERH | Secondary Event Register High | Section 8.5.1.6.8 |
| 1040h | SECR | Secondary Event Clear Register | Section 8.5.1.6.9 |
| 1044h | SECRH | Secondary Event Clear Register High | Section 8.5.1.6.9 |
| 1050h | IER | Interrupt Enable Register | Section 8.5.1.7.1 |
| 1054h | IERH | Interrupt Enable Register High | Section 8.5.1.7.1 |
| 1058h | IECR | Interrupt Enable Clear Register | Section 8.5.1.7.2 |
| 105Ch | IECRH | Interrupt Enable Clear Register High | Section 8.5.1.7.2 |
| 1060h | IESR | Interrupt Enable Set Register | Section 8.5.1.7.3 |
| 1064h | IESRH | Interrupt Enable Set Register High | Section 8.5.1.7.3 |
| 1068h | IPR | Interrupt Pending Register | Section 8.5.1.7.4 |
| 106Ch | IPRH | Interrupt Pending Register High | Section 8.5.1.7.4 |
| 1070h | ICR | Interrupt Clear Register | Section 8.5.1.7.5 |
| 1074h | ICRH | Interrupt Clear Register High | Section 8.5.1.7.5 |
| 1078h | IEVAL | Interrupt Evaluate Register | Section 8.5.1.7.6 |
| 1080h | QER | QDMA Event Register | Section 8.5.1.8.1 |
| 1084h | QEER | QDMA Event Enable Register | Section 8.5.1.8.2 |
| 1088h | QEECR | QDMA Event Enable Clear Register | Section 8.5.1.8.3 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-19. EDMACC Registers (continued)**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 108Ch | QEESR | QDMA Event Enable Set Register | Section 8.5.1.8.4 |
| 1090h | QSER | QDMA Secondary Event Register | Section 8.5.1.8.5 |
| 1094h | QSECR | QDMA Secondary Event Clear Register | Section 8.5.1.8.6 |
| **Shadow Region 0 Channel Registers** | | | |
| 2000h | ER | Event Register | |
| 2004h | ERH | Event Register High | |
| 2008h | ECR | Event Clear Register | |
| 200Ch | ECRH | Event Clear Register High | |
| 2010h | ESR | Event Set Register | |
| 2014h | ESRH | Event Set Register High | |
| 2018h | CER | Chained Event Register | |
| 201Ch | CERH | Chained Event Register High | |
| 2020h | EER | Event Enable Register | |
| 2024h | EERH | Event Enable Register High | |
| 2028h | EECR | Event Enable Clear Register | |
| 202Ch | EECRH | Event Enable Clear Register High | |
| 2030h | EESR | Event Enable Set Register | |
| 2034h | EESRH | Event Enable Set Register High | |
| 2038h | SER | Secondary Event Register | |
| 203Ch | SERH | Secondary Event Register High | |
| 2040h | SECR | Secondary Event Clear Register | |
| 2044h | SECRH | Secondary Event Clear Register High | |
| 2050h | IER | Interrupt Enable Register | |
| 2054h | IERH | Interrupt Enable Register High | |
| 2058h | IECR | Interrupt Enable Clear Register | |
| 205Ch | IECRH | Interrupt Enable Clear Register High | |
| 2060h | IESR | Interrupt Enable Set Register | |
| 2064h | IESRH | Interrupt Enable Set Register High | |
| 2068h | IPR | Interrupt Pending Register | |
| 206Ch | IPRH | Interrupt Pending Register High | |
| 2070h | ICR | Interrupt Clear Register | |
| 2074h | ICRH | Interrupt Clear Register High | |
| 2078h | IEVAL | Interrupt Evaluate Register | |
| 2080h | QER | QDMA Event Register | |
| 2084h | QEER | QDMA Event Enable Register | |
| 2088h | QEECR | QDMA Event Enable Clear Register | |
| 208Ch | QEESR | QDMA Event Enable Set Register | |
| 2090h | QSER | QDMA Secondary Event Register | |
| 2094h | QSECR | QDMA Secondary Event Clear Register | |
| 2200h-2294h | - | Shadow Region 1 Channel Registers | |
| 2400h-2494h | - | Shadow Region 2 Channel Registers | |
| ... | | ... | |
| 2E00h-2E94h | - | Shadow Channel Registers for MP Space 7 | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.1.1 Global Registers

#### *8.5.1.1.1 Peripheral Identification Register (PID)*

The peripheral identification register (PID) uniquely identifies the EDMA3CC and the specific revision of the EDMA3CC.

The PID is shown in Figure 8-40 and described in Table 8-20.

**Figure 8-40. Peripheral ID Register (PID)**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| PID | | PID | |
| R-4001h | | R-1B00h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-20. Peripheral ID Register (PID) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | PID | 0-FFFF FFFFh | Peripheral identifier uniquely identifies the EDMA3CC and the specific revision of the EDMA3CC. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.1.2 EDMA3CC Configuration Register (CCCFG)

The EDMA3CC configuration register (CCCFG) provides the features/resources for the EDMA3CC in a particular device.

The CCCFG is shown in Figure 8-41 and described in Table 8-21.

**Figure 8-41. EDMA3CC Configuration Register (CCCFG)**

| 31 | 26 | 25 | 24 |
|---|---|---|---|
| Reserved | | MP_EXIST | CHMAP_EXIST |
| R-x | | R-1 | R-1 |

| 23 | 22 | 21 | 20 | 19 | 18 | 16 |
|---|---|---|---|---|---|---|
| Reserved | | NUM_REGN | | Reserved | NUM_EVQUE | |
| R-0 | | R-3h | | R-x | R-3h | |

| 15 | 14 | 12 | 11 | 10 | 8 |
|---|---|---|---|---|---|
| Reserved | NUM_PAENTRY | | Reserved | NUM_INTCH | |
| R-x | R-5h | | R-x | R-4h | |

| 7 | 6 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|
| Reserved | NUM_QDMACH | | Reserved | NUM_DMACH | |
| R-x | R-4h | | R-x | R-5h | |

LEGEND: R = Read only; -*n* = value after reset; -x = value is indeterminate after reset

**Table 8-21. EDMA3CC Configuration Register (CCCFG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-26 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 25 | MP_EXIST | | Memory protection existence. |
| | | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 1 | Memory protection logic included. |
| 24 | CHMAP_EXIST | | Channel mapping existence. |
| | | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 1 | Channel mapping logic included. |
| 23-22 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 21-20 | NUM_REGN | 0-3h | Number of MP and shadow regions. |
| | | 0-2h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 3h | 8 regions. |
| 19 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 18-16 | NUM_EVQUE | 0-7h | Number of queues/number of TCs. |
| | | 0-2h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 3h | 4 EDMA3TCs/Event Queues. |
| | | 4h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 15 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-21. EDMA3CC Configuration Register (CCCFG) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 14-12 | NUM_PAENTRY | 0-7h | Number of PaRAM sets. |
| | | 0-3h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 5h | 512 PaRAM sets. |
| | | 5h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 11 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 10-8 | NUM_INTCH | 0-7h | Number of interrupt channels. |
| | | 0-3h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 4h | 64 interrupt channels. |
| | | 5h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 6-4 | NUM_QDMACH | 0-7h | Number of QDMA channels. |
| | | 0-1h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 4h | 8 QDMA channels. |
| | | 3h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 2-0 | NUM_DMACH | 0-7h | Number of DMA channels. |
| | | 0-4h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 5h | 64 DMA channels. |
| | | 6h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.1.3 DMA Channel Map *n* Registers (DCHMAP*n*)

The DMA channel map *n* register (DCHMAP*n*) is shown in Figure 8-42 and described in Table 8-22.

**Figure 8-42. DMA Channel Map *n* Registers (DCHMAP*n*)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 5 | 4 | 0 |
|---|---|---|---|---|---|
| Reserved | | PAENTRY | | Reserved | |
| R-0 | | R/W-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-22. DMA Channel Map *n* Registers (DCHMAP*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-14 | Reserved | 0 | Reserved |
| 13-5 | PAENTRY | 0-1FFh | Points to the PaRAM set number for DMA channel *n*. |
| 4-0 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.1.4 QDMA Channel Map *n* Registers (QCHMAP*n*)

Each QDMA channel in EDMA3CC can be associated with any PaRAM set available on the device. Furthermore, the specific trigger word (0-7) of the PaRAM set can be programmed. The PaRAM set association and trigger word for every QDMA channel register is configurable using the QDMA channel map *n* register (QCHMAP*n*).

The QCHMAP*n* is shown in Figure 8-43 and described in Table 8-23.

**NOTE:** At reset the QDMA channel map registers for all QDMA channels point to PaRAM set 0. If an application makes use of both a DMA channel that points to PaRAM set 0 and any QDMA channels, ensure that QCHMAP*n* is programmed appropriately to point to a different PaRAM entry.

**Figure 8-43. QDMA Channel Map *n* Registers (QCHMAP*n*)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 5 | 4 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PAENTRY | | TRWORD | | Reserved | |
| R-0 | | R/W-0 | | R/W-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-23. QDMA Channel Map *n* Registers (QCHMAP*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-14 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 13-5 | PAENTRY | 0-1FFh | PAENTRY points to the PaRAM set number for QDMA channel *n*. |
| 4-2 | TRWORD | 0-7h | Points to the specific trigger word of the PaRAM set defined by PAENTRY. A write to the trigger word results in a QDMA event being recognized. |
| 1-0 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.1.5 DMA Channel Queue *n* Number Registers (DMAQNUM*n*)

The DMA channel queue number register (DMAQNUM*n*) allows programmability of each of the 64 DMA channels in the EDMA3CC to submit its associated synchronization event to any event queue in the EDMA3CC. At reset, all channels point to event queue 0.

The DMAQNUM*n* is shown in Figure 8-44 and described in Table 8-24. Table 8-25 shows the channels and their corresponding bits in DMAQNUM*n*.

**NOTE:** Because the event queues in EDMA3CC have a fixed association to the transfer controllers, that is, Q0 TRs are submitted to TC0, Q1 TRs are submitted to TC1, etc., by programming DMAQNUM*n* for a particular DMA channel *n* also dictates which transfer controller is utilized for the data movement (or which EDMA3TC receives the TR request).

**Figure 8-44. DMA Channel Queue *n* Number Registers (DMAQNUM*n*)**

| 31 | 30 28 | 27 | 26 24 | 23 | 22 20 | 19 | 18 16 |
|---|---|---|---|---|---|---|---|
| Rsvd | E*n* | Rsvd | E*n* | Rsvd | E*n* | Rsvd | E*n* |
| R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 |

| 15 | 14 12 | 11 | 10 8 | 7 | 6 4 | 3 | 2 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | E*n* | Rsvd | E*n* | Rsvd | E*n* | Rsvd | E*n* |
| R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-24. DMA Channel Queue *n* Number Registers (DMAQNUM*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | 0-7h | DMA queue number. Contains the event queue number to be used for the corresponding DMA channel. Programming DMAQNUM*n* for an event queue number to a value more then the number of queues available in the EDMA3CC results in undefined behavior. |
| | | 0 | Event *n* is queued on Q0. |
| | | 1h | Event *n* is queued on Q1. |
| | | 2h | Event *n* is queued on Q2. |
| | | 3h | Event *n* is queued on Q3. |
| | | 4h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |

**Table 8-25. Bits in DMAQNUM*n***

| | Channel Number (DMAQNUM*n*) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| E*n* bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0-2 | E0 | E8 | E16 | E24 | E32 | E40 | E48 | E56 |
| 4-6 | E1 | E9 | E17 | E25 | E33 | E41 | E49 | E57 |
| 8-10 | E2 | E10 | E18 | E26 | E34 | E42 | E50 | E58 |
| 12-14 | E3 | E11 | E19 | E27 | E35 | E43 | E51 | E59 |
| 16-18 | E4 | E12 | E20 | E28 | E36 | E44 | E52 | E60 |
| 20-22 | E5 | E13 | E21 | E29 | E37 | E45 | E53 | E61 |
| 24-26 | E6 | E14 | E22 | E30 | E38 | E46 | E54 | E62 |
| 28-30 | E7 | E15 | E23 | E31 | E39 | E47 | E55 | E63 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.1.6 QDMA Channel Queue Number Register (QDMAQNUM)

The QDMA channel queue number register (QDMAQNUM) is used to program all the QDMA channels in the EDMA3CC to submit the associated QDMA event to any of the event queues in the EDMA3CC.

The QDMAQNUM is shown in Figure 8-45 and described in Table 8-26.

**Figure 8-45. QDMA Channel Queue Number Register (QDMAQNUM)**

| 31 | 30 28 | 27 | 26 24 | 23 | 22 20 | 19 | 18 16 |
|---|---|---|---|---|---|---|---|
| Rsvd | E7 | Rsvd | E6 | Rsvd | E5 | Rsvd | E4 |
| R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 |

| 15 | 14 12 | 11 | 10 8 | 7 | 6 4 | 3 | 2 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | E3 | Rsvd | E2 | Rsvd | E1 | Rsvd | E0 |
| R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-26. QDMA Channel Queue Number Register (QDMAQNUM) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 14-0 | E*n* | 0-7h | QDMA queue number. Contains the event queue number to be used for the corresponding QDMA channel. |
| | | 0 | Event *n* is queued on Q0. |
| | | 1h | Event *n* is queued on Q1. |
| | | 2h | Event *n* is queued on Q2. |
| | | 3h | Event *n* is queued on Q3. |
| | | 4h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.1.2 Error Registers

The EDMA3CC contains a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

#### *8.5.1.2.1 Event Missed Registers (EMR/EMRH)*

For a particular DMA channel, if a second event is received prior to the first event getting cleared/serviced, the bit corresponding to that channel is set/asserted in the event missed registers (EMR/EMRH). All trigger types are treated individually, that is, manual triggered (ESR/ESRH), chain triggered (CER/CERH), and event triggered (ER/ERH) are all treated separately. The EMR/EMRH bits for a channel are also set if an event on that channel encounters a NULL entry (or a NULL TR is serviced). If any EMR/EMRH bit is set (and all errors, including bits in other error registers (QEMR, CCERR) were previously cleared), the EDMA3CC generates an error interrupt. For details on EDMA3CC error interrupt generation, see *Error Interrupts*.

The EMR is shown in Figure 8-46 and described in Table 8-27. The EMRH is shown in Figure 8-47 and described in Table 8-28.

**Figure 8-46. Event Missed Register (EMR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-27. Event Missed Register (EMR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Channel 0-31 event missed. E*n* is cleared by writing a 1 to the corresponding bit in the event missed clear register (EMCR). |
| | | 0 | No missed event. |
| | | 1 | Missed event occurred. |

**Figure 8-47. Event Missed Register High (EMRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-28. Event Missed Register High (EMRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Channel 32–63 event missed. E*n* is cleared by writing a 1 to the corresponding bit in the event missed clear register high (EMCRH). |
| | | 0 | No missed event. |
| | | 1 | Missed event occurred. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.2.2 Event Missed Clear Registers (EMCR/EMCRH)

Once a missed event is posted in the event missed registers (EMR/EMRH), the bit remains set and you need to clear the set bit(s). This is done by way of CPU writes to the event missed clear registers (EMCR/EMCRH). Writing a 1 to any of the bits clears the corresponding missed event (bit) in EMR/EMRH; writing a 0 has no effect.

The EMCR is shown in Figure 8-48 and described in Table 8-29. The EMCRH is shown in Figure 8-49 and described in Table 8-30.

**Figure 8-48. Event Missed Clear Register (EMCR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-29. Event Missed Clear Register (EMCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event missed 0-31 clear. All error bits must be cleared before additional error interrupts will be asserted by the EDMA3CC. |
| | | 0 | No effect. |
| | | 1 | Corresponding missed event bit in the event missed register (EMR) is cleared (E*n* = 0). |

**Figure 8-49. Event Missed Clear Register High (EMCRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-30. Event Missed Clear Register High (EMCRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event missed 32–63 clear. All error bits must be cleared before additional error interrupts will be asserted by the EDMA3CC. |
| | | 0 | No effect. |
| | | 1 | Corresponding missed event bit in the event missed register high (EMRH) is cleared (E*n* = 0). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.2.3 QDMA Event Missed Register (QEMR)

For a particular QDMA channel, if two QDMA events are detected without the first event getting cleared/serviced, the bit corresponding to that channel is set/asserted in the QDMA event missed register (QEMR). The QEMR bits for a channel are also set if a QDMA event on the channel encounters a NULL entry (or a NULL TR is serviced). If any QEMR bit is set (and all errors, including bits in other error registers (EMR/EMRH, CCERR) were previously cleared), the EDMA3CC generates an error interrupt. For details on EDMA3CC error interrupt generation, see *Error Interrupts*.

The QEMR is shown in Figure 8-50 and described in Table 8-31.

**Figure 8-50. QDMA Event Missed Register (QEMR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-31. QDMA Event Missed Register (QEMR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-0 | E*n* | | Channel 0-7 QDMA event missed. E*n* is cleared by writing a 1 to the corresponding bit in the QDMA event missed clear register (QEMCR). |
| | | 0 | No missed event. |
| | | 1 | Missed event occurred. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.2.4 QDMA Event Missed Clear Register (QEMCR)

Once a missed event is posted in the QDMA event missed registers (QEMR), the bit remains set and you need to clear the set bit(s). This is done by way of CPU writes to the QDMA event missed clear registers (QEMCR). Writing a 1 to any of the bits clears the corresponding missed event (bit) in QEMR; writing a 0 has no effect.

The QEMCR is shown in Figure 8-51 and described in Table 8-32.

**Figure 8-51. QDMA Event Missed Clear Register (QEMCR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | w-0 | w-0 | w-0 | w-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-32. QDMA Event Missed Clear Register (QEMCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-0 | E*n* | | QDMA event missed clear. All error bits must be cleared before additional error interrupts will be asserted by the EDMA3CC. |
| | | 0 | No effect. |
| | | 1 | Corresponding missed event bit in the QDMA event missed register (QEMR) is cleared (E*n* = 0). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.2.5 EDMA3CC Error Register (CCERR)

The EDMA3CC error register (CCERR) indicates whether or not at any instant of time the number of events queued up in any of the event queues exceeds or equals the threshold/watermark value that is set in the queue watermark threshold register (QWMTHRA). Additionally, CCERR also indicates if when the number of outstanding TRs that have been programmed to return transfer completion code (TRs which have the TCINTEN or TCCHEN bit in OPT set) to the EDMA3CC has exceeded the maximum allowed value of 63. If any bit in CCERR is set (and all errors, including bits in other error registers (EMR/EMRH, QEMR) were previously cleared), the EDMA3CC generates an error interrupt. For details on EDMA3CC error interrupt generation, see *Error Interrupts*. Once the error bits are set in CCERR, they can only be cleared by writing to the corresponding bits in the EDMA3CC error clear register (CCERRCLR).

The CCERR is shown in Figure 8-52 and described in Table 8-33.

**Figure 8-52. EDMA3CC Error Register (CCERR)**

| 31 | 17 | 16 |
|---|---|---|
| Reserved | | TCCERR |
| R-0 | | R-0 |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | QTHRXCD3 | QTHRXCD2 | QTHRXCD1 | QTHRXCD0 |
| R-0 | | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-33. EDMA3CC Error Register (CCERR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 16 | TCCERR | | Transfer completion code error. TCCERR is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR). |
| | | 0 | Total number of allowed TCCs outstanding has not been reached. |
| | | 1 | Total number of allowed TCCs has been reached. |
| 15-4 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3 | QTHRXCD3 | | Queue threshold error for queue 3. QTHRXCD3 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR). |
| | | 0 | Watermark/threshold has not been exceeded. |
| | | 1 | Watermark/threshold has been exceeded. |
| 2 | QTHRXCD2 | | Queue threshold error for queue 2. QTHRXCD2 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR). |
| | | 0 | Watermark/threshold has not been exceeded. |
| | | 1 | Watermark/threshold has been exceeded. |
| 1 | QTHRXCD1 | | Queue threshold error for queue 1 . QTHRXCD1 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR). |
| | | 0 | Watermark/threshold has not been exceeded. |
| | | 1 | Watermark/threshold has been exceeded. |
| 0 | QTHRXCD0 | | Queue threshold error for queue 0. QTHRXCD0 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR). |
| | | 0 | Watermark/threshold has not been exceeded. |
| | | 1 | Watermark/threshold has been exceeded. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.2.6 EDMA3CC Error Clear Register (CCERRCLR)

The EDMA3CC error clear register (CCERRCLR) is used to clear any error bits that are set in the EDMA3CC error register (CCERR). In addition, CCERRCLR also clears the values of some bit fields in the queue status registers (QSTAT*n*) associated with a particular event queue. Writing a 1 to any of the bits clears the corresponding bit in CCERR; writing a 0 has no effect.

The CCERRCLR is shown in Figure 8-53 and described in Table 8-34.

**Figure 8-53. EDMA3CC Error Clear Register (CCERRCLR)**

| 31 | 17 | 16 |
|---|---|---|
| Reserved | | TCCERR |
| W-0 | | W-0 |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | QTHRXCD3 | QTHRXCD2 | QTHRXCD1 | QTHRXCD0 |
| W-0 | | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-34. EDMA3CC Error Clear Register (CCERRCLR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 16 | TCCERR | | Transfer completion code error clear. |
| | | 0 | No effect. |
| | | 1 | Clears the TCCERR bit in the EDMA3CC error register (CCERR). |
| 15-4 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3 | QTHRXCD3 | | Queue threshold error clear for queue 3. |
| | | 0 | No effect. |
| | | 1 | Clears the QTHRXCD3 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 3 (QSTAT3). |
| 2 | QTHRXCD2 | | Queue threshold error clear for queue 2. |
| | | 0 | No effect. |
| | | 1 | Clears the QTHRXCD2 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 2 (QSTAT2). |
| 1 | QTHRXCD1 | | Queue threshold error clear for queue 1. |
| | | 0 | No effect. |
| | | 1 | Clears the QTHRXCD1 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 1 (QSTAT1). |
| 0 | QTHRXCD0 | | Queue threshold error clear for queue 0. |
| | | 0 | No effect. |
| | | 1 | Clears the QTHRXCD0 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 0 (QSTAT0). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.2.7 Error Evaluation Register (EEVAL)

The EDMA3CC error interrupt is asserted whenever an error bit is set in any of the error registers (EMR/EMRH, QEMR, and CCERR). For subsequent error bits that get set, the EDMA3CC error interrupt is reasserted only when transitioning from an “all the error bits cleared” to “at least one error bit is set”. Alternatively, a CPU write of 1 to the EVAL bit in the error evaluation register (EEVAL) results in reasserting the EDMA3CC error interrupt, if there are any outstanding error bits set due to subsequent error conditions. Writes of 0 have no effect.

The EEVAL is shown in Figure 8-54 and described in Table 8-35.

**Figure 8-54. Error Evaluation Register (EEVAL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | Rsvd | EVAL |
| R-0 | | R/W-0 | W-0 |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

**Table 8-35. Error Evaluation Register (EEVAL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 1 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 0 | EVAL | | Error interrupt evaluate. |
| | | 0 | No effect. |
| | | 1 | EDMA3CC error interrupt will be pulsed if any errors have not been cleared in any of the error registers (EMR/EMRH, QEMR, or CCERR). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.1.3 Region Access Enable Registers

The region access enable register group consists of the DMA access enable registers (DRAE*m* and DRAEH*m*) and the QDMA access enable registers (QRAE*m*). Where *m* is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

#### *8.5.1.3.1 DMA Region Access Enable for Region m (DRAEm)*

The DMA region access enable register for shadow region *m* (DRAE*m*/DRAEH*m*) is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all DMA registers in the shadow region *m* view of the DMA channel registers. See the EDMA3CC register memory map (Table 8-19) for a list of all the DMA channel and interrupt registers mapped in the shadow region view. Additionally, the DRAE*m*/DRAEH*m* configuration determines completion of which DMA channels will result in assertion of the shadow region *m* DMA completion interrupt (see *EDMA3 Interrupts*).

The DRAE*m* is shown in Figure 8-55 and described in Table 8-36. The DRAEH*m* is shown in Figure 8-56 and described in Table 8-36.

**Figure 8-55. DMA Region Access Enable Register for Region *m* (DRAE*m*)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Figure 8-56. DMA Region Access Enable High Register for Region *m* (DRAEH*m*)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 8-36. DMA Region Access Enable Registers for Region M (DRAE*m*/DRAEH*m*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | DMA region access enable for bit *n*/channel *n* in region *m*. |
| | | 0 | Accesses via region *m* address space to bit *n* in any DMA channel register are not allowed. Reads return 0 on bit *n* and writes do not modify the state of bit *n*. Enabled interrupt bits for bit *n* do not contribute to the generation of a transfer completion interrupt for shadow region *m*. |
| | | 1 | Accesses via region *m* address space to bit *n* in any DMA channel register are allowed. Reads return the value from bit *n* and writes modify the state of bit *n*. Enabled interrupt bits for bit *n* contribute to the generation of a transfer completion interrupt for shadow region *m*. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.3.2 QDMA Region Access Enable Registers (QRAE*m*)

The QDMA region access enable register for shadow region *m* (QRAE*m*) is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all QDMA registers in the shadow region *m* view of the QDMA registers. This includes all 4-bit QDMA registers.

The QRAE*m* is shown in Figure 8-57 and described in Table 8-37.

**Figure 8-57. QDMA Region Access Enable for Region *m* (QRAE*m*)**

| 31 | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | |
| R-0 | | | | | | | | |
| 15 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-37. QDMA Region Access Enable for Region M (QRAE*m*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-0 | E*n* | | QDMA region access enable for bit *n*/QDMA channel *n* in region *m*. |
| | | 0 | Accesses via region *m* address space to bit *n* in any QDMA channel register are not allowed. Reads return 0 on bit *n* and writes do not modify the state of bit *n*. |
| | | 1 | Accesses via region *m* address space to bit *n* in any QDMA channel register are allowed. Reads return the value from bit *n* and writes modify the state of bit *n*. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.1.4 Status/Debug Visibility Registers

The following set of registers provide visibility into the event queues and a TR life cycle. These are useful for system debug as they provide in-depth visibility for the events queued up in the event queue and also provide information on what parts of the EDMA3CC logic are active once the event has been received by the EDMA3CC.

#### *8.5.1.4.1 Event Queue Entry Registers (QxEy)*

The event queue entry registers (Q*x*E*y*) exist for all 16 queue entries (the maximum allowed queue entries) for all event queues in the EDMA3CC.

There are Q0E0 to Q0E15, Q1E0 to Q1E15, Q2E0 to Q2E15, and Q3E0 to Q3E15. Each register details the event number (ENUM) and the event type (ETYPE). For example, if the value in Q1E4 is read as 000 004Fh, this means the 4th entry in queue 1 is a manually-triggered event on DMA channel 15.

The Q*x*E*y* is shown in Figure 8-58 and described in Table 8-38.

**Figure 8-58. Event Queue Entry Registers (Q*x*E*y*)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|
| Reserved | | ETYPE | | ENUM | |
| R-0 | | R-x | | R-x | |

LEGEND: R = Read only; -*n* = value after reset; -x = value is indeterminate after reset

**Table 8-38. Event Queue Entry Registers (Q*x*E*y*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-6 | ETYPE | 0-3h | Event entry *y* in queue *x*. Specifies the specific event type for the given entry in the event queue. |
| | | 0 | Event triggered via ER. |
| | | 1h | Manual triggered via ESR. |
| | | 2h | Chain triggered via CER. |
| | | 3h | Auto-triggered via QER. |
| 5-0 | ENUM | 0-3Fh | Event entry *y* in queue *x*. Event number: |
| | | 0-3h | QDMA channel number (0 to 3). |
| | | 0-3Fh | DMA channel/event number (0 to 63). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.4.2 Queue Status Registers (QSTAT*n*)

The queue status registers (QSTAT*n*) is shown in Figure 8-59 and described in Table 8-39.

**Figure 8-59. Queue Status Register *n* (QSTAT*n*)**

| 31 | 25 | 24 | 23 | 21 | 20 | 16 |
|---|---|---|---|---|---|---|
| Reserved | | THRXCD | Reserved | | WM | |
| R-0 | | R-0 | R-0 | | R-0 | |

| 15 | 13 | 12 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | NUMVAL | | Reserved | | STRTPTR | |
| R-0 | | R-0 | | R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-39. Queue Status Register *n* (QSTAT*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-25 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 24 | THRXCD | | Threshold exceeded. THRXCD is cleared by writing a 1 to the corresponding QTHRXCD*n* bit in the EDMA3CC error clear register (CCERRCLR). |
| | | 0 | Threshold specified by the Q*n* bit in the queue watermark threshold A register (QWMTHRA) has not been exceeded. |
| | | 1 | Threshold specified by the Q*n* bit in the queue watermark threshold A register (QWMTHRA) has been exceeded. |
| 23-21 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 20-16 | WM | 0-10h | Watermark for maximum queue usage. Watermark tracks the most entries that have been in queue *n* since reset or since the last time that the watermark (WM) bit was cleared. WM is cleared by writing a 1 to the corresponding QTHRXCD*n* bit in the EDMA3CC error clear register (CCERRCLR). |
| | | 0-10h | Legal values are 0 (empty) to 10h (full). |
| | | 11h-1Fh | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 15-13 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 12-8 | NUMVAL | 0-10h | Number of valid entries in queue *n*. The total number of entries residing in the queue manager FIFO at a given instant. Always enabled. |
| | | 0-10h | Legal values are 0 (empty) to 10h (full). |
| | | 11h-1Fh | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-4 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3-0 | STRTPTR | 0-Fh | Start pointer. The offset to the head entry of queue *n*, in units of entries. Always enabled. Legal values are 0 (0th entry) to Fh (15th entry). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.4.3 Queue Watermark Threshold A Register (QWMTHRA)

The queue watermark threshold A register (QWMTHRA) is shown in Figure 8-60 and described in Table 8-40.

**Figure 8-60. Queue Watermark Threshold A Register (QWMTHRA)**

| 31 29 | 28 24 | 23 21 | 20 16 |
|---|---|---|---|
| Reserved | Q3 | Reserved | Q2 |
| R-0 | R/W-10 | R-0 | R/W-10 |

| 15 13 | 12 8 | 7 5 | 4 0 |
|---|---|---|---|
| Reserved | Q1 | Reserved | Q0 |
| R-0 | R/W-10h | R-0 | R/W-10h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 8-40. Queue Watermark Threshold A Register (QWMTHRA) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-29 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 28-24 | Q3 | 0-1Fh | Queue threshold for queue 3 value. The QTHRXCD3 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 3 (QSTAT3) are set when the number of events in queue 3 at an instant in time (visible via the NUMVAL bit in QSTAT3) equals or exceeds the value specified by Q3. |
| | | 0-10h | The default is 16 (maximum allowed). |
| | | 11h | Disables the threshold errors. |
| | | 12h-1Fh | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 23-21 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 20-16 | Q2 | 0-1Fh | Queue threshold for queue 2 value. The QTHRXCD2 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 2 (QSTAT2) are set when the number of events in queue 2 at an instant in time (visible via the NUMVAL bit in QSTAT2) equals or exceeds the value specified by Q2. |
| | | 0-10h | The default is 16 (maximum allowed). |
| | | 11h | Disables the threshold errors. |
| | | 12h-1Fh | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 15-13 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 12-8 | Q1 | | Queue threshold for queue 1 value. The QTHRXCD1 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 1 (QSTAT1) are set when the number of events in queue 1 at an instant in time (visible via the NUMVAL bit in QSTAT1) equals or exceeds the value specified by Q1. |
| | | 0-10h | The default is 16 (maximum allowed). |
| | | 11h | Disables the threshold errors. |
| | | 12h-1Fh | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-5 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 4-0 | Q0 | | Queue threshold for queue 0 value. The QTHRXCD0 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 0 (QSTAT0) are set when the number of events in queue 0 at an instant in time (visible via the NUMVAL bit in QSTAT0) equals or exceeds the value specified by Q0. |
| | | 0-10h | The default is 16 (maximum allowed). |
| | | 11h | Disables the threshold errors. |
| | | 12h-1Fh | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.4.4 EDMA3CC Status Register (CCSTAT)

The EDMA3CC status register (CCSTAT) has a number of status bits that reflect which parts of the EDMA3CC logic is active at any given instant of time. The CCSTAT is shown in Figure 8-61 and described in Table 8-41.

**Figure 8-61. EDMA3CC Status Register (CCSTAT)**

| 31 | | | | | | | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | Reserved | | | QUEACTV3 | QUEACTV2 | QUEACTV1 | QUEACTV0 |
| R-0 | R-0 | | | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | COMPACTV | | | | | |
| R-0 | | R-0 | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | ACTV | Reserved | TRACTV | QEVTACTV | EVTACTV |
| R-0 | | | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-41. EDMA3CC Status Register (CCSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 19 | QUEACTV3 | | Queue 3 active. |
| | | 0 | No events are queued in queue 3. |
| | | 1 | At least one TR is queued in queue 3. |
| 18 | QUEACTV2 | | Queue 2 active. |
| | | 0 | No events are queued in queue 2. |
| | | 1 | At least one TR is queued in queue 2. |
| 17 | QUEACTV1 | | Queue 1 active. |
| | | 0 | No events are queued in queue 1. |
| | | 1 | At least one TR is queued in queue 1. |
| 16 | QUEACTV0 | | Queue 0 active. |
| | | 0 | No events are queued in queue 0. |
| | | 1 | At least one TR is queued in queue 0. |
| 15-14 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 13-8 | COMPACTV | 0-3Fh | Completion request active. The COMPACTV field reflects the count for the number of completion requests submitted to the transfer controllers. This count increments every time a TR is submitted and is programmed to report completion (the TCINTEN or TCCCHEN bits in OPT in the parameter entry associated with the TR are set). The counter decrements for every valid TCC received back from the transfer controllers. If at any time the count reaches a value of 63, the EDMA3CC will not service any new TRs until the count is less then 63 (or return a transfer completion code from a transfer controller, which would decrement the count). |
| | | 0 | No completion requests outstanding. |
| | | 1h-3Fh | Total of 1 completion request to 63 completion requests are outstanding. |
| 7-5 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-41. EDMA3CC Status Register (CCSTAT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 4 | ACTV | | Channel controller active. Channel controller active is a logical-OR of each of the *ACTV bits. The ACTV bit remains high through the life of a TR. |
| | | 0 | Channel is idle.. |
| | | 1 | Channel is busy. |
| 3 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 2 | TRACTV | | Transfer request active. |
| | | 0 | Transfer request processing/submission logic is inactive. |
| | | 1 | Transfer request processing/submission logic is active. |
| 1 | QEVTACTV | | QDMA event active. |
| | | 0 | No enabled QDMA events are active within the EDMA3CC. |
| | | 1 | At least one enabled QDMA event (QER) is active within the EDMA3CC. |
| 0 | EVTACTV | | DMA event active. |
| | | 0 | No enabled DMA events are active within the EDMA3CC. |
| | | 1 | At least one enabled DMA event (ER and EER, ESR, CER) is active within the EDMA3CC. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.1.5 Memory Protection Address Space

#### *8.5.1.5.1 Memory Protection Fault Address Register (MPFAR)*

The memory protection fault address register (MPFAR) is shown in Figure 8-62 and described in Table 8-42. A CPU write of 1 to the MPFCLR bit in the memory protection fault command register (MPFCR) causes any error conditions stored in MPFAR to be cleared.

**Figure 8-62. Memory Protection Fault Address Register (MPFAR)**

| 31 | 16 |
|---|---|
| FADDR | |
| R-0 | |

| 15 | 0 |
|---|---|
| FADDR | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-42. Memory Protection Fault Address Register (MPFAR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | FADDR | 0-FFFF FFFFh | Fault address. This 32-bit read-only status register contains the fault address when a memory protection violation is detected. This register can only be cleared via the memory protection fault command register (MPFCR). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.5.2 Memory Protection Fault Status Register (MPFSR)

The memory protection fault status register (MPFSR) is shown in Figure 8-63 and described in Table 8-43. A CPU write of 1 to the MPFCLR bit in the memory protection fault command register (MPFCR) causes any error conditions stored in MPFSR to be cleared.

**Figure 8-63. Memory Protection Fault Status Register (MPFSR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 13 | 12 9 | 8 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | FID | Reserved | SRE | SWE | SXE | URE | UWE | UXE |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-43. Memory Protection Fault Status Register (MPFSR) Field Descriptions**

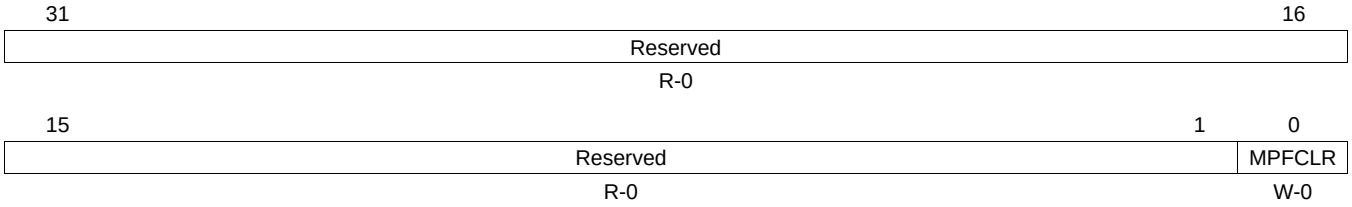
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-13 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 12-9 | FID | 0-Fh | Faulted identification. FID contains valid information if any of the MP error bits (UXE, UWE, URE, SXE, SWE, SRE) are nonzero (that is, if an error has been detected.) The FID field contains the privilege ID for the specific request/requestor that resulted in an MP error. |
| 8-6 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 5 | SRE | | Supervisor read error. |
| | | 0 | No error detected. |
| | | 1 | Supervisor level task attempted to read from a MP page without SR permissions. |
| 4 | SWE | | Supervisor write error. |
| | | 0 | No error detected. |
| | | 1 | Supervisor level task attempted to write to a MP page without SW permissions. |
| 3 | SXE | | Supervisor execute error. |
| | | 0 | No error detected. |
| | | 1 | Supervisor level task attempted to execute from a MP page without SX permissions. |
| 2 | URE | | User read error. |
| | | 0 | No error detected. |
| | | 1 | User level task attempted to read from a MP page without UR permissions. |
| 1 | UWE | | User write error. |
| | | 0 | No error detected. |
| | | 1 | User level task attempted to write to a MP page without UW permissions. |
| 0 | UXE | | User execute error. |
| | | 0 | No error detected. |
| | | 1 | User level task attempted to execute from a MP page without UX permissions. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.5.3 Memory Protection Fault Command Register (MPFCR)

The memory protection fault command register (MPFCR) is shown in Figure 8-64 and described in Table 8-44.

**Figure 8-64. Memory Protection Fault Command Register (MPFCR)**




LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 8-44. Memory Protection Fault Command Register (MPFCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 0 | MPFCLR | | Fault clear register. |
| | | 0 | CPU write of 0 has no effect. |
| | | 1 | CPU write of 1 to the MPFCLR bit causes any error conditions stored in the memory protection fault address register (MPFAR) and the memory protection fault status register (MPFSR) to be cleared. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.5.4 Memory Protection Page Attribute Register (MPPA*n*)

The memory protection page attribute register (MPPA*n*) is shown in Figure 8-65 and described in Table 8-45.

**Figure 8-65. Memory Protection Page Attribute Register (MPPA*n*)**

| 31 | | | | | | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AID5 | AID4 | AID3 | AID2 | AID1 | AID0 | EXT | Rsvd | Reserved | | SR | SW | SX | UR | UW | UX |
| RW-1 | RW-1 | RW-1 | RW-1 | RW-1 | RW-1 | RW-1 | R-0 | RW-1 | | RW-1 | RW-1 | RW-0 | RW-1 | RW-1 | RW-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-45. Memory Protection Page Attribute Register (MPPA*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 15-10 | AID*m* | | Allowed ID 'N' |
| | | 0 | Requests with Privilege ID == N are not allowed to region M, regardless of permission settings (UW, UR, SW, SR). |
| | | 1 | Requests with Privilege ID == N are permitted, if access type is allowed as defined by permission settings (UW, UR, SW, SR). |
| 9 | EXT | | External Allowed ID. |
| | | 0 | Requests with Privilege ID >= 6 are not allowed to region M, regardless of permission settings (UW, UR, SW, SR). |
| | | 1 | Requests with Privilege ID >= 6 are permitted, if access type is allowed as defined by permission settings (UW, UR, SW, SR). |
| 8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-6 | Reserved | 1 | Reserved. Always write 1 to this bit. |
| 5 | SR | | Supervisor read permission. |
| | | 0 | Supervisor read accesses are not allowed from region M. |
| | | 1 | Supervisor write accesses are allowed from region M addresses. |
| 4 | SW | | Supervisor write permission. |
| | | 0 | Supervisor write accesses are not allowed to region M. |
| | | 1 | Supervisor write accesses are allowed to region N addresses. |
| 3 | SX | | Supervisor execute permission. |
| | | 0 | Supervisor execute accesses are not allowed from region M. |
| | | 1 | Supervisor execute accesses are allowed from region M addresses. |
| 2 | UR | | User read permission. |
| | | 0 | User read accesses are not allowed from region M. |
| | | 1 | User read accesses are allowed from region N addresses. |
| 1 | UW | | User write permission. |
| | | 0 | User write accesses are not allowed to region M. |
| | | 1 | User write accesses are allowed to region M addresses. |
| 0 | UX | | User execute permission. |
| | | 0 | User execute accesses are not allowed from region M. |
| | | 1 | User execute accesses are allowed from region M addresses. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.1.6 DMA Channel Registers

The following sets of registers pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUM*n*) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format *reg_name* that corresponds to DMA channels 0 through 31 and *reg_name_High* that corresponds to DMA channels 32 through 64.

For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register.

The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAE*m*/DRAEH*m*). The registers are described in Section 8.5.1.3.1 and the details for shadow region/global region usage is explained in *EDMA3 Channel Controller Regions*.

#### *8.5.1.6.1 Event Registers (ER, ERH)*

All external events are captured in the event register (ER/ERH). The events are latched even when the events are not enabled. If the event bit corresponding to the latched event is enabled (EER.E*n*/EERH.E*n* = 1), then the event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. The event register bits are automatically cleared (ER.E*n*/ERH.E*n* = 0) once the corresponding events are prioritized and serviced. If ER.E*n*/ERH.E*n* are already set and another event is received on the same channel/event, then the corresponding event is latched in the event miss register (EMR.E*n*/EMRH.E*n*), provided that the event was enabled (EER.E*n*/EERH.E*n* = 1).

Event *n* can be cleared by the CPU writing a 1 to corresponding event bit in the event clear register (ECR/ECRH). The setting of an event is a higher priority relative to clear operations (via hardware or software). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues.

The *Debug List* table provides the type of synchronization events and the EDMA3CC channels associated to each of these external events.


Copyright © 2011–2016, Texas Instruments Incorporated

The ER is shown in Figure 8-66 and described in Table 8-46. The ERH is shown in Figure 8-67 and described in Table 8-47.

**Figure 8-66. Event Register (ER)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | 19 | E18 | E17 | E16 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-46. Event Register (ER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event 0-31. Events 0-31 are captured by the EDMA3CC and are latched into ER. The events are set (E*n* = 1) even when events are disabled (E*n* = 0 in the event enable register, EER). |
| | | 0 | EDMA3CC event is not asserted. |
| | | 1 | EDMA3CC event is asserted. Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC. |

**Figure 8-67. Event Register High (ERH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-47. Event Register High (ERH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event 32-63. Events 32-63 are captured by the EDMA3CC and are latched into ERH. The events are set (E*n* = 1) even when events are disabled (E*n* = 0 in the event enable register high, EERH). |
| | | 0 | EDMA3CC event is not asserted. |
| | | 1 | EDMA3CC event is asserted. Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.6.2 Event Clear Registers (ECR, ECRH)

Once an event has been posted in the event registers (ER/ERH), the event is cleared in two ways. If the event is enabled in the event enable register (EER/EERH) and the EDMA3CC submits a transfer request for the event to the EDMA3TC, it clears the corresponding event bit in the event register. If the event is disabled in the event enable register (EER/EERH), the CPU can clear the event by way of the event clear registers (ECR/ECRH).

Writing a 1 to any of the bits clears the corresponding event; writing a 0 has no effect. Once an event bit is set in the event register, it remains set until EDMA3CC submits a transfer request for that event or the CPU clears the event by setting the corresponding bit in ECR/ECRH.

The ECR is shown in Figure 8-68 and described in Table 8-48. The ECRH is shown in Figure 8-69 and described in Table 8-49.

**Figure 8-68. Event Clear Register (ECR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-48. Event Clear Register (ECR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event clear for event 0-31. Any of the event bits in ECR is set to clear the event (E*n*) in the event register (ER). A write of 0 has no effect. |
| | | 0 | No effect. |
| | | 1 | EDMA3CC event is cleared in the event register (ER). |

**Figure 8-69. Event Clear Register High (ECRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-49. Event Clear Register High (ECRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event clear for event 32-63. Any of the event bits in ECRH are set to clear the event (E*n*) in the event register high (ERH). A write of 0 has no effect. |
| | | 0 | No effect. |
| | | 1 | EDMA3CC event is cleared in the event register high (ERH). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.6.3 Event Set Registers (ESR, ESRH)

The event set registers (ESR/ESRH) allow the CPU (EDMA3 programmers) to manually set events to initiate DMA transfer requests. CPU writes of 1 to any event set register (E*n*) bits set the corresponding bits in the registers. The set event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. Writing a 0 has no effect.

The event set registers operate independent of the event registers (ER/ERH), and a write of 1 is always considered a valid event regardless of whether the event is enabled (the corresponding event bits are set or cleared in EER.E*n*/EERH.E*n*).

Once the event is set in the event set registers, it cannot be cleared by CPU writes, in other words, the event clear registers (ECR/ECRH) have no effect on the state of ESR/ESRH. The bits will only be cleared once the transfer request corresponding to the event has been submitted to the transfer controller. The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues.

Manually-triggered transfers via writes to ESR/ESRH allow the CPU to submit DMA requests in the system, these are relevant for memory-to-memory transfer scenarios. If the ESR.E*n*/ESRH.E*n* bit is already set and another CPU write of 1 is attempted to the same bit, then the corresponding event is latched in the event missed registers (EMR.E*n*/EMRH.E*n* = 1).

The ESR is shown in Figure 8-70 and described in Table 8-50. The ESRH is shown in Figure 8-71 and described in Table 8-51.

**Figure 8-70. Event Set Register (ESR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 8-50. Event Set Register (ESR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event set for event 0-31. |
| | | 0 | No effect. |
| | | 1 | Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-71. Event Set Register High (ESRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 | RW-0 |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 8-51. Event Set Register High (ESRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event set for event 32-63. |
| | | 0 | No effect. |
| | | 1 | Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.6.4 Chained Event Registers (CER, CERH)

When the OPTIONS parameter for a PaRAM entry is programmed to returned a chained completion code (ITCCHEN = 1 and/or TCCHEN = 1), then the value dictated by the TCC[5:0] (also programmed in OPT) forces the corresponding event bit to be set in the chained event registers (CER/CERH). The set chained event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. This results in a chained-triggered transfer.

The chained event registers do not have any enables. The generation of a chained event is essentially enabled by the PaRAM entry that has been configured for intermediate and/or final chaining on transfer completion. The E*n* bit is set (regardless of the state of EER.E*n*/EERH.E*n*) when a chained completion code is returned from one of the transfer controllers or is generated by the EDMA3CC via the early completion path. The bits in the chained event register are cleared when the corresponding events are prioritized and serviced.

If the E*n* bit is already set and another chaining completion code is return for the same event, then the corresponding event is latched in the event missed registers (EMR.E*n*/EMRH.E*n* = 1). The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues.

The CER is shown in Figure 8-72 and described in Table 8-52. The CERH is shown in Figure 8-73 and described in Table 8-53.

**Figure 8-72. Chained Event Register (CER)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-52. Chained Event Register (CER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Chained event for event 0-31. |
| | | 0 | No effect. |
| | | 1 | Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 8-73. Chained Event Register High (CERH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-53. Chained Event Register High (CERH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Chained event set for event 32-63. |
| | | 0 | No effect. |
| | | 1 | Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.6.5 Event Enable Registers (EER, EERH)

The EDMA3CC provides the option of selectively enabling/disabling each event in the event registers (ER/ERH) by using the event enable registers (EER/EERH). If an event bit in EER/EERH is set (using the event enable set registers, EESR/EESRH), it will enable that corresponding event. Alternatively, if an event bit in EER/EERH is cleared (using the event enable clear registers, EECR/EECRH), it will disable the corresponding event.

The event registers latch all events that are captured by EDMA3CC, even if the events are disabled (although EDMA3CC does not process it). Enabling an event with a pending event already set in the event registers enables the EDMA3CC to process the already set event like any other new event. The EER/EERH settings do not have any effect on chained events (CER.E*n*/CERH.E*n* = 1) and manually set events (ESR.E*n*/ESRH.E*n* = 1).

The EER is shown in Figure 8-74 and described in Table 8-54. Th EERH is shown in Figure 8-75 and described in Table 8-55.

**Figure 8-74. Event Enable Register (EER)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-54. Event Enable Register (EER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event enable for events 0-31. |
| | | 0 | Event is not enabled. An external event latched in the event register (ER) is not evaluated by the EDMA3CC. |
| | | 1 | Event is enabled. An external event latched in the event register (ER) is evaluated by the EDMA3CC. |

**Figure 8-75. Event Enable Register High (EERH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-55. Event Enable Register High (EERH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event enable for events 32-63. |
| | | 0 | Event is not enabled. An external event latched in the event register high (ERH) is not evaluated by the EDMA3CC. |
| | | 1 | Event is enabled. An external event latched in the event register high (ERH) is evaluated by the EDMA3CC. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.6.6 Event Enable Clear Register (EECR, EECRH)

The event enable registers (EER/EERH) cannot be modified by directly writing to them. The intent is to ease the software burden for the case where multiple tasks are attempting to simultaneously modify these registers. The event enable clear registers (EECR/EECRH) are used to disable events. Writes of 1 to the bits in EECR/EECRH clear the corresponding event bits in EER/EERH; writes of 0 have no effect.

The EECR is shown in Figure 8-76 and described in Table 8-56. The EECRH is shown in Figure 8-77 and described in Table 8-57.

**Figure 8-76. Event Enable Clear Register (EECR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-56. Event Enable Clear Register (EECR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event enable clear for events 0-31. |
| | | 0 | No effect. |
| | | 1 | Event is disabled. Corresponding bit in the event enable register (EER) is cleared (E*n* = 0). |

**Figure 8-77. Event Enable Clear Register High (EECRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-57. Event Enable Clear Register High (EECRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event enable clear for events 32-63. |
| | | 0 | No effect. |
| | | 1 | Event is disabled. Corresponding bit in the event enable register high (EERH) is cleared (E*n* = 0). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.6.7 Event Enable Set Registers (EESR, EESRH)

The event enable registers (EER/EERH) cannot be modified by directly writing to them. The intent is to ease the software burden for the case where multiple tasks are attempting to simultaneously modify these registers. The event enable set registers (EESR/EESRH) are used to enable events. Writes of 1 to the bits in EESR/EESRH set the corresponding event bits in EER/EERH; writes of 0 have no effect.

The EESR is shown in Figure 8-78 and described in Table 8-58. The EESRH is shown in Figure 8-79 and described in Table 8-59.

**Figure 8-78. Event Enable Set Register (EESR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-58. Event Enable Set Register (EESR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event enable set for events 0-31. |
| | | 0 | No effect. |
| | | 1 | Event is enabled. Corresponding bit in the event enable register (EER) is set (E*n* = 1). |

**Figure 8-79. Event Enable Set Register High (EESRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-59. Event Enable Set Register High (EESRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Event enable set for events 32-63. |
| | | 0 | No effect. |
| | | 1 | Event is enabled. Corresponding bit in the event enable register high (EERH) is set (E*n* = 1). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.6.8 Secondary Event Registers (SER, SERH)

The secondary event registers (SER/SERH) provide information on the state of a DMA channel or event (0 through 63). If the EDMA3CC receives a TR synchronization due to a manual-trigger, event-trigger, or chained-trigger source (ESR.E*n*/ESRH.E*n* = 1, ER.E*n*/ERH.E*n* = 1, or CER.E*n*/CERH.E*n* = 1), which results in the setting of a corresponding event bit in SER/SERH (SER.E*n*/SERH.E*n* = 1), it implies that the corresponding DMA event is in the queue.

Once a bit corresponding to an event is set in SER/SERH, the EDMA3CC does not prioritize additional events on the same DMA channel. Depending on the condition that lead to the setting of the SER bits, either the EDMA3CC hardware or the software (using SECR/SECRH) needs to clear the SER/SERH bits for the EDMA3CC to evaluate subsequent events (subsequent transfers) on the same channel. For additional conditions that can cause the secondary event registers to be set, see *EDMA Overview*.

The SER is shown in Figure 8-80 and described in Table 8-60. The SERH is shown in Figure 8-81 and described in Table 8-61.

**Figure 8-80. Secondary Event Register (SER)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-60. Secondary Event Register (SER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Secondary event register. The secondary event register is used along with the event register (ER) to provide information on the state of an event. |
| | | 0 | Event is not currently stored in the event queue. |
| | | 1 | Event is currently stored in the event queue. Event arbiter will not prioritize additional events. |

**Figure 8-81. Secondary Event Register High (SERH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-61. Secondary Event Register High (SERH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Secondary event register. The secondary event register is used along with the event register high (ERH) to provide information on the state of an event. |
| | | 0 | Event is not currently stored in the event queue. |
| | | 1 | Event is currently stored in the event queue. Event submission/prioritization logic will not prioritize additional events. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.6.9 Secondary Event Clear Registers (SECR, SECRH)

The secondary event clear registers (SECR/SECRH) clear the status of the secondary event registers (SER/SERH). CPU writes of 1 clear the corresponding set bits in SER/SERH. Writes of 0 have no effect.

The SECR is shown in Figure 8-82 and described in Table 8-62. The SECRH is shown in Figure 8-83 and described in Table 8-63.

**Figure 8-82. Secondary Event Clear Register (SECR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E31 | E30 | E29 | E28 | E27 | E26 | E25 | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E15 | E14 | E13 | E12 | E11 | E10 | E9 | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-62. Secondary Event Clear Register (SECR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Secondary event clear register. |
| | | 0 | No effect. |
| | | 1 | Corresponding bit in the secondary event register (SER) is cleared (E*n* = 0). |

**Figure 8-83. Secondary Event Clear Register High (SECRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E63 | E62 | E61 | E60 | E59 | E58 | E57 | E56 | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-63. Secondary Event Clear Register High (SECRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | E*n* | | Secondary event clear register. |
| | | 0 | No effect. |
| | | 1 | Corresponding bit in the secondary event registers high (SERH) is cleared (E*n* = 0). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.1.7 Interrupt Registers

All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. The following set of registers is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see *EDMA3 Interrupts*.

#### *8.5.1.7.1 Interrupt Enable Registers (IER, IERH)*

Interrupt enable registers (IER/IERH) are used to enable/disable the transfer completion interrupt generation by the EDMA3CC for all DMA/QDMA channels. The IER/IERH cannot be written to directly. To set any interrupt bit in IER/IERH, a 1 must be written to the corresponding interrupt bit in the interrupt enable set registers (IESR/IESRH). Similarly, to clear any interrupt bit in IER/IERH, a 1 must be written to the corresponding interrupt bit in the interrupt enable clear registers (IECR/IECRH).

The IER is shown in Figure 8-84 and described in Table 8-64. The IERH is shown in Figure 8-85 and described in Table 8-65.

**Figure 8-84. Interrupt Enable Register (IER)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I31 | I30 | I29 | I28 | I27I | I26 | I25 | I24 | I23 | I22 | I21 | I20 | I19 | I18 | I17 | I16 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8 | I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-64. Interrupt Enable Register (IER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | I*n* | | Interrupt enable for channels 0-31. |
| | | 0 | Interrupt is not enabled. |
| | | 1 | Interrupt is enabled. |

**Figure 8-85. Interrupt Enable Register High (IERH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I63 | I62 | I61 | I60 | I59 | I58 | I57 | I56 | I55 | I54 | I53 | I52 | I51 | I50 | I49 | I48 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I47 | I46 | I45 | I44 | I43 | I42 | I41 | I40 | I39 | I38 | I37 | I36 | I35 | I34 | I33 | I32 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-65. Interrupt Enable Register High (IERH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | I*n* | | Interrupt enable for channels 32-63. |
| | | 0 | Interrupt is not enabled. |
| | | 1 | Interrupt is enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.7.2 Interrupt Enable Clear Register (IECR, IECRH)

The interrupt enable clear registers (IECR/IECRH) are used to clear interrupts. Writes of 1 to the bits in IECR/IECRH clear the corresponding interrupt bits in the interrupt enable registers (IER/IERH); writes of 0 have no effect.

The IECR is shown in Figure 8-86 and described in Table 8-66. The IECRH is shown in Figure 8-87 and described in Table 8-67.

**Figure 8-86. Interrupt Enable Clear Register (IECR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I31 | I30 | I29 | I28 | I27 | I26 | I25 | I24 | I23 | I22 | I21 | I20 | I19 | I18 | I17 | 16 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8 | I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-66. Interrupt Enable Clear Register (IECR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | I*n* | | Interrupt enable clear for channels 0-31. |
| | | 0 | No effect. |
| | | 1 | Corresponding bit in the interrupt enable register (IER) is cleared (I*n* = 0). |

**Figure 8-87. Interrupt Enable Clear Register High (IECRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I63 | I62 | I61 | I60 | I59 | I58 | I57 | I56 | I55 | I54 | I53 | I52 | I51 | I50 | I49 | I48 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I47 | I46 | I45 | I44 | I43 | I42 | I41 | I40 | I39 | I38 | I37 | I36 | I35 | I34 | I33 | I32 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-67. Interrupt Enable Clear Register High (IECRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | I*n* | | Interrupt enable clear for channels 32-63. |
| | | 0 | No effect. |
| | | 1 | Corresponding bit in the interrupt enable register high (IERH) is cleared (I*n* = 0). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.7.3 Interrupt Enable Set Registers (IESR, IESRH)

The interrupt enable set registers (IESR/IESRH) are used to enable interrupts. Writes of 1 to the bits in IESR/IESRH set the corresponding interrupt bits in the interrupt enable registers (IER/IERH); writes of 0 have no effect.

The IESR is shown in Figure 8-88 and described in Table 8-68. The IESRH is shown in Figure 8-89 and described in Table 8-69.

**Figure 8-88. Interrupt Enable Set Register (IESR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I31 | I30 | I29 | I28 | I27 | I26 | I25 | I24 | I23 | I22 | I21 | I20 | I19 | II8 | I17 | I16 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8 | I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-68. Interrupt Enable Set Register (IESR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | I*n* | | Interrupt enable set for channels 0-31. |
| | | 0 | No effect. |
| | | 1 | Corresponding bit in the interrupt enable register (IER) is set (I*n* = 1). |

**Figure 8-89. Interrupt Enable Set Register High (IESRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I63 | I62 | I61 | I60 | I59 | I58 | I57 | I56 | I55 | I54 | I53 | I52 | I51 | I50 | I49 | I48 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I47 | I46 | I45 | I44 | I43 | I42 | I41 | I40 | I39 | I38 | I37 | I36 | I35 | I34 | I33 | I32 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-69. Interrupt Enable Set Register High (IESRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | I*n* | | Interrupt enable clear for channels 32-63. |
| | | 0 | No effect. |
| | | 1 | Corresponding bit in the interrupt enable register high (IERH) is set (I*n* = 1). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.7.4 Interrupt Pending Register (IPR, IPRH)

If the TCINTEN and/or ITCINTEN bit in the channel option parameter (OPT) is set in the PaRAM entry associated with the channel (DMA or QDMA), then the EDMA3TC (for normal completion) or the EDMA3CC (for early completion) returns a completion code on transfer or intermediate transfer completion. The value of the returned completion code is equal to the TCC bit in OPT for the PaRAM entry associated with the channel.

When an interrupt transfer completion code with TCC = *n* is detected by the EDMA3CC, then the corresponding bit is set in the interrupt pending register (IPR.I*n*, if *n* = 0 to 31; IPRH.I*n,* if *n* = 32 to 63). Note that once a bit is set in the interrupt pending registers, it remains set; it is your responsibility to clear these bits. The bits set in IPR/IPRH are cleared by writing a 1 to the corresponding bits in the interrupt clear registers (ICR/ICRH).

The IPR is shown in Figure 8-90 and described in Table 8-70. The IPRH is shown in Figure 8-91 and described in Table 8-71.

**Figure 8-90. Interrupt Pending Register (IPR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I31 | I30 | I29 | I28 | I27 | I26 | I25 | I24 | I23 | I22 | I21 | I20 | I19 | I18 | I17 | I16 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8 | I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-70. Interrupt Pending Register (IPR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | I*n* | | Interrupt pending for TCC = 0-31. |
| | | 0 | Interrupt transfer completion code is not detected or was cleared. |
| | | 1 | Interrupt transfer completion code is detected (I*n* = 1, *n* = EDMA3TC[5:0]). |

**Figure 8-91. Interrupt Pending Register High (IPRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I63 | I62 | I61 | I60 | I59 | I58 | I57 | I56 | I55 | I54 | I53 | I52 | I51 | I50 | I49 | I48 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I47 | I46 | I45 | I44 | I43 | I42 | I41 | I40 | I39 | I38 | I37 | I36 | I35 | I34 | I33 | I32 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-71. Interrupt Pending Register High (IPRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | I*n* | | Interrupt pending for TCC = 32-63. |
| | | 0 | Interrupt transfer completion code is not detected or was cleared. |
| | | 1 | Interrupt transfer completion code is detected (I*n* = 1, *n* = EDMA3TC[5:0]). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.7.5 Interrupt Clear Registers (ICR, ICRH)

The bits in the interrupt pending registers (IPR/IPRH) are cleared by writing a 1 to the corresponding bits in the interrupt clear registers(ICR/ICRH). Writes of 0 have no effect. All set bits in IPR/IPRH must be cleared to allow EDMA3CC to assert additional transfer completion interrupts.

The ICR is shown in Figure 8-92 and described in Table 8-72. The ICRH is shown in Figure 8-93 and described in Table 8-73.

**Figure 8-92. Interrupt Clear Register (ICR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I31 | I30 | I29 | I28 | I27 | I26 | I25 | I24 | I23 | I22 | I21 | I20 | I19 | I18 | I17 | I16 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8 | I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-72. Interrupt Clear Register (ICR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | I*n* | | Interrupt clear register for TCC = 0-31. |
| | | 0 | No effect. |
| | | 1 | Corresponding bit in the interrupt pending register (IPR) is cleared (I*n* = 0). |

**Figure 8-93. Interrupt Clear Register High (ICRH)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I63 | I62 | I61 | I60 | I59 | I58 | I57 | I56 | I55 | I54 | I53 | I52 | I51 | I50 | I49 | I48 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I47 | I46 | I45 | I44 | I43 | I42 | I41 | I40 | I39 | I38 | I37 | I36 | I35 | I34 | I33 | I32 |
| W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: W = Write only; -*n* = value after reset

**Table 8-73. Interrupt Clear Register High (ICRH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | I*n* | | Interrupt clear register for TCC = 32-63. |
| | | 0 | No effect. |
| | | 1 | Corresponding bit in the interrupt pending register high (IPRH) is cleared (I*n* = 0). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.1.7.6 Interrupt Evaluate Register (IEVAL)

The interrupt evaluate register (IEVAL) is the only register that physically exists in both the global region and the shadow regions. In other words, the read/write accessibility for the shadow region IEVAL is not affected by the DMA/QDMA region access registers (DRAE*m*/DRAEH*m*, QRAE*n*/QRAEH*n*). IEVAL is needed for robust ISR operations to ensure that interrupts are not missed by the CPU.

The IEVAL is shown in Figure 8-94 and described in Table 8-74.

**Figure 8-94. Interrupt Evaluate Register (IEVAL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | Rsvd | EVAL |
| R-0 | | W-0 | W-0 |

LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 8-74. Interrupt Evaluate Register (IEVAL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 1 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 0 | EVAL | | Interrupt evaluate. |
| | | 0 | No effect. |
| | | 1 | Causes EDMA3CC completion interrupt to be pulsed, if any enabled (IER*n*/IERH*n* = 1) interrupts are still pending (IPR*n*/IPRH*n* = 1).<br>The EDMA3CC completion interrupt that is pulsed depends on which IEVAL is being exercised. For example, writing to the EVAL bit in IEVAL pulses the global completion interrupt, but writing to the EVAL bit in IEVAL0 pulses the region 0 completion interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.1.8 QDMA Registers

The following sets of registers control the QDMA channels in the EDMA3CC. The QDMA channels (with the exception of the QDMA queue number register) consist of a set of registers, each of which have a bit location. Each bit position corresponds to a QDMA channel number. The QDMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write accessibility in the shadow region address region is controlled by the QDMA region access registers (QRAE*n*/QRAEH*n*). *EDMA3 Channel Controller Regions* details shadow region/global region usage.

#### *8.5.1.8.1 QDMA Event Register (QER)*

The QDMA event register (QER) channel *n* bit is set (E*n* = 1) when the CPU or any EDMA3 programmer (including EDMA3) performs a write to the trigger word (using the QDMA channel mapping register (QCHMAP*n*)) in the PaRAM entry associated with QDMA channel *n* (which is also programmed using QCHMAP*n*). The E*n* bit is also set when the EDMA3CC performs a link update on a PaRAM address that matches the QCHMAP*n* settings. The QDMA event is latched only if the QDMA event enable register (QEER) channel *n* bit is also enabled (QEER.E*n* = 1). Once a bit is set in QER, then the corresponding QDMA event (auto-trigger) is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. For additional conditions that can lead to the setting of QER bits, see *EDMA Overview*.

The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then the QDMA event missed register (QEMR) would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues.

The set bits in QER are only cleared when the transfer request associated with the corresponding channels has been processed by the EDMA3CC and submitted to the transfer controller. If the E*n* bit is already set and a QDMA event for the same QDMA channel occurs prior to the original being cleared, then the second missed event is latched in QEMR (E*n* = 1).

The QER is shown in Figure 8-95 and described in Table 8-75.

**Figure 8-95. QDMA Event Register (QER)**

| 31 | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | |
| R-0 | | | | | | | | |
| **15** | | | | | | | **7** | **0** |
| Reserved | | | | | | | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | | | | | | | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-75. QDMA Event Register (QER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-0 | E*n* | | QDMA event for channels 0-7. |
| | | 0 | No effect. |
| | | 1 | Corresponding QDMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.8.2 QDMA Event Enable Register (QEER)

The EDMA3CC provides the option of selectively enabling/disabling each channel in the QDMA event register (QER) by using the QDMA event enable register (QEER). If any of the event bits in QEER is set (using the QDMA event enable set register, QEESR), it will enable that corresponding event. Alternatively, if any event bit in QEER is cleared (using the QDMA event enable clear register, QEECR), it will disable the corresponding QDMA channel. The QDMA event register will not latch any event for a QDMA channel, if it is not enabled via QEER.

The QEER is shown in Figure 8-96 and described in Table 8-76.

**Figure 8-96. QDMA Event Enable Register (QEER)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-76. QDMA Event Enable Register (QEER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-0 | E*n* | | QDMA event enable for channels 0-7. |
| | | 0 | QDMA channel *n* is not enabled. QDMA event will not be recognized and will not latch in the QDMA event register (QER). |
| | | 1 | QDMA channel *n* is enabled. QDMA events will be recognized and will get latched in the QDMA event register (QER). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.8.3 QDMA Event Enable Clear Register (QEECR)

The QDMA event enable register (QEER) cannot be modified by directly writing to the register, to ease the software burden when multiple tasks are attempting to simultaneously modify these registers. The QDMA event enable clear register (QEECR) is used to disable events. Writes of 1 to the bits in QEECR clear the corresponding QDMA channel bits in QEER; writes of 0 have no effect.

The QEECR is shown in Figure 8-97 and described in Table 8-77.

**Figure 8-97. QDMA Event Enable Clear Register (QEECR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 8-77. QDMA Event Enable Clear Register (QEECR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-0 | E*n* | | QDMA event enable clear for channels 0-7. |
| | | 0 | No effect. |
| | | 1 | QDMA event is disabled. Corresponding bit in the QDMA event enable register (QEER) is cleared (E*n* = 0). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.1.8.4 QDMA Event Enable Set Register (QEESR)

The QDMA event enable register (QEER) cannot be modified by directly writing to the register, to ease the software burden when multiple tasks are attempting to simultaneously modify these registers. The QDMA event enable set register (QEESR) is used to enable events. Writes of 1 to the bits in QEESR set the corresponding QDMA channel bits in QEER; writes of 0 have no effect.

The QEESR is shown in Figure 8-98 and described in Table 8-78.

**Figure 8-98. QDMA Event Enable Set Register (QEESR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 8-78. QDMA Event Enable Set Register (QEESR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-0 | E*n* | | QDMA event enable set for channels 0-7. |
| | | 0 | No effect. |
| | | 1 | QDMA event is enabled. Corresponding bit in the QDMA event enable register (QEER) is set (E*n* = 1). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.8.5 QDMA Secondary Event Register (QSER)

The QDMA secondary event register (QSER) provides information on the state of a QDMA event. If at any time a bit corresponding to a QDMA channel is set in QSER, that implies that the corresponding QDMA event is in the queue. Once a bit corresponding to a QDMA channel is set in QSER, the EDMA3CC does not prioritize additional events on the same QDMA channel. Depending on the condition that lead to the setting of the QSER bits, either the EDMA3CC hardware or the software (using QSECR) needs to clear the QSER bits for the EDMA3CC to evaluate subsequent QDMA events on the channel. Based on whether the associated TR request is valid, or it is a null or dummy TR, the implications on the state of QSER and the required user actions to submit another QDMA transfer might be different. For additional conditions that can cause the secondary event registers (QSER\SER) to be set, see *EDMA Overview*.

The QSER is shown in Figure 8-99 and described in Table 8-79.

**Figure 8-99. QDMA Secondary Event Register (QSER)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-79. QDMA Secondary Event Register (QSER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-0 | E*n* | | QDMA secondary event register for channels 0-7. |
| | | 0 | QDMA event is not currently stored in the event queue. |
| | | 1 | QDMA event is currently stored in the event queue. EDMA3CC will not prioritize additional events. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.1.8.6 QDMA Secondary Event Clear Register (QSECR)

The QDMA secondary event clear register (QSECR) clears the status of the QDMA secondary event register (QSER) and the QDMA event register (QER). CPU writes of 1 clear the corresponding set bits in QSER and QER. Writes of 0 have no effect. Note that this differs from the secondary event clear register (SECR) operation, which only clears the secondary event register (SER) bits and does not affect the event registers.

The QSECR is shown in Figure 8-100 and described in Table 8-80.

**Figure 8-100. QDMA Secondary Event Clear Register (QSECR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |
| R-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 8-80. QDMA Secondary Event Clear Register (QSECR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | E*n* | | QDMA secondary event clear register for channels 0-7. |
| | | 0 | No effect. |
| | | 1 | Corresponding bit in the QDMA secondary event register (QSER) and the QDMA event register (QER) is cleared (E*n* = 0). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.2 EDMA3 Transfer Controller Registers

Table 8-81 lists the memory-mapped registers for the EDMA3 transfer controller (EDMA3TC).

**Table 8-81. EDMA3TC Registers**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 00h | PID | Peripheral Identification Register | Section 8.5.2.1 |
| 04h | TCCFG | EDMA3TC Configuration Register | Section 8.5.2.2 |
| 0100h | TCSTAT | EDMA3TC Channel Status Register | Section 8.5.2.3 |
| 0120h | ERRSTAT | Error Register | Section 8.5.2.4.1 |
| 0124h | ERREN | Error Enable Register | Section 8.5.2.4.2 |
| 0128h | ERRCLR | Error Clear Register | Section 8.5.2.4.3 |
| 012Ch | ERRDET | Error Details Register | Section 8.5.2.4.4 |
| 0130h | ERRCMD | Error Interrupt Command Register | Section 8.5.2.4.5 |
| 0140h | RDRATE | Read Rate Register | Section 8.5.2.5 |
| 0240h | SAOPT | Source Active Options Register | Section 8.5.2.6.1 |
| 0244h | SASRC | Source Active Source Address Register | Section 8.5.2.6.2 |
| 0248h | SACNT | Source Active Count Register | Section 8.5.2.6.3 |
| 024Ch | SADST | Source Active Destination Address Register | Section 8.5.2.6.4 |
| 0250h | SABIDX | Source Active Source B-Index Register | Section 8.5.2.6.5 |
| 0254h | SAMPPRXY | Source Active Memory Protection Proxy Register | Section 8.5.2.6.6 |
| 0258h | SACNTRLD | Source Active Count Reload Register | Section 8.5.2.6.7 |
| 025Ch | SASRCBREF | Source Active Source Address B-Reference Register | Section 8.5.2.6.8 |
| 0260h | SADSTBREF | Source Active Destination Address B-Reference Register | Section 8.5.2.6.9 |
| 0280h | DFCNTRLD | Destination FIFO Set Count Reload | Section 8.5.2.6.16 |
| 0284h | DFSRCBREF | Destination FIFO Set Destination Address B Reference Register | Section 8.5.2.6.17 |
| 0288h | DFDSTBREF | Destination FIFO Set Destination Address B Reference Register | Section 8.5.2.6.18 |
| 0300h | DFOPT0 | Destination FIFO Options Register 0 | Section 8.5.2.6.10 |
| 0304h | DFSRC0 | Destination FIFO Source Address Register 0 | Section 8.5.2.6.11 |
| 0308h | DFCNT0 | Destination FIFO Count Register 0 | Section 8.5.2.6.12 |
| 030Ch | DFDST0 | Destination FIFO Destination Address Register 0 | Section 8.5.2.6.13 |
| 0310h | DFBIDX0 | Destination FIFO BIDX Register 0 | Section 8.5.2.6.14 |
| 0314h | DFMPPRXY0 | Destination FIFO Memory Protection Proxy Register 0 | Section 8.5.2.6.15 |
| 0340h | DFOPT1 | Destination FIFO Options Register 1 | Section 8.5.2.6.10 |
| 0344h | DFSRC1 | Destination FIFO Source Address Register 1 | Section 8.5.2.6.11 |
| 0348h | DFCNT1 | Destination FIFO Count Register 1 | Section 8.5.2.6.12 |
| 034Ch | DFDST1 | Destination FIFO Destination Address Register 1 | Section 8.5.2.6.13 |
| 0350h | DFBIDX1 | Destination FIFO BIDX Register 1 | Section 8.5.2.6.14 |
| 0354h | DFMPPRXY1 | Destination FIFO Memory Protection Proxy Register 1 | Section 8.5.2.6.15 |
| 0380h | DFOPT2 | Destination FIFO Options Register 2 | Section 8.5.2.6.10 |
| 0384h | DFSRC2 | Destination FIFO Source Address Register 2 | Section 8.5.2.6.11 |
| 0388h | DFCNT2 | Destination FIFO Count Register 2 | Section 8.5.2.6.12 |
| 038Ch | DFDST2 | Destination FIFO Destination Address Register 2 | Section 8.5.2.6.13 |
| 0390h | DFBIDX2 | Destination FIFO BIDX Register 2 | Section 8.5.2.6.14 |
| 0394h | DFMPPRXY2 | Destination FIFO Memory Protection Proxy Register 2 | Section 8.5.2.6.15 |
| 03C0h | DFOPT3 | Destination FIFO Options Register 3 | Section 8.5.2.6.10 |
| 03C4h | DFSRC3 | Destination FIFO Source Address Register 3 | Section 8.5.2.6.11 |
| 03C8h | DFCNT3 | Destination FIFO Count Register 3 | Section 8.5.2.6.12 |
| 03CCh | DFDST3 | Destination FIFO Destination Address Register 3 | Section 8.5.2.6.13 |
| 03D0h | DFBIDX3 | Destination FIFO BIDX Register 3 | Section 8.5.2.6.14 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-81. EDMA3TC Registers (continued)**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 03D4h | DFMPPRXY3 | Destination FIFO Memory Protection Proxy Register 3 | Section 8.5.2.6.15 |

### 8.5.2.1 Peripheral Identification Register (PID)

The peripheral identification register (PID) is a constant register that uniquely identifies the EDMA3TC and specific revision of the EDMA3TC. The PID is shown in Figure 8-101 and described in Table 8-82.

**Figure 8-101. Peripheral ID Register (PID)**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| PID | | PID | |
| R-0333h | | R-4425h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-82. Peripheral ID Register (PID) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | PID | 0-FFFF FFFFh | Peripheral identifier. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.2.2 EDMA3TC Configuration Register (TCCFG)

The EDMA3TC configuration register (TCCFG) is shown in Figure 8-102 and described in Table 8-83.

**Figure 8-102. EDMA3TC Configuration Register (TCCFG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | DREGDEPTH | | Reserved | | BUSWIDTH | | Rsvd | FIFOSIZE | |
| R-0 | | R-x | | R-0 | | R-n | | R-0 | R-x | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset; -x = value is indeterminate after reset

**Table 8-83. EDMA3TC Configuration Register (TCCFG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 9-8 | DREGDEPTH | 0-3h | Destination register FIFO depth parameterization. |
| | | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 1h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 2h | 4 entry (for TC0, TC1, TC2, and TC3) |
| | | 3h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7-6 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 5-4 | BUSWIDTH | 0-3h | Bus width parameterization. |
| | | 0-1h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 2h | 128-bit |
| | | 3h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 2-0 | FIFOSIZE | 0-7h | FIFO size |
| | | 0-1h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 2h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 3h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 4h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| | | 5h | 1024 byte FIFO (for TC0, TC1, TC2 and TC3) |
| | | 6h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.2.3 EDMA3TC Channel Status Register (TCSTAT)

The EDMA3TC channel status register (TCSTAT) is shown in Figure 8-103 and described in Table 8-84.

**Figure 8-103. EDMA3TC Channel Status Register (TCSTAT)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 12 | 11 | 9 | 8 |
|---|---|---|---|---|---|---|
| Reserved | | DFSTRTPTR | | Reserved | | Reserved |
| R-0 | | R-0 | | R-0 | | R-1 |

| 7 | 6 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | DSTACTV | | Reserved | WSACTV | SRCACTV | PROGBUSY |
| R-0 | R-0 | | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-84. EDMA3TC Channel Status Register (TCSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-14 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 13-12 | DFSTRTPTR | 0-3h | Destination FIFO start pointer. Represents the offset to the head entry of the destination register FIFO, in units of entries. |
| 11-9 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 8 | Reserved | 1 | Reserved. Always read as 1. |
| 7 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 6-4 | DSTACTV | 0-7h | Destination active state. Specifies the number of transfer requests (TRs) that are resident in the destination register FIFO at a given instant. This bit field can be primarily used for advanced debugging. Legal values are constrained by the destination register FIFO depth parameterization (DSTREGDEPTH) parameter. |
| | | 0 | Destination FIFO is empty. |
| | | 1h | Destination FIFO contains 1 TR. |
| | | 2h | Destination FIFO contains 2 TRs. |
| | | 3h | Destination FIFO contains 3 TRs. |
| | | 4h | Destination FIFO contains 4 TRs. (Full if DSTREGDEPTH==4). |
| | | | If the destination register FIFO is empty, then any TR written to Prog Set immediately transitions to the destination register FIFO. If the destination register FIFO is not empty and not full, then any TR written to Prog Set immediately transitions to the destination register FIFO set if the source active state (SRCACTV) bit is set to idle. |
| | | | If the destination register FIFO is full, then TRs cannot transition to the destination register FIFO. The destination register FIFO becomes not full when the TR at the head of the destination register FIFO is completed. |
| | | 5h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 2 | WSACTV | | Write status active |
| | | 0 | Write status is not pending. Write status has been received for all previously issued write commands. |
| | | 1 | Write status is pending. Write status has not been received for all previously issued write commands. |
| 1 | SRCACTV | | Source active state |
| | | 0 | Source controller is idle. Source active register set contains a previously processed transfer request. |
| | | 1 | Source controller is busy servicing a transfer request. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-84. EDMA3TC Channel Status Register (TCSTAT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | PROGBUSY | | Program register set busy |
| | | 0 | Program set idle and is available for programming by the EDMA3CC. |
| | | 1 | Program set busy |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.2.4 Error Registers

#### *8.5.2.4.1 Error Register (ERRSTAT)*

The error status register (ERRSTAT) is shown in Figure 8-104 and described in Table 8-85.

**Figure 8-104. Error Register (ERRSTAT)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | MMRAERR | TRERR | Reserved | BUSERR |
| R-0 | | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-85. Error Register (ERRSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3 | MMRAERR | | MMR address error. |
| | | 0 | Condition is not detected. |
| | | 1 | User attempted to read or write to an invalid address in configuration memory map. |
| 2 | TRERR | | Transfer request (TR) error event. |
| | | 0 | Condition is not detected. |
| | | 1 | TR detected that violates constant addressing mode transfer (SAM or DAM is set) alignment rules or has ACNT or BCNT == 0. |
| 1 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 0 | BUSERR | | Bus error event. |
| | | 0 | Condition is not detected. |
| | | 1 | EDMA3TC has detected an error at source or destination address. Error information can be read from the error details register (ERRDET). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.4.2 Error Enable Register (ERREN)

The error enable register (ERREN) is shown in Figure 8-105 and described in Table 8-86.

When any of the enable bits are set, a bit set in the corresponding ERRSTAT causes an assertion of the EDMA3TC interrupt.

**Figure 8-105. Error Enable Register (ERREN)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | MMRAERR | TRERR | Reserved | BUSERR |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-86. Error Enable Register (ERREN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3 | MMRAERR | | Interrupt enable for MMR address error (MMRAERR). |
| | | 0 | MMRAERR is disabled. |
| | | 1 | MMRAERR is enabled and contributes to the state of EDMA3TC error interrupt generation |
| 2 | TRERR | | Interrupt enable for transfer request error (TRERR). |
| | | 0 | TRERR is disabled. |
| | | 1 | TRERR is enabled and contributes to the state of EDMA3TC error interrupt generation. |
| 1 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 0 | BUSERR | | Interrupt enable for bus error (BUSERR). |
| | | 0 | BUSERR is disabled. |
| | | 1 | BUSERR is enabled and contributes to the state of EDMA3TC error interrupt generation. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.2.4.3 Error Clear Register (ERRCLR)

The error clear register (ERRCLR) is shown in Figure 8-106 and described in Table 8-87.

**Figure 8-106. Error Clear Register (ERRCLR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | MMRAERR | TRERR | Reserved | BUSERR |
| R-0 | | W-0 | W-0 | R-0 | W-0 |

LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 8-87. Error Clear Register (ERRCLR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3 | MMRAERR | | Interrupt enable clear for the MMRAERR bit in the error status register (ERRSTAT). |
| | | 0 | No effect. |
| | | 1 | Clears the MMRAERR bit in ERRSTAT but does not clear the error details register (ERRDET). |
| 2 | TRERR | | Interrupt enable clear for the TRERR bit in the error status register (ERRSTAT). |
| | | 0 | No effect. |
| | | 1 | Clears the TRERR bit in ERRSTAT but does not clear the error details register (ERRDET). |
| 1 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 0 | BUSERR | | Interrupt clear for the BUSERR bit in the error status register (ERRSTAT). |
| | | 0 | No effect. |
| | | 1 | Clears the BUSERR bit in ERRSTAT and clears the error details register (ERRDET). |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.4.4 Error Details Register (ERRDET)

The error details register (ERRDET) is shown in Figure 8-107 and described in Table 8-88.

**Figure 8-107. Error Details Register (ERRDET)**

| 31 | 18 | 17 | 16 |
|---|---|---|---|
| Reserved | | TCCHEN | TCINTEN |
| R-0 | | R-0 | R-0 |

| 15 | 14 | 13 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | TCC | | Reserved | | STAT | |
| R-0 | | R-0 | | R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 8-88. Error Details Register (ERRDET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 17 | TCCHEN | 0-1 | Transfer completion chaining enable. Contains the TCCHEN value in the channel options parameter (OPT) programmed by the channel controller for the read or write transaction that resulted in an error. |
| 16 | TCINTEN | 0-1 | Transfer completion interrupt enable. Contains the TCINTEN value in the channel options parameter (OPT) programmed by the channel controller for the read or write transaction that resulted in an error. |
| 15-14 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 13 - 8 | TCC | 0-3Fh | Transfer complete code. Contains the TCC value in the channel options parameter (OPT) programmed by the channel controller for the read or write transaction that resulted in an error. |
| 7-4 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3-0 | STAT | 0-Fh | Transaction status. Stores the nonzero status/error code that was detected on the read status or write status bus. If read status and write status are returned on the same cycle, then the EDMA3TC chooses nonzero version. If both are nonzero, then the write status is treated as higher priority. |
| | | 0 | No error. |
| | | 1h-7h | Read error. |
| | | 8h-Fh | Write error. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.2.4.5 Error Interrupt Command Register (ERRCMD)

The error command register (ERRCMD) is shown in Figure 8-108 and described in Table 8-89.

**Figure 8-108. Error Interrupt Command Register (ERRCMD)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | Rsvd | EVAL |
| R-0 | | W-0 | W-0 |

LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 8-89. Error Interrupt Command Register (ERRCMD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 1 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 0 | EVAL | | Error evaluate |
| | | 0 | No effect |
| | | 1 | EDMA3TC error line is pulsed if any of the error status register (ERRSTAT) bits are set. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.2.5 Read Rate Register (RDRATE)

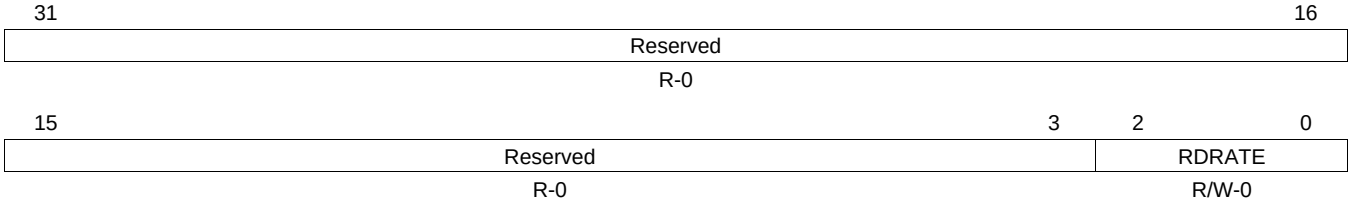
The EDMA3 transfer controller issues read commands at a rate controlled by the read rate register (RDRATE). The RDRATE defines the number of idle cycles that the read controller must wait before issuing subsequent commands. This applies both to commands within a transfer request packet (TRP) and for commands that are issued for different transfer requests (TRs). For instance, if RDRATE is set to 4 cycles between reads, there are 3 inactive cycles between reads.

RDRATE allows flexibility in transfer controller access requests to an endpoint. For an application, RDRATE can be manipulated to slow down the access rate, so that the endpoint may service requests from other masters during the inactive EDMA3TC cycles.

The RDRATE is shown in Figure 8-109 and described in Table 8-90.

**NOTE:** It is expected that the RDRATE value for a transfer controller is static, as it is decided based on the application requirement. It is not recommended to change this setting on the fly.

**Figure 8-109. Read Rate Register (RDRATE)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-90. Read Rate Register (RDRATE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 2-0 | RDRATE | 0-7h | Read rate. Controls the number of cycles between read commands. This is a global setting that applies to all TRs for this EDMA3TC. |
| | | 0 | Reads issued as fast as possible. |
| | | 1h | 4 cycles between reads. |
| | | 2h | 8 cycles between reads. |
| | | 3h | 16 cycles between reads. |
| | | 4h | 32 cycles between reads. |
| | | 5h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.2.6 EDMA3TC Channel Registers

The EDMA3TC channel registers are split into three parts: the programming registers, the source active registers, and the destination FIFO register. This section describes the registers and their functions. The program register set is programmed by the channel controller, and is for internal use. The other two sets are read-only and provided to facilitate advanced debug capabilities. The number of destination FIFO register sets depends on the destination FIFO depth.

TC0 has a FIFO depth of 2, so there are two sets of destination FIFO registers associated with TC0. TC1, TC2, and TC3 have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

#### *8.5.2.6.1 Source Active Options Register (SAOPT)*

The source active options register (SAOPT) is shown in Figure 8-110 and described in Table 8-91.

**Figure 8-110. Source Active Options Register (SAOPT)**

| 31 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | TCCHEN | Rsvd | TCINTEN | Reserved | | TCC | |
| R-0 | | R/W-0 | R-0 | R/W-0 | R-0 | | R/W-0 | |

| 15 | 12 | 11 | 10 | 8 | 7 | 6 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TCC | | Rsvd | FWID | | Rsvd | PRI | | Reserved | | DAM | SAM |
| R/W-0 | | R-0 | R/W-0 | | R-0 | R/W-0 | | R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 8-91. Source Active Options Register (SAOPT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-23 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 22 | TCCHEN | | Transfer complete chaining enable |
| | | 0 | Transfer complete chaining is disabled. |
| | | 1 | Transfer complete chaining is enabled. |
| 21 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 20 | TCINTEN | | Transfer complete interrupt enable. |
| | | 0 | Transfer complete interrupt is disabled. |
| | | 1 | Transfer complete interrupt is enabled. |
| 19-18 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 17-12 | TCC | 0-3Fh | Transfer complete code. This 6-bit code is used to set the relevant bit in CER or IPR of the EDMA3PCC module. |
| 11 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 10-8 | FWID | 0-7h | FIFO width. Applies if either SAM or DAM is set to constant addressing mode. |
| | | 0 | FIFO width is 8-bit. |
| | | 1h | FIFO width is 16-bit. |
| | | 2h | FIFO width is 32-bit. |
| | | 3h | FIFO width is 64-bit. |
| | | 4h | FIFO width is 128-bit. |
| | | 5h | FIFO width is 256-bit. |
| | | 6h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-91. Source Active Options Register (SAOPT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 6-4 | PRI | 0-7h | Transfer priority. Reflects the values programmed in the QUEPRI register in the EDMACC. |
| | | 0 | Priority 0 - Highest priority |
| | | 1h-6h | Priority 1 to priority 6 |
| | | 7h | Priority 7 - Lowest priority |
| 3-2 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 1 | DAM | | Destination address mode within an array |
| | | 0 | Increment (INCR) mode. Destination addressing within an array increments. |
| | | 1 | Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width. |
| 0 | SAM | | Source address mode within an array |
| | | 0 | Increment (INCR) mode. Source addressing within an array increments. |
| | | 1 | Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.6.2 Source Active Source Address Register (SASRC)

The source active source address register (SASRC) is shown in Figure 8-111 and described in Table 8-92.

**Figure 8-111. Source Active Source Address Register (SASRC)**

| 31 | 0 |
|---|---|
| SADDR | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-92. Source Active Source Address Register (SASRC) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SADDR | 0-FFFF FFFFh | Source address for program register set. EDMA3TC updates value according to source addressing mode (SAM bit in the source active options register, SAOPT) . |

#### 8.5.2.6.3 Source Active Count Register (SACNT)

The source active count register (SACNT) is shown in Figure 8-112 and described in Table 8-93.

**Figure 8-112. Source Active Count Register (SACNT)**

| 31 | 16 |
|---|---|
| BCNT | |
| R-0 | |
| 15 | 0 |
| ACNT | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-93. Source Active Count Register (SACNT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | BCNT | 0-FFFFh | B dimension count. Number of arrays to be transferred, where each array is ACNT in length. It is decremented after each read command appropriately. Represents the amount of data remaining to be read. It should be 0 when transfer request (TR) is complete. |
| 15-0 | ACNT | 0-FFFFh | A dimension count. Number of bytes to be transferred in first dimension. It is decremented after each read command appropriately. Represents the amount of data remaining to be read. It should be 0 when transfer request (TR) is complete. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.6.4 Source Active Destination Address Register (SADST)

The source active destination address register (SADST) is shown in Figure 8-113 and described in Table 8-94.

**Figure 8-113. Source Active Destination Address Register (SADST)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 0 |
|---|---|
| Reserved | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-94. Source Active Destination Address Register (SADST) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Reserved | 0 | Reserved. Always reads as 0. |

#### 8.5.2.6.5 Source Active Source B-Dimension Index Register (SABIDX)

The source active set B-dimension index register (SABIDX) is shown in Figure 8-114 and described in Table 8-95.

**Figure 8-114. Source Active Source B-Dimension Index Register (SABIDX)**

| 31 | 16 |
|---|---|
| DBIDX | |
| R-0 | |

| 15 | 0 |
|---|---|
| SBIDX | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-95. Source Active Source B-Dimension Index Register (SABIDX) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | DBIDX | 0 | B-Index offset between destination arrays. Represents the offset in bytes between the starting address of each destination. Always reads as 0. |
| 15-0 | SBIDX | 0-FFFFh | B-Index offset between source arrays. Represents the offset in bytes between the starting address of each source array. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.6.6 Source Active Memory Protection Proxy Register (SAMPPRXY)

The source active memory protection proxy register (SAMPPRXY) is shown in Figure 8-115 and described in Table 8-96.

**Figure 8-115. Source Active Memory Protection Proxy Register (SAMPPRXY)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 9 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | PRIV | Reserved | | PRIVID | |
| R-0 | | R-0 | R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-96. Source Active Memory Protection Proxy Register (SAMPPRXY) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 8 | PRIV | | Privilege level. The privilege level used by the host to set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC.<br>The privilege ID is used while issuing read and write command to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIV of the host that set up the DMA transaction. |
| | | 0 | User-level privilege. |
| | | 1 | Supervisor-level privilege. |
| 7-4 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3-0 | PRIVID | 0-Fh | Privilege ID. This contains the privilege ID of the host that set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC.<br>This PRIVID value is used while issuing read and write commands to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIVID of the host that set up the DMA transaction. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.6.7 Source Active Count Reload Register (SACNTRLD)

The source active count reload register (SACNTRLD) is shown in Figure 8-116 and described in Table 8-97.

**Figure 8-116. Source Active Count Reload Register (SACNTRLD)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |
| 15 | 0 |
| ACNTRLD | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-97. Source Active Count Reload Register (SACNTRLD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 15-0 | ACNTRLD | 0-FFFFh | A-count reload value. Represents the originally programmed value of ACNT. The reload value is used to reinitialize ACNT after each array is serviced. |

#### 8.5.2.6.8 Source Active Source Address B-Reference Register (SASRCBREF)

The source active source address B-reference register (SASRCBREF) is shown in Figure 8-117 and described in Table 8-98.

**Figure 8-117. Source Active Source Address B-Reference Register (SASRCBREF)**

| 31 | 16 |
|---|---|
| SADDRBREF | |
| R-0 | |
| 15 | 0 |
| SADDRBREF | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-98. Source Active Source Address B-Reference Register (SASRCBREF) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SADDRBREF | 0-FFFF FFFFh | Source address B-reference. Represents the starting address for the array currently being read. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.6.9 Source Active Destination Address B-Reference Register (SADSTBREF)

The source active destination address B-reference register (SADSTBREF) is shown in Figure 8-118 and described in Table 8-99.

**Figure 8-118. Source Active Destination Address B-Reference Register (SADSTBREF)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 0 |
|---|---|
| Reserved | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-99. Source Active Destination Address B-Reference Register (SADSTBREF) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Reserved | 0 | Reserved. Always reads as 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.5.2.6.10 Destination FIFO Options Register (DFOPT*n*)

The destination FIFO options register (DFOPT*n*) is shown in Figure 8-119 and described in Table 8-100.

**NOTE:** The value for *n* varies from 0 to DSTREGDEPTH for the given EDMA3TC.

**Figure 8-119. Destination FIFO Options Register (DFOPT*n*)**

| 31 | | | | | | | | 23 | 22 | 21 | 20 | 19 | | 18 | 17 | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | TCCHEN | Rsvd | TCINTEN | Reserved | | | TCC | | |
| R-0 | | | | | | | | | R/W-0 | R-0 | R/W-0 | R-0 | | | R/W-0 | | |
| 15 | | | 12 | 11 | 10 | | 8 | 7 | 6 | | 4 | 3 | | 2 | 1 | | 0 |
| TCC | | | | Rsvd | FWID | | | Rsvd | PRI | | | Reserved | | | DAM | | SAM |
| R/W-0 | | | | R-0 | R/W-0 | | | R-0 | R/W-0 | | | R-0 | | | R/W-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 8-100. Destination FIFO Options Register (DFOPT*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-23 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 22 | TCCHEN | | Transfer complete chaining enable |
| | | 0 | Transfer complete chaining is disabled |
| | | 1 | Transfer complete chaining is enabled |
| 21 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 20 | TCINTEN | | Transfer complete interrupt enable. |
| | | 0 | Transfer complete interrupt is disabled. |
| | | 1 | Transfer complete interrupt is enabled. |
| 19-18 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 17-12 | TCC | 0-3Fh | Transfer complete code. This 6-bit code is used to set the relevant bit in CER or IPR of the EDMA3PCC module. |
| 11 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 10-8 | FWID | 0-7h | FIFO width. Applies if either SAM or DAM is set to constant addressing mode. |
| | | 0 | FIFO width is 8-bit. |
| | | 1h | FIFO width is 16-bit. |
| | | 2h | FIFO width is 32-bit. |
| | | 3h | FIFO width is 64-bit. |
| | | 4h | FIFO width is 128-bit. |
| | | 5h | FIFO width is 256-bit. |
| | | 6h-7h | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 7 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 6-4 | PRI | 0-7h | Transfer priority |
| | | 0 | Priority 0 - Highest priority |
| | | 1h-6h | Priority 1 to priority 6 |
| | | 7h | Priority 7 - Lowest priority |
| 3-2 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 8-100. Destination FIFO Options Register (DFOPT*n*) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | DAM | | Destination address mode within an array |
| | | 0 | Increment (INCR) mode. Destination addressing within an array increments. |
| | | 1 | Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width. |
| 0 | SAM | | Source address mode within an array |
| | | 0 | Increment (INCR) mode. Source addressing within an array increments. |
| | | 1 | Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.6.11 Destination FIFO Source Address Register (DFSRC*n*)

The destination FIFO source address register (DFSRC*n*) is shown in Figure 8-120 and described in Table 8-101.

**NOTE:** The value for *n* varies from 0 to DSTREGDEPTH for the given EDMA3TC.

**Figure 8-120. Destination FIFO Source Address Register (DFSRC*n*)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 0 |
|---|---|
| Reserved | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-101. Destination FIFO Source Address Register (DFSRC*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Reserved | 0 | Reserved. Always read as 0. |

#### 8.5.2.6.12 Destination FIFO Count Register (DFCNT*n*)

The destination FIFO count register (DFCNT*n*) is shown in Figure 8-121 and described in Table 8-102.

**NOTE:** The value for *n* varies from 0 to DSTREGDEPTH for the given EDMA3TC.

**Figure 8-121. Destination FIFO Count Register (DFCNT*n*)**

| 31 | 16 |
|---|---|
| BCNT | |
| R-0 | |

| 15 | 0 |
|---|---|
| ACNT | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-102. Destination FIFO Count Register (DFCNT*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | BCNT | 0-FFFFh | B-dimension count. Number of arrays to be transferred, where each array is ACNT in length. Count/count remaining for destination register set. Represents the amount of data remaining to be written. |
| 15-0 | ACNT | 0-FFFFh | A-dimension count. Number of bytes to be transferred in first dimension count/count remaining for destination register set. Represents the amount of data remaining to be written. |

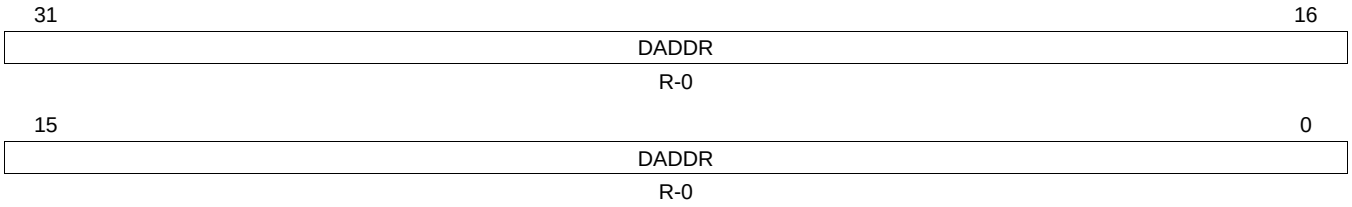

Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.6.13 Destination FIFO Destination Address Register (DFDSTn)

The destination FIFO destination address register (DFDSTn) is shown in Figure 8-122 and described in Table 8-103.

**NOTE:** The value for *n* varies from 0 to DSTREGDEPTH for the given EDMA3TC.

**Figure 8-122. Destination FIFO Destination Address Register (DFDSTn)**




LEGEND: R = Read only; -*n* = value after reset

**Table 8-103. Destination FIFO Destination Address Register (DFDSTn) Field Descriptions**

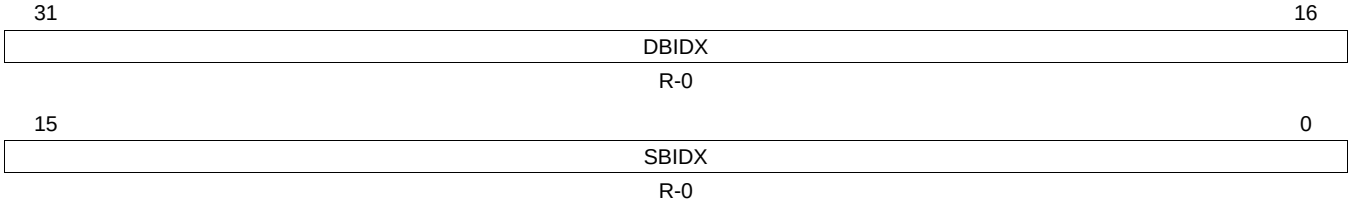
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | DADDR | 0 | Destination address for the destination FIFO register set. When a transfer request (TR) is complete, the final value should be the address of the last write command issued. |

#### 8.5.2.6.14 Destination FIFO B-Index Register (DFBIDXn)

The destination FIFO B-index register (DFBIDXn) is shown in Figure 8-123 and described in Table 8-104.

**NOTE:** The value for *n* varies from 0 to DSTREGDEPTH for the given EDMA3TC.

**Figure 8-123. Destination FIFO B-Index Register (DFBIDXn)**




LEGEND: R = Read only; -*n* = value after reset

**Table 8-104. Destination FIFO B-Index Register (DFBIDXn) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | DBIDX | 0-FFFFh | B-Index offset between destination arrays. Represents the offset in bytes between the starting address of each destination. |
| 15-0 | SBIDX | 0 | Always read as 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.6.15 Destination FIFO Memory Protection Proxy Register (DFMPPRXY*n*)

The destination FIFO memory protection proxy register (DFMPPRXY*n*) is shown in Figure 8-124 and described in Table 8-96.

**Figure 8-124. Destination FIFO Memory Protection Proxy Register (DFMPPRXY*n*)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 9 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | PRIV | Reserved | | PRIVID | |
| R-0 | | R-0 | R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-105. Destination FIFO Memory Protection Proxy Register (DFMPPRXY*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 8 | PRIV | | Privilege level. This contains the Privilege level used by the EDMA3 programmer to set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC.<br>The privilege ID is used while issuing read and write command to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIV of the host that set up the DMA transaction. |
| | | 0 | User-level privilege |
| | | 1 | Supervisor-level privilege |
| 7-4 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 3-0 | PRIVID | 0-Fh | Privilege ID. This contains the Privilege ID of the EDMA3 programmer that set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC.<br>This PRIVID value is used while issuing read and write commands to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIVID of the host that set up the DMA transaction. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.6.16 Destination FIFO Count Reload Register (DFCNTRLD*n*)

The destination FIFO count reload register (DFCNTRLD*n*) is shown in Figure 8-125 and described in Table 8-106.

**Figure 8-125. Destination FIFO Count Reload Register (DFCNTRLD*n*)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 0 |
|---|---|
| ACNTRLD | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-106. Destination FIFO Count Reload Register (DFCNTRLD*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior. |
| 15-0 | ACNTRLD | 0-FFFFh | A-count reload value. Represents the originally programmed value of ACNT. The reload value is used to reinitialize ACNT after each array is serviced. |

#### 8.5.2.6.17 Destination FIFO Source Address B-Reference Register (DFSRCBREF*n*)

The destination FIFO source address B-reference register (DFSRCBREF*n*) is shown in Figure 8-126 and described in Table 8-107.

**Figure 8-126. Destination FIFO Source Address B-Reference Register (DFSRCBREF*n*)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 0 |
|---|---|
| Reserved | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-107. Destination FIFO Source Address B-Reference Register (DFSRCBREF*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Reserved | 0 | Reserved. Always read as 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 8.5.2.6.18 Destination FIFO Destination Address B-Reference (DFDSTBREF*n*)

The destination FIFO destination address B-reference register (DFDSTBREF*n*) is shown in Figure 8-127 and described in Table 8-108.

**Figure 8-127. Destination FIFO Destination Address B-Reference Register (DFDSTBREF*n*)**

| 31 | 16 |
|---|---|
| DADDRBREF | |
| R-0 | |

| 15 | 0 |
|---|---|
| DADDRBREF | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 8-108. Destination FIFO Destination Address B-Reference Register (DFDSTBREF*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | DADDRBREF | 0-FFFF FFFFh | Destination address reference for the destination FIFO register set. Represents the starting address for the array currently being written. |


Copyright © 2011–2016, Texas Instruments Incorporated

## 8.6 Appendix A

### 8.6.1 Debug Checklist

This section lists some tips to keep in mind while debugging applications using the EDMA3.

The following table provides some common issues and their probable causes and resolutions.

**Table 8-109. Debug List**

| Issue | Description/Solution |
|---|---|
| The transfer associated with the channel does not happen. The channel does not get serviced. | The EDMA3CC may not service a transfer request, even though the associated PaRAM set is programmed appropriately. Check for the following:<br>1) Verify that events are enabled, i.e., if an external/peripheral event is latched in Event Registers (ER/ERH), make sure that the event is enabled in the Event Enable Registers (EER/EERH). Similarly, for QDMA channels, make sure that QDMA events are appropriately enabled in the QDMA Event Enable Register (QEER).<br>2) Verify that the DMA or QDMA Secondary Event Register (SER/SERH/QSERH) bits corresponding to the particular event or channel are not set. |
| The Secondary Event Registers bits are set, not allowing additional transfers to occur on a channel. | It is possible that a trigger event was received when the parameter set associated with the channel/event was a NULL set for a previous transfer on the channel. This is typical in two cases:<br>1) QDMA channels: Typically if the parameter set is non-static and expected to be terminated by a NULL set (i.e., OPT.STATIC = 0, LINK = 0xFFFF), the parameter set is updated with a NULL set after submission of the last TR. Because QDMA channels are auto-triggered, this update caused the generation of an event. An event generated for a NULL set causes an error condition and results in setting the bits corresponding to the QDMA channel in the QEMR and QSER. This will disable further prioritization of the channel.<br>2) DMA channels used in a continuous mode: The peripheral may be set up to continuously generate infinite events (for instance, in case of McASP, every time the data shifts out from the DXR register, it generates an XEVT). The parameter set may be programmed to expect only a finite number of events and to be terminated by a NULL link. After the expected number of events, the parameter set is reloaded with a NULL parameter set. Because the peripheral will generate additional events, an error condition is set in the SER.Ex and EMR.Ex set, preventing further event prioritization. You must ensure that the number of events received is limited to the expected number of events for which the parameter set is programmed, or you must ensure that bits corresponding to particular channel or event are not set in the Secondary event registers (SER/SERH/QSER) and Event Missed Registers (EMR/EMRH/QEMR) before trying to perform subsequent transfers for the event/channel. |
| Completion interrupts are not asserted, or no further interrupts are received after the first completion interrupt. | You must ensure the following:<br>1) The interrupt generation is enabled in the OPT of the associated PaRAM set (TCINTEN = 1 and/or ITCINTEN = 1).<br>2) The interrupts are enabled in the EDMA3 Channel Controller, via the Interrupt Enable Registers (IER/IERH).<br>3) The corresponding interrupts are enabled in the device interrupt controller.<br>4) The set interrupts are cleared in the interrupt pending registers (IPR/IPRH) before exiting the transfer completion interrupt service routine (ISR). See Section 8.3.9.1.2 for details on writing EDMA3 ISRs.<br>5) If working with shadow region interrupts, make sure that the DMA Region Access registers (DRAE/DRAEH) are set up properly, because the DRAE/DRAEH registers act as secondary enables for shadow region completion interrupts, along with the IER/IERH registers.<br>If working with shadow region interrupts, make sure that the bits corresponding to the transfer completion code (TCC) value are also enabled in the DRAE/DRAEH registers. For instance, if the PaRAM set associated with Channel 0 returns a completion code of 63 (OPT.TCC=63), ensure that DRAEH.E63 is also set for a shadow region completion interrupt because the interrupt pending register bit set will be IPRH.I63 (not IPR.I0). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 8.6.2 Miscellaneous Programming/Debug Tips

1. For several registers, the setting and clearing of bits needs to be done via separate dedicated registers. For example, the Event Register (ER/ERH) can only be cleared by writing a 1 to the corresponding bits in the Event Clear Registers (ECR/ECRH). Similarly, the Event Enable Register (EER/EERH) bits can only be set with writes of 1 to the Event Enable Set Registers (EESR/EESRH) and cleared with writes of 1 to the corresponding bits in the Event Enable Clear Register (EECR/EECRH).
2. Writes to the shadow region memory maps are governed by region access registers (DRAE/DRAEH/QRAE). If the appropriate channels are not enabled in these registers, read/write access to the shadow region memory map is not enabled.
3. When working with shadow region completion interrupts, ensure that the DMA Region Access Registers (DRAE/DRAEH) for every region are set in a mutually exclusive way (unless it is a requirement for an application). If there is an overlap in the allocated channels and transfer completion codes (setting of Interrupt Pending Register bits) in the region resource allocation, it results in multiple shadow region completion interrupts. For example, if DRAE0.E0 and DRAE1.E0 are both set, then on completion of a transfer that returns a TCC=0, they will generate both shadow region 0 and 1 completion interrupts.
4. While programming a non-dummy parameter set, ensure the CCNT is not left to zero.
5. Enable the EDMA3CC error interrupt in the device controller and attach an interrupt service routine (ISR) to ensure that error conditions are not missed in an application and are appropriately addressed with the ISR.
6. Depending on the application, you may want to break large transfers into smaller transfers and use self-chaining to prevent starvation of other events in an event queue.
7. In applications where a large transfer is broken into sets of small transfers using chaining or other methods, you might choose to use the early chaining option to reduce the time between the sets of transfers and increase the throughput. However, keep in mind that with early completion, all data might have not been received at the end point when completion is reported because the EDMA3CC internally signals completion when the TR is submitted to the EDMA3TC, potentially before any data has been transferred.
8. The event queue entries can be observed to determine the last few events if there is a system failure (provided the entries were not bypassed).


Copyright © 2011–2016, Texas Instruments Incorporated

## 8.7 Setting Up a Transfer

The following list provides a quick guide for the typical steps involved in setting up a transfer.

Step 1. Initiating a DMA/QDMA channel

(a) Determine the type of channel (QDMA or DMA) to be used.

(b) Channel mapping

(i) If using a QDMA channel, program the QCHMAP with the parameter set number to which the channel maps and the trigger word.

(ii) If using a DMA channel, program the DCHMAP with the parameter set number to which the channel maps.

(c) If the channel is being used in the context of a shadow region, ensure the DRAE/DRAEH for the region is properly set up to allow read write accesses to bits in the event registers and interrupt registers in the Shadow region memory map. The subsequent steps in this process should be done using the respective shadow region registers. (Shadow region descriptions and usage are provided in Section 8.3.7.1.)

(d) Determine the type of triggering used.

(i) If external events are used for triggering (DMA channels), enable the respective event in EER/EERH by writing into EESR/EESRH.

(ii) If QDMA Channel is used, enable the channel in QEER by writing into QEESR.

(e) Queue setup

(i) If a QDMA channel is used, set up the QDMAQNUM to map the channel to the respective event queue.

(ii) If a DMA channel is used, set up the DMAQNUM to map the event to the respective event queue.

Step 2. Parameter set setup

(a) Program the PaRAM set number associated with the channel. Note that if it is a QDMA channel, the PaRAM entry that is configured as trigger word is written to last. Alternatively, enable the QDMA channel (step 1-b-ii above) just before the write to the trigger word.
See Section 8.4 for parameter set field setups for different types of transfers. See the sections on chaining (Section 8.3.8) and interrupt completion (Section 8.3.9) on how to set up final/intermediate completion chaining and/or interrupts.

Step 3. Interrupt setup

(a) Enable the interrupt in the IER/IERH by writing into IESR/IESRH.

(b) Ensure that the EDMA3CC completion interrupt (either the global or the shadow region interrupt) is enabled properly in the device interrupt controller.

(c) Ensure the EDMA3CC completion interrupt (this refers to either the Global interrupt or the shadow region interrupt) is enabled properly in the Device Interrupt controller.

(d) Set up the interrupt controller properly to receive the expected EDMA3 interrupt.


Copyright © 2011–2016, Texas Instruments Incorporated

Step 4. Initiate transfer

(a) This step is highly dependent on the event trigger source:

(i) If the source is an external event coming from a peripheral, the peripheral will be enabled to start generating relevant EDMA3 events that can be latched to the ER transfer.

(ii) For QDMA events, writes to the trigger word (step 2-a above) will initiate the transfer.

(iii) Manually triggered transfers will be initiated by writes to the Event Set Registers (ESR/ESRH).

(iv) Chained-trigger events initiate when a previous transfer returns a transfer completion code equal to the chained channel number.

Step 5. Wait for completion

(a) If the interrupts are enabled as mentioned in step 3 above, then the EDMA3CC will generate a completion interrupt to the CPU whenever transfer completion results in setting the corresponding bits in the interrupt pending register (IPR/IPRH). The set bits must be cleared in the IPR\IPRH by writing to corresponding bit in ICR\ICRH.

(b) If polling for completion (interrupts not enabled in the device controller), then the application code can wait on the expected bits to be set in the IPR\IPRH. Again, the set bits in the IPR\IPRH must be manually cleared via ICR\ICRH before the next set of transfers is performed for the same transfer completion code values.


Copyright © 2011–2016, Texas Instruments Incorporated

# 3PSW Ethernet Subsystem (EMAC)

This chapter describes the three port switch (3PSW) Ethernet subsystem in the device.

**Topic** **Page**


| | | |
|---|---|---|
| **9.1** | **Introduction** | **1451** |
| **9.2** | **CPSW Architecture** | **1457** |
| **9.3** | **Software Operation** | **1503** |
| **9.4** | **Registers** | **1509** |



Copyright © 2011–2016, Texas Instruments Incorporated

## 9.1 Introduction

The three port switch gigabit ethernet subsystem provides ethernet packet communication and can be configured as an ethernet switch. It provides the gigabit media independent interface (G/MII), reduced gigabit media independent interface (RGMII), reduced media independent interface (RMII), the management data input output (MDIO) for physical layer device (PHY) management.

### *9.1.1 Features*

The 3PSW Ethernet Subsystem provides these functions:

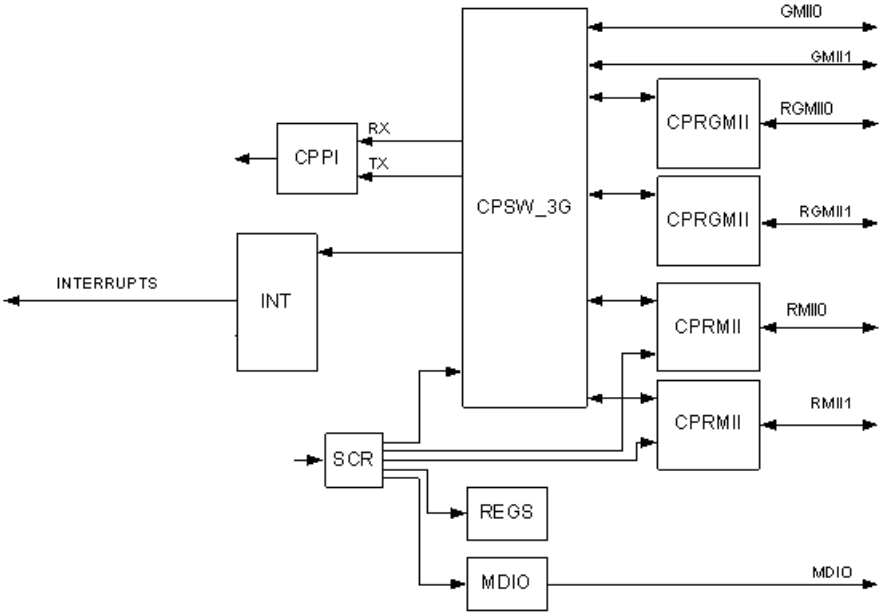
- Two Ethernet ports with Selectable G/MII, RMII, and RGMII interfaces
- Synchronous 10/100/1000 Mbit operation.
- Wire rate switching (802.1d)
- Non Blocking switch fabric
- Flexible logical FIFO based packet buffer structure
- Four priority level QOS support (802.1p)
- Support for Audio/Video Bridging (P802.1Qav/D6.0)
- Support for IEEE 1588 Clock Synchronization
- Reset isolation
- Communications port programming interface (CPPI) 3.1 compliant DMA controllers
- Maximum frame size 2016 bytes (2020 with VLAN)
- Programmable priority escalation on transmit
- Flow Control Support (802.3x)
- MDIO module for PHY Management
- Address Lookup
  - 1024 total address lookup engine (ALE) entries of VLANs and/or MAC addresses
  - L2 address lock and L2 filtering support
  - Multicast/broadcast filtering and forwarding state control
  - Receive-based or destination-based multicast and broadcast rate limits
  - MAC address blocking
  - Source port locking
  - OUI (Vendor ID) host accept/deny feature
  - Host controlled time-based aging
  - MAC authentication (802.1x)
  - Remapping of priority level of VLAN or ports
  - Multiple spanning tree support (spanning tree per VLAN)
- VLAN support
  - 802.1Q compliant
    - Auto add port VLAN for untagged frames on ingress
    - Auto VLAN removal on egress and auto pad to minimum frame size
- Ethernet Statistics:
  - EtherStats and 802.3 Stats RMON statistics gathering (shared)
  - Programmable statistics interrupt mask when a statistic is above one half its 32-bit value
- CPGMAC_SL transmit to CPGMAC_SL receive Loopback mode (digital loopback) supported
- CPGMAC_SL receive to CPGMAC_SL transmit Loopback mode (FIFO loopback) supported
- Programmable interrupt control with selected interrupt pacing
- Programmable Transmit Inter Packet Gap (IPG)
- Half duplex mode supported only in 10/100 Mbps mode.


Copyright © 2011–2016, Texas Instruments Incorporated

- Full duplex mode supported in 10/100/1000 Mbps mode
- 8k (2048 × 32) internal CPPI buffer descriptor memory
- Emulation support
- IEEE 802.3 gigabit Ethernet conformant

### 9.1.2 Functional Block Diagram

**Figure 9-1. 3PSW Ethernet Subsystem Top Level Block Diagram**




### 9.1.3 Interface Mode Selection

The 3 port switch (3PSW) Ethernet Subsystem has two 10/100/1000 Ethernet ports with selectable G/MII, RMII, and RGMII interfaces.

The interface mode is selected by configuring the MII mode selection register (GMII_SEL) in the *Control Module* chapter.

### 9.1.4 Subsystem Signal Summary

The tables below provide the signals names for different Interfaces (RGMII, G/MII, RMII). These signals are pin muxed. For more details on the signal descriptions/connections please refer to section of Ethernet Mac Sliver in this document.

See the pin mux selection details in the device data manual for configuring the PINMUX as per the Interface (RGMII, G/MII, RMII) selected.

**Table 9-1. G/MII Signal Description**

| Signal name | Width | Input/Output | Description |
|---|---|---|---|
| EMAC[0]_MtCLK | 1 | I | GMII0 transmit clock |
| EMAC[0]_Mtxd[0-7] | 8 | O | GMII0 transmit data |
| EMAC[0]_Mtxen | 1 | O | GMII0 transmit data enable |
| EMAC[0]_gMTclk | 1 | O | GMII0 source synchronous transmit clock |
| EMAC[0]_Mcol | 1 | I | GMII0 collision (asynchronous) |
| EMAC[0]_Mcrs | 1 | I | GMII0 carrier sense (asynchronous) |
| EMAC[0]_MRClk | 1 | I | GMII0 receive clock |
| EMAC[0]_Mrxd[0-7] | 8 | I | GMII0 receive data |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-1. G/MII Signal Description (continued)**

| Signal name | Width | Input/Output | Description |
|---|---|---|---|
| EMAC[0]_Mrxdv | 1 | I | GMII0 receive data valid |
| EMAC[0]_Mrxer | 1 | I | GMII0 receive data error |
| EMAC[1]_MtCLK | 1 | I | GMII1 transmit clock |
| EMAC[1]_Mtxd[0-7] | 8 | O | GMII1 transmit data |
| EMAC[1]_Mtxen | 1 | O | GMII1 transmit data enable |
| EMAC[1]_gMTclk | 1 | O | GMII1 source synchronous transmit clock |
| EMAC[1]_Mcol | 1 | I | GMII1 collision (asynchronous) |
| EMAC[1]_Mcrs | 1 | I | GMII1 carrier sense (asynchronous) |
| EMAC[1]_MRClk | 1 | I | GMII1 receive clock |
| EMAC[1]_Mrxd[0-7] | 8 | I | GMII1 receive data |
| EMAC[1]_Mrxdv | 1 | I | GMII1 receive data valid |
| EMAC[1]_Mrxer | 1 | I | GMII1 receive data error |

**Table 9-2. RMII Signal Description**

| Signal name | Width | Input/Output | Description |
|---|---|---|---|
| EMAC_RMREFCLK | 1 | IO | RMII0/1 reference clock |
| EMAC[0]_RMTXD[0-1] | 2 | O | RMII0 transmit data |
| EMAC[0]_RMTXEN | 1 | O | RMII0 transmit data enable |
| EMAC[0]_RMCRSDV | 1 | I | RMII0 carrier sense and data valid |
| EMAC[0]_RMRXD[0-1] | 2 | I | RMII0 receive data |
| EMAC[0]_RMRXER | 1 | I | RMII0 receive error |
| EMAC[1]_RMTXD[0-1] | 2 | O | RMII1 transmit data |
| EMAC[1]_RMTXEN | 1 | O | RMII1 transmit data enable |
| EMAC[1]_RMCRSDV | 1 | I | RMII1 carrier sense and data valid |
| EMAC[1]_RMRXD[0-1] | 2 | I | RMII1 receive data |
| EMAC[1]_RMRXER | 1 | I | RMII1 receive error |

**Table 9-3. RGMII Signal Description**

| Signal name | Width | Input/Output | Description |
|---|---|---|---|
| EMAC[0]_RGTXD[0-3] | 4 | O | RGMII0 Transmit Data |
| EMAC[0]_RGTXCTL | 1 | O | RGMII0 Transmit Control/Enable |
| EMAC[0]_RGTXC | 1 | O | RGMII0 Transmit Clock |
| EMAC[0]_RGRXD[0-3] | 4 | I | RGMII0 Receive Data |
| EMAC[0]_RGRXCTL | 1 | I | RGMII0 Receive Control |
| EMAC[0]_RGRXC | 1 | I | RGMII0 Receive Clock |
| EMAC[1]_RGTXD[0-3] | 4 | O | RGMII1 Transmit Data |
| EMAC[1]_RGTXCTL | 1 | O | RGMII1 Transmit Control/Enable |
| EMAC[1]_RGTXC | 1 | O | RGMII1 Transmit Clock |
| EMAC[1]_RGRXD[0-3] | 4 | I | RGMII1 Receive Data |
| EMAC[1]_RGRXCTL | 1 | I | RGMII1 Receive Control |
| EMAC[1]_RGRXC | 1 | I | RGMII1 Receive Clock |


Copyright © 2011–2016, Texas Instruments Incorporated

## 9.1.5 Clocking

### 9.1.5.1 Subsystem Clocking

The 3 port switch (3PSW) Ethernet Subsystem clocks are derived from SATA SERDES PLL.

The SATA SERDES provides reference clocks to the modules G/MII (125 MHz), RMII (50 MHz), RGMII (50 MHz) and RGMII (5 MHz) as shown in Figure 9-2.

The SATA SerDes PLL configuration registers are available in the device control module register space. The *Control Module* chapter has more details on SATA PLL Register configuration.

The 3PSW Ethernet clocking provides option to receive or output the RMII reference clock (EMAC_RMREFCLK) from the pin as shown in Figure 9-2. See the RMII _REFCLK_SRC register in the device Control Module to configure the clocking option of RMII.

1588 reference clock, CPTS_RFT_CLK (250 MHz) and RGMII (250 MHz) reference clocks are provided by any of the Video-0/Video-1/Video_clk2/Audio/L3 PLLs as shown in Figure 9-2. The PLL is used for generating the clocks for CPTS_RFT_CLK, RGMII shall be selected by configuring the RMII_REFCLK_SRC register in the device Control Module.

### 9.1.5.2 Interface Clocking

Data is transmitted and received with respect to the reference clocks of the interface pins.

#### *9.1.5.2.1 G/MII Interface Clocking*

EMAC [0/1] _MtCLK and EMAC [0/1] _MRCLK frequencies are fixed by the 802.3 specification.

- 2.5 MHz at 10 Mbps
- 25 MHz at 100 Mbps
- 125 MHz at 1000 Mbps (This is valid only for the EMAC[0/1]_MRCLK, the EMAC[0/1]_MTCLK is not used in gigabit/1000Mbps mode.)

EMAC [0/1] _gMTclk runs at 125 MHz in all the modes of operation.

#### *9.1.5.2.2 RGMII Interface Clocking*

RGMII_RXC, RGMII_TXC frequencies are:

- 2.5 MHz at 10 Mbps
- 25 MHz at 100 Mbps
- 125 MHz at 1000 Mbps

See the MII mode selection register (GMII_SEL) in the device Control Module to configure the CPRGMII Internal Delay/No Delay in Transmit Mode of Operation. See Section 9.2.1.8.3 for more details.

#### *9.1.5.2.3 RMII Interface Clocking*

RMII interface clock EMAC_RMREFCLK frequencies are:

- 50 MHz at 10 Mbps
- 50 MHz at 100 Mbps

See the RMII _REFCLK_SRC register in the device Control Module to configure the Clocking Option of RMII.

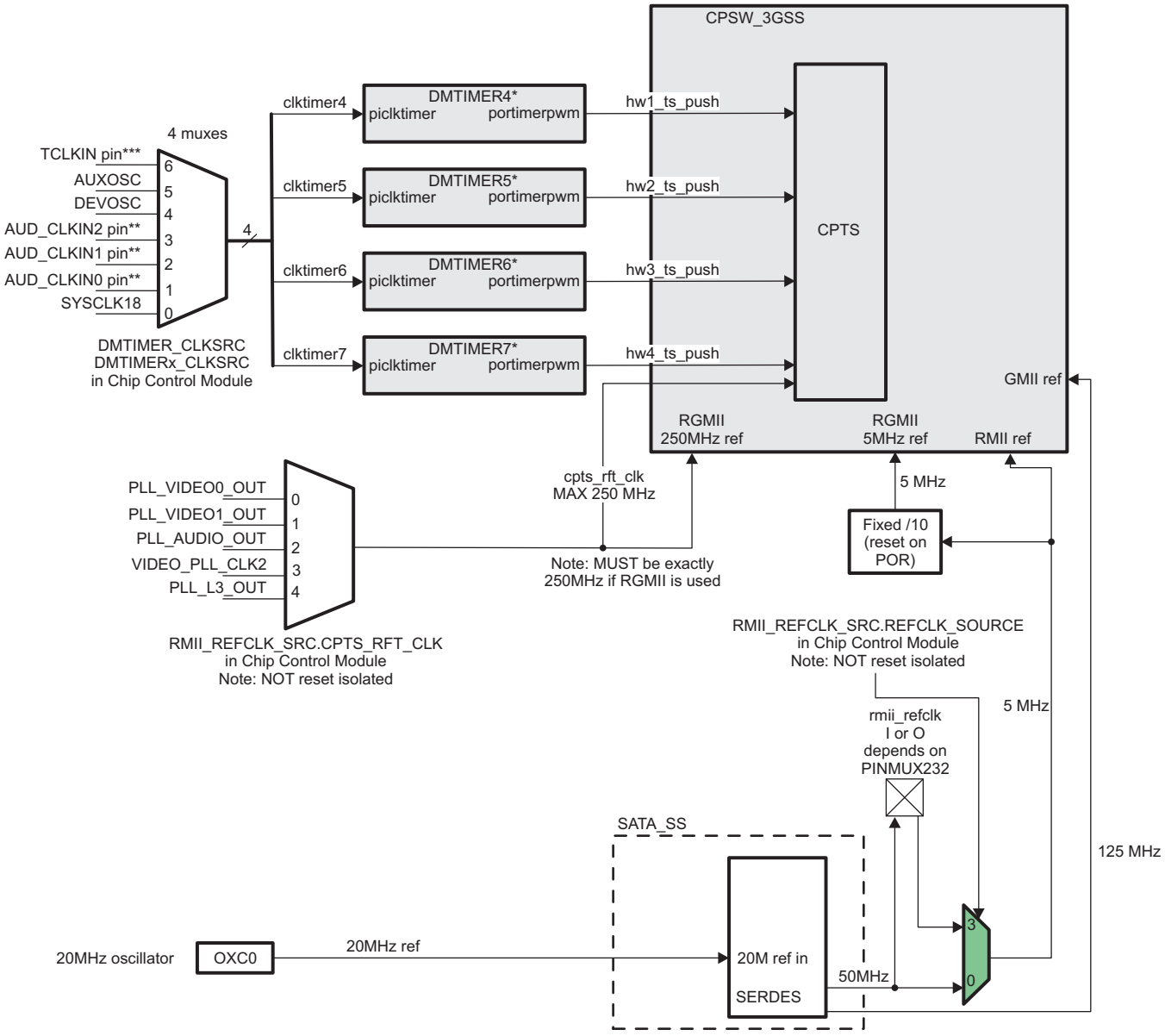
#### *9.1.5.2.4 MDIO Clocking*

The MDIO clock is based on a divide-down of the peripheral clock, running at 125 MHz.The application software or driver must control the divide-down value.

See the MDIO Control Register for configuring the Clock Divider (CLKDIV) value.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 9-2. 3PSW Subsystem Clocking Block Diagram**




### 9.1.6 Interrupt Support

3PSW Ethernet Subsystem has 4 Interrupts:

- RX_thresh_pulse - Receive Threshold Interrupt
- RX_pulse - Receive Interrupt
- TX_pulse - Transmit Interrupt
- Misc_pulse - Miscellaneous Interrupt.

See Section 9.2.5 for more details. See the device data manual for more details on the mapping of these interrupts at the device level.

### 9.1.7 Memory Map

The Addresses for the Modules in the 3 port switch (3PSW) Ethernet Subsystem are:


Copyright © 2011–2016, Texas Instruments Incorporated

- CPSW_3G Module: The start address of the CPSW_3G module is 4A10 0000h. See Section 9.4.1.
- MDIO Module: The start address of the MDIO module is 4A10 0800h. See Section 9.4.2.
- CPSW Subsystem Module: The start address of the CPSW subsystem is 4A10 0900h. See Section 9.4.3.
- CPPI Descriptors: The start address of the CPPI descriptors is 4A10 2000h.

  The 8K bytes of CPPI memory begin at address 4A10 2000h from the EMAC perspective. The buffer descriptors programmed to access the CPPI RAM memory from the CPU perspective should use the address range from 4A10 2000h to 4A10 4000h.

### 9.1.8 3PSW Ports

Ethernet Subsystem has 3 Ports. Port 0 is the Host port (internal to the Subsystem). Ports 1 and 2 are the external ports connected to G/MII, RGMII, or RMII interfaces as per the interface selected.

Naming conventions followed in this chapter:

- Port0 is referred to the Host Port
- Port1 is referred to the interfaces GMII0/RGMII0/RMII0
- Port2 is referred to the interfaces GMII1/RGMII1/RMII1


Copyright © 2011–2016, Texas Instruments Incorporated

## 9.2 CPSW Architecture

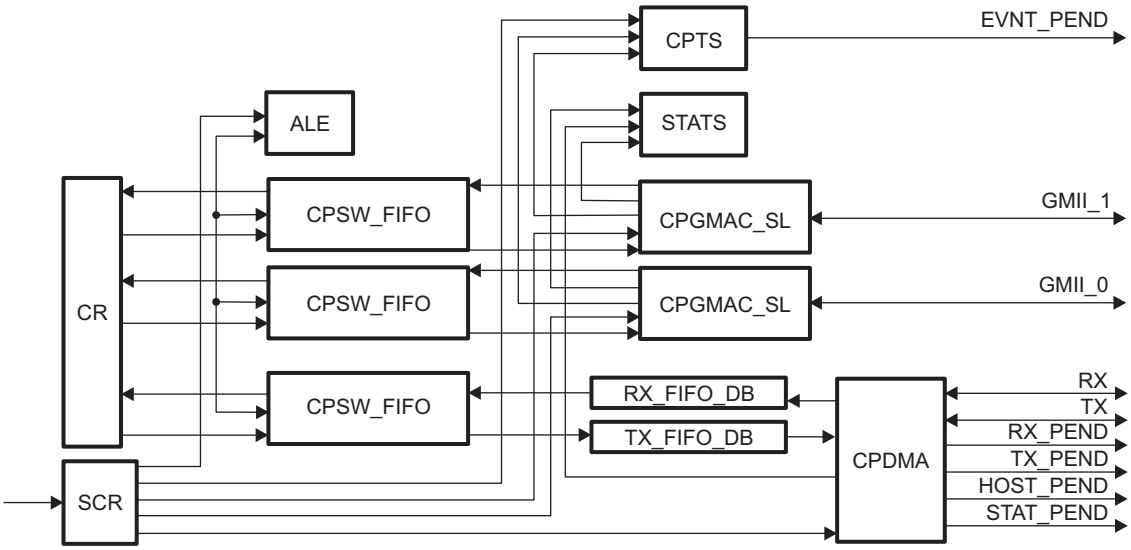
The 3 port switch (3PSW) Ethernet Subsystem peripheral are compliant to the IEEE Std 802.3 Specification. The 3PSW Ethernet Subsystem contains two RGMII/RMII interfaces, one CPPI 3.0 interface, Interrupt Controller, MDI0 and CPSW_3G which contains two G/MII interfaces as shown in Figure 9-3.

### *9.2.1 CPSW_3G*

The CPSW_3G G/MII interfaces are compliant to the IEEE Std 802.3 Specification.

The CPSW_3G contains two CPGMAC_SL interfaces (ports 1 and 2), one CPPI 3.0 interface Host Port (port 0), Common Platform Time Sync (CPTS), ALE Engine and CPDMA. A top-level block diagram of the CPSW_3G is shown in Figure 9-3.

**Figure 9-3. CPSW_3G Block Diagram**




#### 9.2.1.1 CPDMA RX and TX Interfaces

The CPDMA submodule is a CPPI 3.0 compliant packet DMA transfer controller. The CPPI 3.0 interface is port 0.

##### *9.2.1.1.1 Functional Operation*

After reset, initialization, and configuration the host may initiate transmit operations. Transmit operations are initiated by host writes to the appropriate transmit channel head descriptor pointer contained in the STATERAM block. The transmit DMA controller then fetches the first packet in the packet chain from memory in accordance with CPPI 3.0 protocol. The DMA controller writes the packet into the external transmit FIFO in 64-byte bursts (maximum).

Receive operations are initiated by host writes to the appropriate receive channel head descriptor pointer after host initialization and configuration. The receive DMA controller writes the receive packet data to external memory in accordance with CPPI 3.0 protocol. See the CPPI Buffer Descriptors section for detailed description of Buffer Descriptors.

##### *9.2.1.1.2 Receive DMA Interface*

The receive DMA is an eight channel CPPI 3.0 compliant interface. Each channel has a single queue for frame reception.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.1.1.2.1 Receive DMA Host Configuration

To configure the RX DMA for operation the software must perform the following:

1. Initialize the receive addresses.
2. Initialize the RX_HDP registers to 0.
3. Enable the desired receive interrupts in the INTMASK register.
4. Write the RX_BUFFER_OFFSET register value.
5. Setup the receive channel(s) buffer descriptors in host memory as required by CPPI 3.0.
6. Enable the RX DMA controller by setting the RX_EN bit in the RX_CONTROL register.

#### 9.2.1.1.2.2 Receive Channel Teardown

The host commands a receive channel teardown by writing the channel number to the RX_TEARDOWN register. When a teardown command is issued to an enabled receive channel the following will occur:

- Any current frame in reception will complete normally.
- The teardown_complete bit will be set in the next buffer descriptor in the chain (if there is one).
- The channel head descriptor pointer will be cleared to 0.
- A receive interrupt for the channel will be issued to the host.
- The software should acknowledge a teardown interrupt with a FFFF FFFCh Acknowledge value.

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by a set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a FFFF FFFCh acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be FFFF FFFCh if the interrupt was due to a teardown command.

### 9.2.1.1.3 Transmit DMA Interface

The tranmsit DMA is an eight channel CPPI 3.0 compliant interface. Priority between the eight queues may be either fixed or round robin as selected by the TX_PTYPE bit in the DMACONTROL register. If the priority type is fixed, then channel 7 has the highest priority and channel 0 has the lowest priority. Round robin priority proceeds from channel 0 to channel 7.

#### 9.2.1.1.3.1 Transmit DMA Host Configuration

To configure the TX DMA for operation the software must do the following:

1. Initialize the TX_HDP registers to 0.
2. Enable the desired transmit interrupts in the INTMASK register.
3. Setup the transmit channel(s) buffer descriptors in host memory as defined in CPPI 3.0.
4. Configure and enable the transmit operation as desired in the TX_CONTROL register.
5. Write the appropriate TX_HDP registers with the appropriate values to start transmit operations.

#### 9.2.1.1.3.2 Transmit Channel Teardown

The host commands a transmit channel teardown by writing the channel number to the TX_TEARDOWN register. When a teardown command is issued to an enabled transmit channel the following will occur:

- Any frame currently in transmission will complete normally
- The teardown complete bit will be set in the next sop buffer descriptor (if there is one).
- The channel head descriptor pointer will be cleared to 0.
- An interrupt will be issued to inform the host of the channel teardown.
- The software should acknowledge a teardown interrupt with a FFFF FFFCh acknowledge value


Copyright © 2011–2016, Texas Instruments Incorporated

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by the set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a FFFF FFFCh acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be FFFF FFFCh if the interrupt was due to a teardown command.

#### *9.2.1.1.4 Transmit Rate Limiting*

Transmit operations can be configured to rate limit the transmit data for each transmit priority. Rate limiting is enabled for a channel when the TX_RLIM bit associated with that channel is set in the DMACONTROL register. Rate limited channels must be the highest priority channels. For example, if two rate limited channels are required then TX_RLIM should be set to 11000000b with the MSB corresponding to channel 7. When any channels are configured to be rate-limiting, the priority type must be fixed for transmit. Round-robin priority type is not allowed when rate-limiting. Each of the eight transmit priorities has an associated register to control the rate at which the priority is allowed to send data (TX_PRI(0..7)_RATE) when the channel is rate-limiting. Each priority has a send count (PRI(0..7)_SEND_CNT) and an idle count (PRI(0..7)_IDLE_CNT). The transfer rate includes the inter-packet gap (12 bytes) and the preamble (8 bytes). The rate in Mbits/second that each priority is allowed to send is controlled by the equation:

*Priority Transfer rate in Mbit/s = ((PRI_IDLE_CNT/(PRI_IDLE_CNT + PRI_SEND_CNT)) × frequency × 32*

Where the frequency is the CLK frequency.

#### *9.2.1.1.5 Command IDLE*

The CMD_IDLE bit in the DMACONTROL register allows CPDMA operation to be suspended. When the idle state is commanded, the CPDMA will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For transmission, any complete or partial frame in the tx cell FIFO will be transmitted. For receive, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Commanded idle is similar in operation to emulation control and clock stop.

### 9.2.1.2 Address Lookup Engine (ALE)

The address lookup engine (ALE) processes all received packets to determine which port(s) if any that the packet should the forwarded to. The ALE uses the incoming packet received port number, destination address, source address, length/type, and VLAN information to determine how the packet should be forwarded. The ALE outputs the port mask to the switch fabric that indicates the port(s) the packet should be forwarded to. The ALE is enabled when the ENABLE_ALE bit in the ALE_CONTROL register is set. All packets are dropped when the ENABLE_ALE bit is cleared to 0.

In normal operation, the CPGMAC_SL modules are configured to issue an abort, instead of an end of packet, at the end of a packet that contains an error (runt, frag, oversize, jabber, crc, alignment, code etc.) or at the end of a mac control packet. However, when the CPGMAC_SL configuration bit(s) cef, csf, or cmf are set, error frames, short frames or mac control frames have a normal end of packet instead of an abort at the end of the packet. When the ALE receives a packet that contains errors (due to a set header error bit), or a mac control frame and does not receive an abort, the packet will be forwarded only to the host port (port 0). No ALE learning occurs on packets with errors or mac control frames. Learning is based on source address and lookup is based on destination address.

The ALE may be configured to operate in bypass mode by setting the ALE_BYPASS bit in the ALE_CONTROL register. When in bypass mode, all CPGMAC_SL received packets are forwarded only to the host port (port 0). Packets from the two ports can be on separate RX DMA channels by configuring the CPDMA_RX_CH_MAP register. In bypass mode, the ALE processes host port transmit packets the same as in normal mode. In general, packets would be directed by the host in bypass mode.


Copyright © 2011–2016, Texas Instruments Incorporated

The ALE may be configured to operate in OUI deny mode by setting the ENABLE_OUI_DENY bit in the ALE_CONTROL register. When in OUI deny mode, a packet with a non-matching OUI source address will be dropped unless the destination address matches a multicast table entry with the super bit set. Broadcast packets will be dropped unless the broadcast address is entered into the table with the super bit set. Unicast packets will be dropped unless the unicast address is in the table with block and secure both set (supervisory unicast packet).

Multicast supervisory packets are designated by the super bit in the table entry. Unicast supervisory packets are indicated when block and secure are both set. Supervisory packets are not dropped due to rate limiting, OUI, or VLAN processing.

#### *9.2.1.2.1 Address Table Entry*

The ALE table contains 1024 entries. Each table entry represents a free entry, an address, a VLAN, an address/VLAN pair, or an OUI address. Software should ensure that there are not double address entries in the table. The double entry used would be indeterminate. Reserved table bits must be written with zeroes.

Source Address learning occurs for packets with a unicast, multicast or broadcast destination address and a unicast or multicast (including broadcast) source address. Multicast source addresses have the group bit (bit 40) cleared before ALE processing begins, changing the multicast source address to a unicast source address. A multicast address of all ones is the broadcast address which may be added to the table. A learned unicast source address is added to the table with the following control bits:

**Table 9-4. Learned Address Control Bits**

| | |
|---|---|
| unicast_type | 11 |
| Block | 0 |
| Secure | 0 |

If a received packet has a source address that is equal to the destination address then the following occurs:

- The address is learned if the address is not found in the table.
- The address is updated if the address is found.
- The packet is dropped.

**Table Entry Type**

00 - Free Entry

01 - Address Entry : unicast or multicast determined by destination **address bit 40 .**

10 - VLAN entry

11 - VLAN Address Entry : unicast or multicast determined by **address bit 40.**

##### *9.2.1.2.1.1 Free Table Entry*

**Table 9-5. Free (Unused) Address Table Entry Bit Values**

| 71-62 | 61-60 | 59-0 |
|---|---|---|
| Reserved | ENTRY_TYPE(00) | Reserved |

##### *9.2.1.2.1.2 Multicast Address Table Entry*

**Table 9-6. Multicast Address Table Entry Bit Values**

| 71-69 | 68-66 | 65 | 64 | 63-62 | 61-60 | 59-48 | 47-0 |
|---|---|---|---|---|---|---|---|
| Reserved | PORT_MASK | SUPER | Reserved | MCAST_FWD_ STATE | ENTRY_TYPE (01) | Reserved | MULTICAST_ ADDRESS |



Copyright © 2011–2016, Texas Instruments Incorporated

**Supervisory Packet (SUPER)**

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

**Port Mask(2:0) (PORT_MASK)**

This 3-bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

**Multicast Forward State (MCAST_FWD_STATE)**

Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit PORT_MASK has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 - Forwarding

01 - Blocking/Forwarding/Learning

10 - Forwarding/Learning

11 - Forwarding

The forward state test returns a true value if both the RX and TX ports are in the required state.

**Table Entry Type (ENTRY_TYPE)**

Address entry type. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

**Packet Address (MULTICAST_ADDRESS)**

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

#### *9.2.1.2.1.3 VLAN/Multicast Address Table Entry*

**Table 9-7. VLAN/Multicast Address Table Entry Bit Values**

| 71-69 | 68-66 | 65 | 64 | 63-62 | 61-60 | 59-48 | 47-0 |
|---|---|---|---|---|---|---|---|
| Reseved | PORT_MASK | SUPER | Reserved | MCAST_FWD_STATE | ENTRY_TYPE (11) | VLAN_ID | MULTICAST_ADDRESS |

**Supervisory Packet (SUPER)**

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

**Port Mask(2:0) (PORT_MASK)**

This 3-bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

**Multicast Forward State (MCAST_FWD_STATE)**


Copyright © 2011–2016, Texas Instruments Incorporated

Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit PORT_MASk has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 - Forwarding

01 - Blocking/Forwarding/Learning

10 - Forwarding/Learning

11 - Forwarding

The forward state test returns a true value if both the RX and TX ports are in the required state.

**Table Entry Type (ENTRY_TYPE)**

Address entry type. Unicast or multicast determined by address bit 40.

11: VLAN address entry. Unicast or multicast determined by address bit 40.

**VLAN ID (VLAN_ID)**

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

**Packet Address (MULTICAST_ADDRESS)**

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

#### *9.2.1.2.1.4 Unicast Address Table Entry*

**Table 9-8. Unicast Address Table Entry Bit Values**

| 71-68 | 67-66 | 65 | 64 | 63-62 | 61-60 | 59-48 | 47-0 |
|---|---|---|---|---|---|---|---|
| Reseved | PORT_ NUMBER | BLOCK | SECURE | UNICAST_ TYPE (00) or (X1) | ENTRY_TYPE (01) | Reserved | UNICAST_ ADDRESS |

**Port Number (PORT_NUMBER)**

This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).

**Block (BLOCK)**

The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 - Address is not blocked.

1 - Drop a packet with a matching source or destination address (secure

must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

**Secure (SECURE)**

This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry port_number.

0 - Received port number is a don't care.

1 - Drop the packet if the received port is not the secure port for the source


Copyright © 2011–2016, Texas Instruments Incorporated

address and do not update the address (block must be zero)

**Unicast Type (UNICAST_TYPE)**

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

**Table Entry Type (ENTRY_TYPE)**

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

**Packet Address (UNICAST_ADDRESS)**

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

#### *9.2.1.2.1.5 OUI Unicast Address Table Entry*

**Table 9-9. OUI Unicast Address Table Entry Bit Values**

| 71-64 | 63-62 | 61-60 | 59-48 | 47-24 | 23-0 |
|---|---|---|---|---|---|
| Reserved | UNICAST_TYPE(10) | ENTRY_TYPE(01) | Reserved | UNICAST_OUI | Reserved |

**Unicast Type (UNICAST_TYPE)**

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

**Table Entry Type (ENTRY_TYPE)**

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

**Packet Address (UNICAST_OUI)**

For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup.

#### *9.2.1.2.1.6 VLAN/Unicast Address Table Entry*

**Table 9-10. Unicast Address Table Entry Bit Values**

| 71:68 | 67:66 | 65 | 64 | 63:62 | 61:60 | 59:48 | 47:0 |
|---|---|---|---|---|---|---|---|
| Reseved | PORT_ NUMBER | BLOCK | SECURE | UNICAST_ TYPE (00) or (X1) | ENTRY_TYPE (11) | VLAN_ID | UNICAST_ ADDRESS |


Copyright © 2011–2016, Texas Instruments Incorporated

**Port Number (PORT_NUMBER)**

This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).]

**Block (BLOCK)**

The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 - Address is not blocked.

1 - Drop a packet with a matching source or destination address (secure must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

**Secure (SECURE)**

This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry PORT_NUMBER.

0 - Received port number is a don't care.

1 - Drop the packet if the received port is not the secure port for the source

address and do not update the address (block must be zero)

**Unicast Type (UNICAST_TYPE)**

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

**Table Entry Type (ENTRY_TYPE)**

Address entry. Unicast or multicast determined by address bit 40.

11: VLAN address entry. Unicast or multicast determined by address bit 40.

**VLAN ID (VLAN_ID)**

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

**Packet Address (UNICAST_ADDRESS)**

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

#### *9.2.1.2.1.7 VLAN Table Entry*

**Table 9-11. VLAN Table Entry**

| 71-62 | 61-60 | 59-48 | 47-27 | 26-24 | 23-19 | 18-16 | 15-11 | 10-8 | 7-3 | 2-0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | ENTRY_ TYPE(10) | VLAN_ID | Reserved | FORCE_ UNTAGGED_ EGRESS | Reserved | REG_ MCAST_ FLOOD_MASK | Reserved | UNREG_MCAST_ FLOOD_MASK | Reserved | VLAN_ MEMBER_ LIST |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table Entry Type (ENTRY_TYPE)**

10: VLAN entry

**VLAN ID (VLAN_ID)**

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

**Force Untagged Packet Egress (FORCE_UNTAGGED_EGRESS)**

This field causes the packet VLAN tag to be removed on egress (except on port 0).

**Registered Multicast Flood Mask (REG_MCAST_FLOOD_MASK)**

Mask used for multicast when the multicast address is found.

**Unregistered Multicast Flood Mask (UNREG_MCAST_FLOOD_MASK)**

Mask used for multicast when the multicast address is not found.

**VLAN Member List (VLAN_MEMBER_LIST)**

This 3-bit field indicates which port(s) are members of the associated VLAN.

#### *9.2.1.2.2 Packet Forwarding Processes*

There are four processes that an incoming received packet may go through to determine packet forwarding. The processes are Ingress Filtering, VLAN_Aware Lookup, VLAN_Unaware Lookup, and Egress.

Packet processing begins in the Ingress Filtering process. Each port has an associated packet forwarding state that can be one of four values (Disabled, Blocked, Learning, or Forwarding). The default state for all ports is Disabled. The host sets the packet forwarding state for each port. The receive packet processes

In the packet ingress process (receive packet process), there is a forward state test for unicast destination addresses and a forward state test for multicast addresses. The multicast forward state test indicates the port states required for the receiving port in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state for the packet to be forwarded for transmission. The mcast_fwd_state indicates the required port state for the receiving port as indicated in the preceding table. The unicast forward state test indicates the port state required for the receiving port in order to forward the unicast packet. The transmit port must be in the Forwarding state in order to forward the packet. The block and secure bits determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state. The transmit port must be in the Forwarding state regardless. The forward state test used in the ingress process is determined by the destination address packet type (multicast/unicast).

In general, packets received with errors are dropped by the address lookup engine without learning, updating, or touching the address. The error condition and the abort are indicated by the CPGMAC_SL to the ALE. Packets with errors may be passed to the host (not aborted) by a CPGMAC_SL port, if the port has the RX_CMF_EN, RX_CEF_EN, or RX_CSF_EN bit(s) set. Error packets that are passed to the host by the CPGMAC_SL are considered to be bypass packets by the ALE and are sent only to the host. Error packets do not learn, update, or touch addresses regardless of whether they are aborted or sent to the host. Packets with errors received by the host are forwarded as normal.

The following control bits are in the SL1/2_MACCONTROL register.

**RX_CEF_EN -**This CPGMAC_SL control bit enables frames that are fragments, long, jabber, CRC, code, and alignment errors to be forwarded.

**RX_CSF_EN -**This CPGMAC_SL bit enables short frames to be forwarded.

**RX_CMF_EN -**This CPGMAC_SL control bit enables mac control frames to be forwarded.

#### *9.2.1.2.3 Learning Process*

The learning process is applied to each receive packet that is not aborted. The learning process is a concurrent process with the packet forwarding process.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.1.2.4 VLAN Aware Mode

The CPSW_3G is in VLAN aware mode when the VLAN_AWARE bit is set in the CPSW_CONTROL register. In VLAN aware mode, ports 0 receive packets (out of the CPSW_3G) may or may not be VLAN encapsulated depending on the RX_VLAN_ENCAP bit in the CPSW_CONTROL register. Port 0 receive packet data is never modified. VLAN is not removed regardless of the force untagged egress bit for Port 0. VLAN encapsulated receive packets have a 32-bit VLAN header encapsulation word added to the packet data. VLAN encapsulated packets are specified by a set RX_VLAN_ENCAP bit in the packet buffer descriptor.

Port 0 transmit packets are never VLAN encapsulated (encapsulation is not allowed).

In VLAN aware mode, transmitted packet data is changed depending on the packet type (pkt_type), packet priority (pkt_pri), and VLAN information.

#### 9.2.1.2.5 VLAN Unaware Mode

The CPSW_3G is in VLAN unaware mode when the VLAN_AWARE bit is cleared to 0 in the CPSW_CONTROL register. Port 0 receive packets (out of the CPSW_3G) may or may not be VLAN encapsulated depending on the RX_VLAN_ENCAP bit in the CPSW_CONTROL register. Port 0 transmit packets are never VLAN encapsulated.

### 9.2.1.3 Packet Priority Handling

Packets are received on three ports, two are CPGMAC_SL Ethernet ports and the third port is the CPPI host port. Received packets have a received packet priority (0 to 7, with 7 being the highest priority).

The received packet priority is the port priority for untagged packets, and the actual packet priority for priority tagged and VLAN tagged packets. The received packet priority is mapped through the receive ports associated packet priority to header packet priority mapping register to obtain the header packet priority (the CPDMA RX and TX nomenclature is reversed from the CPGMAC_SL nomenclature).

The header packet priority is mapped through the header priority to switch priority mapping register to obtain the hardware switch priority (0 to 3, with 3 being the highest priority). The header packet priority is then used as the actual transmit packet priority if the VLAN information is to be sent on egress.

### 9.2.1.4 FIFO Memory Control

Each of the three CPSW_3G ports has an identical associated FIFO. Each FIFO contains a single logical receive queue and four logical transmit queues (priority 0 through 3). Each FIFO memory contains 20,480 bytes (20k) total organized as 2560 by 64-bit words contained in a single memory instance. The FIFO memory is used for the associated port transmit and receive queues. The TX_MAX_BLKS field in the FIFOs associated MAX_BLKS register determines the maximum number of 1k FIFO memory blocks to be allocated to the four logical transmit queues (transmit total). The RX_MAX_BLKS field in the FIFO's associated MAX_BLKS register determines the maximum number of 1k memory blocks to be allocated to the logical receive queue. The TX_MAX_BLKS value plus the RX_MAX_BLKS value must sum to 20 (the total number of blocks in the FIFO). If the sum were less than 20, then some memory blocks would be unused. The default is 17 (decimal) transmit blocks and three receive blocks. The FIFOs follow the naming convention of the Ethernet ports. Host Port is Port0 and External Ports are Port1 and Port2.

### 9.2.1.5 FIFO Transmit Queue Control

There are four transmit queues in each transmit FIFO. Software has some flexibility in determining how packets are loaded into the queues and on how packet priorities are selected for transmission (how packets are removed and transmitted from queues). All ports on the switch have identical FIFO's. For the purposes of the below the transmit FIFO is switch egress even though the port 0 transmit FIFO is connected to the CPDMA receive (also switch egress). The CPDMA nomenclature is reversed from the CPGMAC_SL nomenclature due to legacy reasons.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.1.5.1 Normal Priority Mode

When operating in normal mode, lower priority frames are dropped before higher priority frames. The intention is to give preference to higher priority frames. Priority 3 is the highest priority and is allowed to fill the FIFO. Priority 2 will drop packets if the packet is going to take space in the last 2k available. Priority 1 will drop packets if the packet is going to take space in the last 4k available. Priority 0 will drop packets if the packet is going to take space in the last 6k available. If fewer than 4 priorities are to be implemented then the priorities should be mapped such that the highest priorities are used. For example, if two priorities are going to be used then all packets should be mapped to priorities 3 and 2 and priorities 1 and 0 should be unused. Priority escalation may be used in normal priority mode if desired. Normal priority mode is configured as described below:

- Select normal priority mode by setting TX_IN_SEL = 00 for all ports (default value in P0/1/2_TX_IN_CTL)
- Configure priority mapping to use only the highest priorities if less than 4 priorities are used. Refer to Section 9.2.1.3.

#### 9.2.1.5.2 Dual Mac Mode

When operating in dual mac mode the intention is to transfer packets between ports 0 and 1 and ports 0 and 2, but not between ports 1 and 2. Each CPGMAC_SL appears as a single MAC with no bridging between MAC's. Each CPGMAC_SL has at least one unique (not the same) mac address.

Dual mac mode is configured as described below:

- Set the ALE_VLAN_AWARE bit in the ALE_CONTROL register. This bit configures the ALE to process in VLAN aware mode. The CPSW_3G VLAN aware bit (VLAN_AWARE in CPSW_CONTROL) determines how packets VLAN's are processed on CPGMAC_SL egress and does not affect how the ALE processes packets or the packet destination. The CPSW_3G VLAN aware bit may be set or not as required (must be set, if VLAN's are to exit the switch).
- Configure the Port 1 to Port 0 VLAN
  - Add a VLAN Table Entry with ports 0 and 1 as members (clear the flood masks).
  - Add a VLAN/Unicast Address Table Entry with the Port1/0 VLAN and a port number of 0. Packets received on port 1 with this unicast address will be sent only to port 0 (egress). If multiple mac addresses are desired for this port then multiple entries of this type may be configured.
- Configure the Port 2 to Port 0 VLAN
  - Add a VLAN Table Entry with ports 0 and 2 as members (clear the flood masks).
  - Add a VLAN/Unicast Address Table Entry with the Port2/0 VLAN and a port number of 0. Packets received on port 2 with this unicast address will be sent only to port 0 (egress). If multiple mac addresses are desired for this port then multiple entries of this type may be configured.
- Packets from the host (port 0) to ports 1 and 2 should be directed. If directed packets are not desired then VLAN with addresses can be added for both destination ports.
- Select the dual mac mode on the port 0 FIFO by setting TX_IN_SEL = 01 in P0_TX_IN_CTL. The intention of this mode is to allow packets from both ethernet ports to be written into the FIFO without one port starving the other port.
- The priority levels may be configured such that packets received on port 1 egress on one CPDMA RX channel while packets received on port 2 egress on a different CPDMA RX channel.

### 9.2.1.6 Audio Video Bridging

IEEE802.1BA is the standard for networked Audio Video Bridging (AVB) Systems. It defines the concept of an AVB domain as the intersection of a time domain and a stream reservation domain. The time domain is established based on the IEEE 802.1AS standard, wherein each node must support the generalized Precision Time Protocol (gPTP). Typically this takes the form of a state machine running on the host processor (such as the MPU) which, among other duties, processes the associated ethernet packets carrying timing information. A necessary input is the accurate time stamping of ethernet packets. This process is facilitated on SoC by the Common Platform Time Sync (CPTS), which is able to time stamp ingress and egress packets at the MAC layer with sufficient precision and accuracy to meet the AVB standard (refer to *Common Platform Time Sync (CPTS)*). A stream reservation domain is established


Copyright © 2011–2016, Texas Instruments Incorporated

between nodes, primarily through the use of two mechanisms: the Stream Reservation Protocol (SRP) and Forwarding and Queuing of Time Sensitive Streams (FQTSS). Both mechanisms are specified in the IEEE802.1Q standard (clauses 35 and 34, respectively). SRP is a software component which normally runs on the host processor and requires no hardware assistance. On the other hand, FQTSS has significant performance demands, as it requires the shaping of egress packets at small time scales. The SoC CPSW facilitates this process through the conjunction of rate-limited CPDMA channels and egress traffic shaping (refer to *CPSW 3G* for more details).

#### 9.2.1.7 Hardware Example Usage

##### *9.2.1.7.1 Rate Limit Mode*

Rate-limit mode is intended to allow some CPDMA transmit (switch ingress) channels and some CPGMAC_SL FIFO priorities (switch egress) to be rate-limited. Non rate-limited traffic (bulk traffic) is allowed on non rate-limited channels and FIFO priorities. The bulk traffic does not impact the rate-limited traffic. Rate-limited traffic must be configured to be sent to rate-limited queues (via packet priority handling). The allocated rates for rate-limited traffic must not be oversubscribed. For example, if port 1 is sending 15% rate limited traffic to port 2 priority 3, and port 0 is also sending 10% rate-limited traffic to port 2 priority 3, then the port 2 priority 3 egress rate must be configured to be 25% plus a percent or two for margin. The switch must be configured to allow some percentage of non rate-limited traffic. Non rate-limited traffic must be configured to be sent to non rate-limited queues. No packets from the host should be dropped, but non rate-limited traffic received on an ethernet port can be dropped. Rate-limited mode is configured as shown below:

1. Set TX_IN_SEL = 10 in P1/2_TX_IN_CTL to enable ports 1 and 2 transmit FIFO inputs to be configured for rate-limiting queues. Enabling a queue to be rate-limiting with this field affects only the packet being loaded into the FIFO, it does not configure the transmit for queue shaping.
2. Configure the number of rate-limited queues for port 1 and 2 transmit FIFO's by setting the TX_RATE_EN field in P1/2_TX_IN_CTL. Rate limited queues must be the highest number. For example, if there are two rate limited queues then 1100 would be written to this field for priorities 3 and 2. This field enables the FIFO to allow rate-limited traffic into rate-limited queues while discriminating against non rate-limited queues.
3. Set P1_PRIN_SHAPE_EN and P2_PRIN_SHAPE_EN in the CPSW_3G PTYPE register. These bits determine which queues actually shape the output data stream. In general, the same priorities that are set in TX_RATE_EN are set in these bits as well, but the FIFO input and output enable bits are separate to allow rate-limiting from the host to non shaped channels if desired. When queue shaping is not enabled for a queue then packets are selected for egress based on priority. When queue shaping is enabled then packets are selected for egress based on queue percentages. If shaping is required on a single queue then it must be priority 3 (priorities 2, 1 and 0 are strict priority). If shaping is required on two queues then it must be on priorities 2 and 3 (priorities 1 and 0 are strict priority). If shaping is required on three queues then it must be priorities 3, 2, and 1 (priority 0 would then get the leftovers). Priority shaping follows the requirements in the IEEE P802.1Qav/D6.0 specification. Priority shaping is not compatible with priority escalation (escalation must be disabled).
4. P0_TX_IN_CTL[17:16] TX_IN_SEL should be set to 00, so Port 0 egress (CPDMA RX) is not rate-limited.
5. The CPDMA is configured for rate-limited transmit (switch ingress) channels by setting the highest bits of the TX_RLIM field in the CPDMA DMACONTROL register. If there are two rate-limited channels, then TX_RLIM = 11000000 (the rate limited channels must be the highest priorities). Also, the TX_PTYPE bit in the DMACONTROL register must be set (fixed priority mode). Rate-limited channels must go to rate-limited FIFO queues, and the FIFO queue rate must not be oversubscribed.

##### *9.2.1.7.2 Configuring the device for 802.1Qav operation:*

There is no dedicated register-set to be configured for the time-sensitive stream handling. The list of functional features of CPSW_3G that will have to be configured are:

- DESCRIPTORS and CHANNEL CONFIGURATIONS:
  - CPPI TX and RX descriptors
  - VLAN and Priority tags


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-12. Example of TX Configuration**

| TX DMA CHANNEL | Packet Priority | Switch Queue Priority |
|---|---|---|
| 7 | 7 | 3 |
| 6 | 5 | 2 |
| 5 | 3 | 1 |
| 4 | 1 | 0 |

**Table 9-13. Example of RX Configuration**

| RX DMA CHANNEL | Packet Priority | Switch Queue Priority |
|---|---|---|
| 0 | 7 | 0 |
| 0 | 5 | 0 |
| 0 | 3 | 0 |
| 0 | 1 | 0 |

- ALE Configuration:
  - ALE in VLAN-ware mode, Non-ALE in bypass mode.

**Table 9-14. Example of Rate-limit Configurations**

| Register | Value | DESCRIPTION |
|---|---|---|
| CPSW_PTYPE | 0006 0000h | For Port1 -- 2 highest priority channels. |
| P1_TX_IN_CTL:<br>TX_RATE_EN<br>TX_IN_SEL | <br>1100<br>10 | <br>2 highest priority channels are rate limited<br>Rate limit mode |
| P1_SEND_PERCENT | 14 3E00h | 20% PRI7, 62% PRI5 |
| DMACONTROL | C001h | Chan7, Chan6 are Rate Limited. Round-robin selection of DMA channel. |
| TX_PRI7_RATE | 10013h | 200 Mbps |
| TX_PRI6_RATE | 10005h | ~600 Mbps |

### 9.2.1.8 Ethernet Mac Sliver (CPGMAC_SL)

The CPGMAC_SL peripheral shall be compliant to the IEEE Std 802.3 Specification. Half duplex mode is supported in 10/100 Mbps mode, but not in 1000 Mbps (gigabit) mode.

Features:

- Synchronous 10/100/1000 Mbit operation.
- G/MII Interface.
- Hardware Error handling including CRC.
- Full Duplex Gigabit operation (half duplex gigabit is not supported).
- EtherStats and 802.3Stats RMON statistics gathering support for external statistics collection module.
- Transmit CRC generation selectable on a per channel basis.
- Emulation Support.
- VLAN Aware Mode Support.
- Hardware flow control.
- Programmable Inter Packet Gap (IPG)

#### *9.2.1.8.1 G/MII Media Independent Interface*

The following sections cover operation of the Media Independent Interface in 10/100/1000 Mbps modes. An IEEE 802.3 compliant Ethernet MAC controls the interface.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.1.8.1.1 Data Reception

##### 9.2.1.8.1.1.1 Receive Control

Data received from the PHY is interpreted and output. Interpretation involves detection and removal of the preamble and start of frame delimiter, extraction of the address and frame length, data handling, error checking and reporting, cyclic redundancy checking (CRC), and statistics control signal generation.

##### 9.2.1.8.1.1.2 Receive Inter-Frame Interval

The 802.3 required inter-packet gap (IPG) is 24 GMII clocks (96 bit times) for 10/100 Mbit modes, and 12 GMII clocks (96 bit times) for 1000 Mbit mode. However, the MAC can tolerate a reduced IPG (2 GMII clocks in 10/100 mode and 5 GMII clocks in 1000 mode) with a correct preamble and start frame delimiter.

This interval between frames must comprise (in the following order):

- An Inter-Packet Gap (IPG).
- A seven octet preamble (all octets 0x55).
- A one octet start frame delimiter (0x5D).

#### 9.2.1.8.1.2 Data Transmission

The Gigabit Ethernet Mac Sliver (GMII) passes data to the PHY when enabled. Data is synchronized to the transmit clock rate. The smallest frame that can be sent is two bytes of data with four bytes of CRC (6 byte frame).

##### 9.2.1.8.1.2.1 Transmit Control

A jam sequence is output if a collision is detected on a transmit packet. If the collision was late (after the first 64 bytes have been transmitted) the collision is ignored. If the collision is not late, the controller will back off before retrying the frame transmission. When operating in full duplex mode the carrier sense (CRS) and collision sensing modes are disabled.

##### 9.2.1.8.1.2.2 CRC Insertion

The MAC generates and appends a 32-bit Ethernet CRC onto the transmitted data, if the transmit packet header PASS_CRC bit is 0. For the CPMAC_SL generated CRC case, a CRC at the end of the input packet data is not allowed.

If the header word PASS_CRC bit is set, then the last four bytes of the TX data are transmitted as the frame CRC. The four CRC data bytes should be the last four bytes of the frame and should be included in the packet byte count value. The MAC performs no error checking on the outgoing CRC when the PASS_CRC bit is set.

##### 9.2.1.8.1.2.3 MTXER

The GMII_MTXER signal is not used. If an underflow condition occurs on a transmitted frame, the frame CRC will be inverted to indicate the error to the network. Underflow is a hardware error.

##### 9.2.1.8.1.2.4 Adaptive Performance Optimization (APO)

The Ethernet MAC port incorporates Adaptive Performance Optimization (APO) logic that may be enabled by setting the TX_PACE bit in the MACCONTROL register. Transmission pacing to enhance performance is enabled when set. Adaptive performance pacing introduces delays into the normal transmission of frames, delaying transmission attempts between stations, reducing the probability of collisions occurring during heavy traffic (as indicated by frame deferrals and collisions) thereby increasing the chance of successful transmission.

When a frame is deferred, suffers a single collision, multiple collisions or excessive collisions, the pacing counter is loaded with an initial value of 31. When a frame is transmitted successfully (without experiencing a deferral, single collision, multiple collision or excessive collision) the pacing counter is decremented by one, down to zero.


Copyright © 2011–2016, Texas Instruments Incorporated

With pacing enabled, a new frame is permitted to immediately (after one IPG) attempt transmission only if the pacing counter is zero. If the pacing counter is non zero, the frame is delayed by the pacing delay, a delay of approximately four inter-packet gap delays. APO only affects the IPG preceding the first attempt at transmitting a frame. It does not affect the back-off algorithm for re-transmitted frames.

#### 9.2.1.8.1.2.5 Inter-Packet-Gap Enforcement

The measurement reference for the IPG of 96 bit times is changed depending on frame traffic conditions. If a frame is successfully transmitted without collision, and MCRS is de-asserted within approximately 48 bit times of MTXEN being de-asserted, then 96 bit times is measured from MTXEN. If the frame suffered a collision, or if MCRS is not de-asserted until more than approximately 48 bit times after MTXEN is de-asserted, then 96 bit times (approximately, but not less) is measured from MCRS.

The transmit IPG can be shortened by eight bit times when enabled and triggered. The TX_SHORT_GAP_EN bit in the MACCONTROL register enables the TX_SHORT_GAP input to determine whether the transmit IPG is shorted by eight bit times.

#### 9.2.1.8.1.2.6 Back Off

The Gigabit Ethernet Mac Sliver (GMII) implements the 802.3 binary exponential back-off algorithm.

#### 9.2.1.8.1.2.7 Programmable Transmit Inter-Packet Gap

The transmit inter-packet gap (IPG) is programmable through the TX_GAP register. The default value is decimal 12. The transmit IPG may be increased to the maximum value of 1FFh. Increasing the IPG is not compatible with transmit pacing. The short gap feature will override the increased gap value, so the short gap feature may not be compatible with an increased IPG.

#### 9.2.1.8.1.2.8 Speed, Duplex and Pause Frame Support Negotiation

The CPMAC_SL can operate in half duplex or full duplex in 10/100 Mbit modes, and can operate in full duplex only in 1000 Mbit mode. Pause frame support is included in 10/100/1000 Mbit modes as configured by the host.

### 9.2.1.8.1.3 G/MII Interface Signal Connections and Descriptions

G/MII Interface can operate in GIG/MII Modes.

- In GIG (1000 Mbps) Mode: 3PSW operates only in full duplex mode.
- In MII Mode (100/10 Mbps): 3PSW operates in full duplex and half duplex.

The pin connections of the G/MII Interface is shown in Figure 9-4. The detailed description of the signals in GIG/MII Mode are listed in the following tables.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 9-4. G/MII Interface Connections**

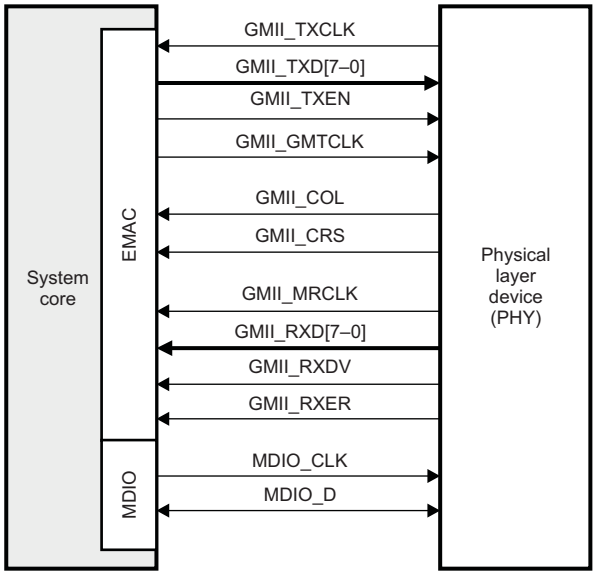



**Table 9-15. G/MII Interface Signal Descriptions in GIG (1000 Mbps) Mode**

| Signal | Type | Description |
|---|---|---|
| GMTCLK | O | The transmit clock is a continuous clock that provides the timing reference for transmit operations. The clock is generated by the CPSW and is 125 MHz at 1000 Mbps operation. It requires a free running clock on MTCLK pin.<br>If the MTCLK is not generated by the PHY, then configure the CPGMAC_SL Mac Control Register Force Gig bit to 1, to generate the required o/p GMTCLK of 125 MHz. |
| MTCLK | I | The transmit clock is a continuous free running clock and is generated by the PHY. |
| MTXD | O | The transmit data pins are a collection of 8 data signals comprising 8 bits of data. MTXD0 is the least-significant bit (LSB). The signals are synchronized by GMTCLK and valid only when MTXEN is asserted. |
| MTXEN | O | The transmit enable signal indicates that the MTXD pins are generating 8bit data for use by the PHY. It is driven synchronously by GMTCLK. |
| MCOL | I | In full-duplex operation, the MCOL pin is used for hardware transmit flow control. Asserting the MCOL pin will stop packet transmissions; packets in the process of being transmitted when MCOL is asserted will complete transmission. The MCOL pin should be held low if hardware transmit flow control is not used. |
| MCRS | I | In full-duplex operation, the MCRS pin should be held low. |
| MRCLK | I | The receive clock is a continuous clock that provides the timing reference for receive operations. The MRXD, MRXDV, and MRXER signals are tied to this clock. The clock is generated by the PHY and is 125 MHz at 1000Mbps of operation. |
| MRXD | I | The receive data pins are a collection of 8 data signals comprising 8 bits of data.MRXD0 is the least-significant bit (LSB). The signals are synchronized by MRCLK and valid only when MRXDV is asserted. |
| MRXDV | I | The receive data valid signal indicates that the MRXD pins are generating byte data for use by the 3PSW. It is driven synchronously to MRCL. |
| MDCLK | O | Management data clock (MDIO_CLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin. |
| MDIO | I/O | MDIO DATA (MDI0_D). MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-16. G/MII Interface Signal Descriptions in MII (100/10 Mbps) Mode**

| Signal | Type | Description |
|---|---|---|
| GMTCLK | O | The clock is generated by the CPSW and is running at 125 MHz. |
| MTCLK | I | The transmit clock is a continuous clock that provides the timing reference for transmit operations. The MTXD and MTXEN signals are tied to this clock. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation. |
| MTXD | O | MTXD[7-4] pins of MTXD data are not used. The transmit data pins are a collection of 4 data signals MTXD[3-0] comprising 4 bits of data. MTXD0 is the least-significant bit (LSB). The signals are synchronized by MTCLK and valid only when MTXEN is asserted. |
| MTXEN | O | The transmit enable signal indicates that the MTXD pins are generating 4bit data for use by the PHY. It is driven synchronously by MTCLK. |
| MCOL | I | In half-duplex operation, the MCOL pin is asserted by the PHY when it detects a collision on the network. It remains asserted while the collision condition persists. This signal is not necessarily synchronous to MTCLK nor MRCLK.<br>In full-duplex operation, the MCOL pin is used for hardware transmit flow control. Asserting the MCOL pin will stop packet transmissions; packets in the process of being transmitted when MCOL is asserted will complete transmission. The MCOL pin should be held low if hardware transmit flow control is not used. |
| MCRS | I | In half-duplex operation, the MCRS pin is asserted by the PHY when the network is not idle in either transmit or receive. The pin is de-asserted when both transmit and receive are idle. This signal is not necessarily synchronous to MTCLK nor MRCLK.<br>In full-duplex operation, the MCRS pin should be held low. |
| MRCLK | I | The receive clock is a continuous clock that provides the timing reference for receive operations. The MRXD, MRXDV, and MRXER signals are tied to this clock. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation. |
| MRXD | I | MRXD[7-4] bits are unused in the MII mode. The receive data pins are a collection of 4 data signals comprising 4 bits of data.MRXD0 is the least-significant bit (LSB).The signals are synchronized by MRCLK and valid only when MRXDV is asserted. |
| MRXDV | I | The receive data valid signal indicates that the MRXD pins are generating nibble data for use by the 3PSW. It is driven synchronously to MRCLK. |
| MDCLK | O | Management data clock (MDIO_CLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin. |
| MDIO | I/O | MDIO DATA (MDI0_D). MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations. |

#### *9.2.1.8.2 RMII Interface*

The CPRMII peripheral shall be compliant to the RMII specification document.

##### *9.2.1.8.2.1 Features*

- Source Synchronous 10/100 Mbit operation
- Full and Half Duplex support

##### *9.2.1.8.2.2 RMII Receive (RX)*

The CPRMII receive (RX) interface converts the input data from the external RMII PHY (or switch) into the required MII (CPGMAC) signals. The carrier sense and collision signals are determined from the RMII input data stream and transmit inputs as defined in the RMII specification.

An asserted RMRXER on any di-bit in the received packet will cause an MRXER assertion to the CPGMAC during the packet. In 10Mbps mode, the error is not required to be duplicated on 10 successive clocks. Any di-bit which has an asserted RMII_RXER during any of the 10 replications of the data will cause the error to be propagated.


Copyright © 2011–2016, Texas Instruments Incorporated

Any received packet that ends with an improper nibble boundary aligned RMCRSDV toggle will issue an MRXER during the packet to the CPGMAC. Also, a change in speed or duplex mode during packet operations will cause packet corruption.

The CPRMII can accept receive packets with shortened preambles, but 0x55 followed by a 0x5D is the shortest preamble that will be recognized (1 preamble byte with the start of frame byte). At least one byte of preamble with the start of frame indicator is required to begin a packet. An asserted RMCRSDV without at least a single correct preamble byte followed by the start of frame indicator will be ignored.

#### 9.2.1.8.2.3 RMII Transmit (TX)

The CPRMII transmit (TX) interface converts the 3PSW MII input data into the RMII transmit format. The data is then output to the external RMII PHY.

The 3PSW does not source the transmit error (MII TXERR) signal. Any transmit frame from the CPGMAC with an error (underrun) will be indicated as an error by an error CRC. Transmit error is assumed to be de-asserted at all times and is not an input into the CPRMII module. Zeroes are output on RMTXD[1:0] for each clock that RMTXEN is de-asserted.

#### 9.2.1.8.2.4 RMII Signal Connections and Descriptions

Figure 9-5 shows a device with integrated 3PSW and MDIO interfaced via a RMII connection in a typical system. The individual CPSW and MDIO signals for the RMII interface are summarized in Table 9-17.

For more information, refer to either the IEEE 802.3 standard or ISO/IEC 8802-3:2000(E).

**Figure 9-5. RMII Interface Connections**

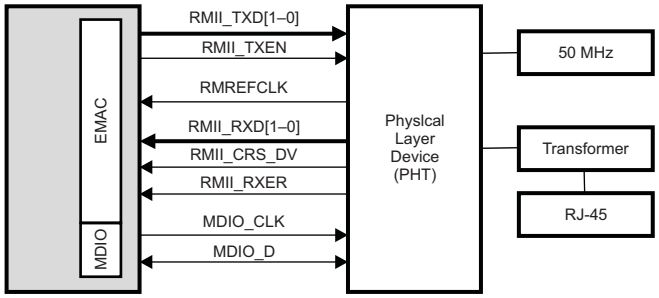



**Table 9-17. RMII Signal Descriptions**

| Signal | Type | Description |
|---|---|---|
| RMTXD (1-0) | O | Transmit data (RMII_TXD). The transmit data pins are a collection of 2 bits of data. RMTDX0 is the least-significant bit (LSB). The signals are synchronized by RMREFCLK and valid only when RMTXEN is asserted. |
| RMXEN | O | Transmit enable (RMII_TXEN). The transmit enable signal indicates that the RMII_TXD pins are generating data for use by the PHY. RMII_TXEN is synchronous to RMREFCLK. |
| RMREFCLK | I/O | RMII reference clock. (RMREFCLK).<br>The reference clock is used to synchronize all RMII signals. RMREFCLK must be continuous and fixed at 50 MHz. |
| RMRXD (1-0) | I | Receive data (RMII_RXD). The receive data pins are a collection of 2 bits of data. RMRDX0 is the least-significant bit (LSB). The signals are synchronized by RMREFCLK and valid only when RMCRSDV is asserted and RMRXER is de-asserted. |
| RMCRSDV | I | Carrier sense/receive data valid (RMCRSDV). Multiplexed signal between carrier sense and receive data valid. |
| R,RXER | I | Receive error (RMII_RXER). The receive error signal is asserted to indicate that an error was detected in the received frame. |
| MDCLK | O | Management data clock (MDIO_CLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-17. RMII Signal Descriptions (continued)**

| Signal | Type | Description |
|---|---|---|
| MDIO | I/O | MDIO DATA(MDI0_D). MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations. |

#### 9.2.1.8.3 RGMII Interface

The CPRGMII peripheral shall be compliant to the RGMII specification document.

##### 9.2.1.8.3.1 Features

- Supports 1000/100/10 Mbps Speed
- MII mode is not supported
- Selectable internal delay on transmit

##### 9.2.1.8.3.2 RGMII Receive (RX)

The CPRGMII receive (RX) interface converts the source synchronous DDR input data from the external RGMII PHY into the required G/MII (CPGMAC) signals.

##### 9.2.1.8.3.3 In-Band Mode of Operation

The CPRGMII is operating in the in-band mode of operation when the RGMII_RX_INBAND input is asserted. RGMII_RX_INPUT is asserted by configuring the EXT_EN bit to 1 of the SLn_MACCONTROL register. The link status, duplexity, and speed are determined from the RGMII input data stream as defined in the RGMII specification. The link speed is indicated as shown in Table 9-18.

**Table 9-18. Link Speed**

| RGMII_SPEED (1:0) | Link Speed |
|---|---|
| 00 | 10 Mbs mode |
| 01 | 100 Mbs mode |
| 10 | 1000 Mbs mode |
| 11 | reserved |

##### 9.2.1.8.3.4 Forced Mode of Operation

The CPRGMII is operating in the forced mode of operation when the RGMII_RX_INBAND input is de-asserted. In the forced mode of operation, the in-band data is ignored if present. The link status is forced high, and the duplexity and speed are determined from the RGMII_FULLDUPLEX_IN and RGMII_GIG_IN inputs. If the RGMII_GIG_IN input is asserted, then operation is gigabit mode (RGMII_SPEED (1:0) = 10). If the RGMII_GIG_IN input is de-asserted, the operation is 100 Mbps mode (RGMII_SPEED (1:0) = 01).

##### 9.2.1.8.3.5 RGMII Transmit (TX)

The CPRGMII transmit (TX) interface converts the CPGMAC G/MII input data into the DDR RGMII format. The DDR data is then output to the external PHY.

The CPGMAC does not source the transmit error (MTXERR) signal. Any transmit frame from the CPGMAC with an error (underrun) will be indicated as an error by an error CRC. Transmit error is assumed to be de-asserted at all times and is not an input into the CPRGMII module.


Copyright © 2011–2016, Texas Instruments Incorporated

In 10/100 mode, the MTXD (7:0) data bus uses only the lower nibble. The CPRGMII will output the lower nibble twice in 10/100 mode to avoid unnecessary signal switching.

Packets will be precluded from transmission through the CPRGMII module for 4096 transmit clocks after the rising edge of RGMII_LINK. Packet transmission will begin on the first GMII_MTXEN rising edge after the 4096 transmit clock count has expired.

The RGMII0/1_ID_MODE bit value in the GMII_SEL register determines whether or not the transmit delay is included in the CPRGMII or not. When RGMII0/1_ID_MODE bit is cleared to 0, the transmit delay is included. The RGMII0/1_ID_MODE input is a configuration input only and is not intended to be changed during packet operations.

See the *Control Module* chapter, MII mode selection register (GMII_SEL) to configure the CPRGMII Internal Delay/No Delay in Transmit Mode of Operation.

#### 9.2.1.8.3.6 RGMII Signal Connections and Descriptions

Figure 9-6 shows a device with integrated CPSW and MDIO interfaced via a RGMII connection in a typical system. The individual CPSW and MDIO signals for the RGMII interface are summarized in Table 9-19.

**Figure 9-6. RGMII Interface Connections**

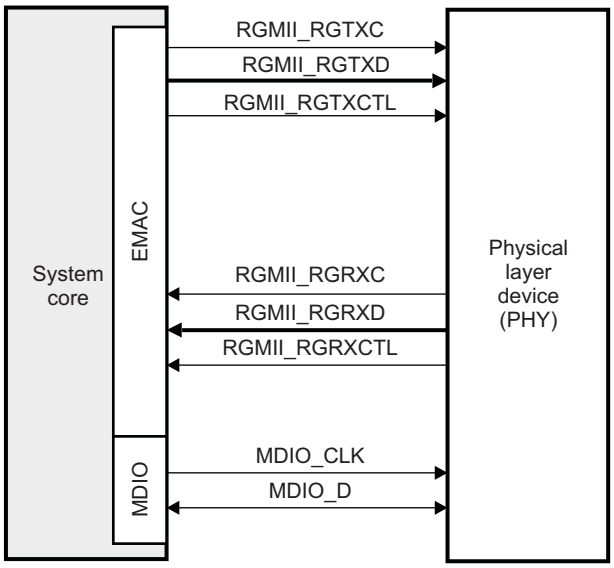



**Table 9-19. RGMII Signal Descriptions**

| Signal | Type | Description |
|---|---|---|
| RGTXD[3-0] | O | The transmit data pins are a collection of 4 bits of data.RGTDX0 is the least-significant bit (LSB). The signals are valid only when RGTXCTL is asserted. |
| RGTXCTL | O | Transmit Control/enable. The transmit enable signal indicates that the RGTXD pins are generating data for use by the PHY. |
| RGTXC | O | The transmit reference clock will be 125 MHz, 25 MHz, or 2.5 MHz depending on speed of operation. |
| RGRXD[3-0] | I | The receive data pins are a collection of 4 bits of data. RGRDX0 is the least-significant bit (LSB).<br>The signals are valid only when RGRXCTL is asserted |
| RGRXCTL | I | The receive data valid/control signal indicates that the RGRXD pins are nibble data for use by the 3PSW. |
| RGRXC | I | The receive clock is a continuous clock that provides the timing reference for receive operations. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation, 125 MHz at 1000 Mbps of operation. |
| MDCLK | O | Management data clock (MDIO_CLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-19. RGMII Signal Descriptions (continued)**

| Signal | Type | Description |
|---|---|---|
| MDIO | I/O | MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations. |

#### *9.2.1.8.4 Frame Classification*

Received frames are proper (good) frames if they are between 64 and RX_MAXLEN in length (inclusive) and contain no errors (code/align/CRC).

Received frames are long frames if their frame count exceeds the value in the RX_MAXLEN register. The RX_MAXLEN register reset (default) value is 1518 (decimal). Long received frames are either oversized or jabber frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment errors are jabber frames.

Received frames are short frames if their frame count is less than 64 bytes. Short frames that contain no errors are undersized frames. Short frames with CRC, code, or alignment errors are fragment frames.

A received long packet will always contain RX_MAXLEN number of bytes transferred to memory (if RX_CEF_EN = 1). An example with RX_MAXLEN = 1518 is:

- If the frame length is 1518, then the packet is not a long packet and there will be 1518 bytes transferred to memory.
- If the frame length is 1519, there will be 1518 bytes transferred to memory. The last three bytes will be the first three CRC bytes.
- If the frame length is 1520, there will be 1518 bytes transferred to memory. The last two bytes will be the first two CRC bytes.
- If the frame length is 1521, there will be 1518 bytes transferred to memory. The last byte will be the first CRC byte.

If the frame length is 1522, there will be 1518 bytes transferred to memory. The last byte will be the last data byte.

### 9.2.1.9 Embedded Memories

**Table 9-20. Embedded Memories**

| Memory Type Description | Number of Instances | |
|---|---|---|
| Single port 2560 by 64 RAM | 3 | (Packet FIFO's) |
| Single port 64-word by 1152-bit RAM | 1 | (ALE) |
| Single port 2048-word by 32-bit RAM | 1 | (CPPI) |

### 9.2.1.10 Flow Control

There are two types of switch flow control: CPPI port flow control and Ethernet port flow control. The CPPI and Ethernet port naming conventions for data flow into and out of the switch are reversed. For the CPPI port (port 0), transmit operations move packets from external memory into the switch and then out to either or both Ethernet transmit ports (ports 1 and 2). CPPI receive operations move packets that were received on either or both Ethernet receive ports to external memory.

#### *9.2.1.10.1 CPPI Port Flow Control*

The CPPI port has flow control available for transmit (switch ingress). CPPI receive operations (switch egress) do not require flow control. CPPI Transmit flow control is initiated when enabled and triggered. CPPI transmit flow control is enabled by setting the P0_FLOW_EN bit in the CPSW_FLOW_CONTROL register. CPPI transmit flow control is enabled by default on reset because host packets should not be dropped in any mode of operation.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.1.10.2 Ethernet Port Flow Control

The Ethernet ports have flow control available for transmit and receive. Transmit flow control stops the Ethernet port from transmitting packets to the wire (switch egress) in response to a received pause frame. Transmit flow control does not depend on FIFO usage.

The ethernet ports have flow control available for receive operations (packet ingress). Ethernet port receive flow control is initiated when enabled and triggered. Packets received on an ethernet port can be sent to the other ethernet port or the CPPI port (or both). Each destination port can trigger the receive ethernet port flow control. An ethernet destination port triggers another ethernet receive flow control when the destination port is full.

When a packet is received on an ethernet port interface with enabled flow control the below occurs:

- The packet will be sent to all ports that currently have room to take the entire packet.
- The packet will be retried until successful to all ports that indicate they don't have room for the packet.

The flow control trigger to the CPGMAC_SL will be asserted until the packet has been sent, and there is room in the logical receive FIFO for packet runout from another flow control trigger (RX_PKT_CNT = 0). Ethernet port receive flow control is disabled by default on reset. Ethernet port receive flow control requires that the RX_FLOW_EN bit in the associated CPGMAC_SL be set to 1. When receive flow control is enabled on a port, the port's associated FIFO block allocation must be adjusted. The port RX allocation must increase from the default three blocks to accommodate the flow control runout. A corresponding decrease in the TX block allocation is required. If a sending port ignores a pause frame then packets may overrun on receive (and be dropped) but will not be dropped on transmit. If flow control is disabled for G/MII ports, then any packets that are dropped are dropped on transmit and not on receive.

##### 9.2.1.10.2.1 Receive Flow Control

When enabled and triggered, receive flow control is initiated to limit the CPGMAC_SL from further frame reception. Half-duplex mode receive flow control is collision based while full duplex mode issues 802.3X pause frames. In either case, receive flow control prevents frame reception by issuing the flow control appropriate for the current mode of operation. Receive flow control is enabled by the RX_FLOW_EN bit in the MACCONTROL register. Receive flow control is triggered (when enabled) when the RX_FLOW_TRIGGER input is asserted. The CPGMAC_SL is configured for collision or IEEE 802.3X flow control via the fullduplex bit in the MACCONTROL register.

###### 9.2.1.10.2.1.1 Collision Based Receive Buffer Flow Control

Collision-based receive buffer flow control provides a means of preventing frame reception when the port is operating in half-duplex mode (FULLDUPLEX is cleared in MACCONTROL). When receive flow control is enabled and triggered, the port will generate collisions for received frames. The jam sequence transmitted will be the twelve byte sequence C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3 (hex). The jam sequence will begin no later than approximately as the source address starts to be received. Note that these forced collisions will not be limited to a maximum of 16 consecutive collisions, and are independent of the normal back-off algorithm. Receive flow control does not depend on the value of the incoming frame destination address. A collision will be generated for any incoming packet, regardless of the destination address.

###### 9.2.1.10.2.1.2 IEEE 802.3X Based Receive Flow Control

IEEE 802.3x based receive flow control provides a means of preventing frame reception when the port is operating in full-duplex mode (FULLDUPLEX is set in MACCONTROL). When receive flow control is enabled and triggered, the port will transmit a pause frame to request that the sending station stop transmitting for the period indicated within the transmitted pause frame.

The CPGMAC_SL will transmit a pause frame to the reserved multicast address at the first available opportunity (immediately if currently idle, or following the completion of the frame currently being transmitted). The pause frame will contain the maximum possible value for the pause time (FFFFh). The MAC will count the receive pause frame time (decrements FF00h down to 0) and retransmit an outgoing pause frame if the count reaches zero. When the flow control request is removed, the MAC will transmit a pause frame with a zero pause time to cancel the pause request.


Copyright © 2011–2016, Texas Instruments Incorporated

Note that transmitted pause frames are only a request to the other end station to stop transmitting. Frames that are received during the pause interval will be received normally (provided the RX FIFO is not full).

Pause frames will be transmitted if enabled and triggered regardless of whether or not the port is observing the pause time period from an incoming pause frame.

The CPGMAC_SL will transmit pause frames as:

- The 48-bit reserved multicast destination address 01.80.C2.00.00.01.
- The 48-bit source address - SL_SA(47:0).
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause time value FF.FF. A pause-quantum is 512 bit-times. Pause frames sent to cancel a pause request will have a pause time value of 00.00.
- Zero padding to 64-byte data length (The CPGMAC_SL will transmit only 64 byte pause frames).
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

If RX_FLOW_EN is cleared to 0 while the pause time is nonzero, then the pause time will be cleared to 0 and a zero count pause frame will be sent.

#### *9.2.1.10.2.2 Transmit Flow Control*

Incoming pause frames are acted upon, when enabled, to prevent the CPGMAC_SL from transmitting any further frames. Incoming pause frames are only acted upon when the FULLDUPLEX and TX_FLOW_EN bits in the MACCONTROL register are set. Pause frames are not acted upon in half-duplex mode. Pause frame action will be taken if enabled, but normally the frame will be filtered and not transferred to memory. MAC control frames will be transferred to memory if the RX_CMF_EN (copy MAC frames) bit in the MACCONTROL register is set. The TX_FLOW_EN and FULLDUPLEX bits effect whether or not MAC control frames are acted upon, but they have no effect upon whether or not MAC control frames are transferred to memory or filtered.

Pause frames are a subset of MAC Control Frames with an opcode field = 0001h. Incoming pause frames will only be acted upon by the port if:

- TX_FLOW_EN is set in MACCONTROL register, and
- the frame's length is 64 to RX_MAXLEN bytes inclusive, and
- the frame contains no CRC error or align/code errors.

The pause time value from valid frames will be extracted from the two bytes following the opcode. The pause time will be loaded into the port's transmit pause timer and the transmit pause time period will begin.

If a valid pause frame is received during the transmit pause time period of a previous transmit pause frame then:

- if the destination address is not equal to the reserved multicast address or any enabled or disabled unicast address, then the transmit pause timer will immediately expire, or
- if the new pause time value is zero then the transmit pause timer will immediately expire, else
- the port transmit pause timer will immediately be set to the new pause frame pause time value. (Any remaining pause time from the previous pause frame will be discarded).

If TX_FLOW_EN in MACCONTROL register is cleared, then the pause-timer will immediately expire.

The port will not start the transmission of a new data frame any sooner than 512-bit times after a pause frame with a non-zero pause time has finished being received (MRXDV going inactive). No transmission will begin until the pause timer has expired (the port may transmit pause frames in order to initiate outgoing flow control). Any frame already in transmission when a pause frame is received will be completed and unaffected.

Incoming pause frames consist of the below:


Copyright © 2011–2016, Texas Instruments Incorporated

- A 48-bit destination address equal to:
- The reserved multicast destination address 01.80.C2.00.00.01, or the SL_SA (47:0) input mac source address.
- The 48-bit source address of the transmitting device.
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause_time. A pause-quantum is 512 bit-times.
- Padding to 64-byte data length.
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

The padding is required to make up the frame to a minimum of 64 bytes. The standard allows pause frames longer than 64 bytes to be discarded or interpreted as valid pause frames. The CPGMAC_SL will recognize any pause frame between 64 bytes and RX_MAXLEN bytes in length.

#### 9.2.1.11 Short Gap

The port 1 (and port 2) transmit inter-packet gap (IPG) may be shortened by eight bit times when enabled and triggered. The TX_SHORT_GAP_EN bit in the SL1_MACCONTROL (SL2_MACCONTROL) register enables the gap to be shortened when triggered. The condition is triggered when the port 1 (port 2) transmit FIFO has a user defined number of FIFO blocks used. The port 1 transmit FIFO blocks used determines if the port 1 gap is shortened, and the port 2 transmit FIFO blocks used determines if the port 2 gap is shortened. The CPSW_GAP_THRESH register value determines the port 1 short gap threshold, and the CPSW_GAP_THRESH register value determines the port 2 short gap threshold.

#### 9.2.1.12 Switch Latency

The CPSW_3G is a store and forward switch. The switch latency is defined as the amount of time between the end of packet reception of the received packet to the start of the output packet transmit.

**Table 9-21. Switch Latency**

| Mode | Latency |
|---|---|
| Gig (1000) | 880 ns |
| 100 | 1.3 µs |
| 10 | 6.5 µs |

#### 9.2.1.13 Emulation Control

The emulation control input (EMUSUSP) and submodule emulation control registers allow CPSW_3G operation to be completely or partially suspended. There are three CPSW_3G submodules that contain emulation control registers (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). The submodule emulation control registers must be accessed to facilitate CPSW_3G emulation control. The CPSW_3G module enters the emulation suspend state if all three submodules are configured for emulation suspend and the emulation suspend input is asserted. A partial emulation suspend state is entered if one or two submodules is configured for emulation suspend and the emulation suspend input is asserted. Emulation suspend occurs at packet boundaries. The emulation control feature is implemented for compatibility with other peripherals.

**CPGMAC_SL Emulation Control**


Copyright © 2011–2016, Texas Instruments Incorporated

The emulation control input (TBEMUSUP) and register bits (SOFT and FREE bits in the EMCONTROL register) allow CPGMAC_SL operation to be suspended. When the emulation suspend state is entered, the CPGMAC_SL will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For receive, frames that are detected by the CPGMAC_SL after the suspend state is entered are ignored. Emulation control is implemented for compatibility with other peripherals.

**CPDMA Emulation Control**

The emulation control input (TBEMUSUP) and register bits (SOFT and FREE bits in the EMCONTROL register) allow CPDMA operation to be suspended. When the emulation suspend state is entered, the CPDMA will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For transmission, any complete or partial frame in the tx cell FIFO will be transmitted. For receive, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Emulation control is implemented for compatibility with other peripherals

Table 9-22 shows the operations of the emulation control input and register bits.

**Table 9-22. Emulation Control Input**

| EMUSUSP | SOFT | FREE | Description |
|---|---|---|---|
| 0 | X | X | Normal Operation |
| 1 | 0 | 0 | Normal Operation |
| 1 | 1 | 0 | Emulation Suspend |
| 1 | X | 1 | Normal Operation |

#### 9.2.1.14 Software IDLE

The submodule software idle register bits enable CPSW_3G operation to be completely or partially suspended by software control. There are three CPSW_3G submodules that contain software idle register bits (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). Each of the three submodules may be individually commanded to enter the idle state. The idle state is entered at packet boundaries, and no further packet operations will occur on an idled submodule until the idle command is removed. The CPSW_3G module enters the idle state when all three submodules are commanded to enter and have entered the idle state. Idle status is determined by reading or polling the three submodule idle bits. The CPSW_3G is in the idle state when all three submodules are in the idle state. The CPSW_SOFT_IDLE bit may be set if desired after the submodules are in the idle state. The CPSW_SOFT_IDLE bit causes packets to not be transferred from one FIFO to another FIFO internal to the switch.

#### 9.2.1.15 Software Reset

The CPSW_3G software reset register, CPSW_3GSS software reset register and the three submodule software reset registers enable the CPSW_3GSS to be reset by software. There are three CPSW_3G submodules that contain software reset registers (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). Each of the three submodules may be individually commanded to be reset by software.

For the CPDMA, the reset state is entered at packet boundaries, at which time the CPDMA reset occurs. The CPGMAC_SL soft reset is immediate. Submodule reset status is determined by reading or polling the submodule reset bit. If the submodule reset bit is read as a one, then the reset process has not yet completed. The submodule soft reset process could take up to 2ms each. The reset has completed if the submodule reset bit is read as a zero.

After all three submodules (in any order) have been reset and a read of each submodule reset bit indicates that the reset process is complete, the CPSW_3G software reset register bit may be written to complete the CPSW_3G module software reset operation. The CPSW_3G software reset bit controls the reset of the FIFO's, the statistics submodule, and the address lookup engine (ALE). The CPSW_3G software reset is immediate and will be indicated by reading a zero from the soft reset bit.

The CPSW_3GSS software reset bit controls the reset of the INT, REGS, and CPPI. The CPSW_3GSS software reset is immediate and will be indicated by reading a zero from the soft reset bit.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.1.16 FIFO Loopback

FIFO loopback mode is entered when the FIFO_LOOPBACK bit in the CPSW_CONTROL register is set. FIFO loopback mode causes packets received on a port to be turned around and transmitted back on the same port. Port 0 receive is fixed on channel 0 in FIFO loopback mode. The RXSOFOVERRUN statistic is incremented for each packet sent in FIFO loopback mode. Packets sent in with errors are returned with errors (they are not dropped). FIFO loopback is intended as a simple mechanism for test purposes. FIFO loopback should be performed in fullduplex mode only.


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.2.2 Common Platform Time Sync (CPTS)

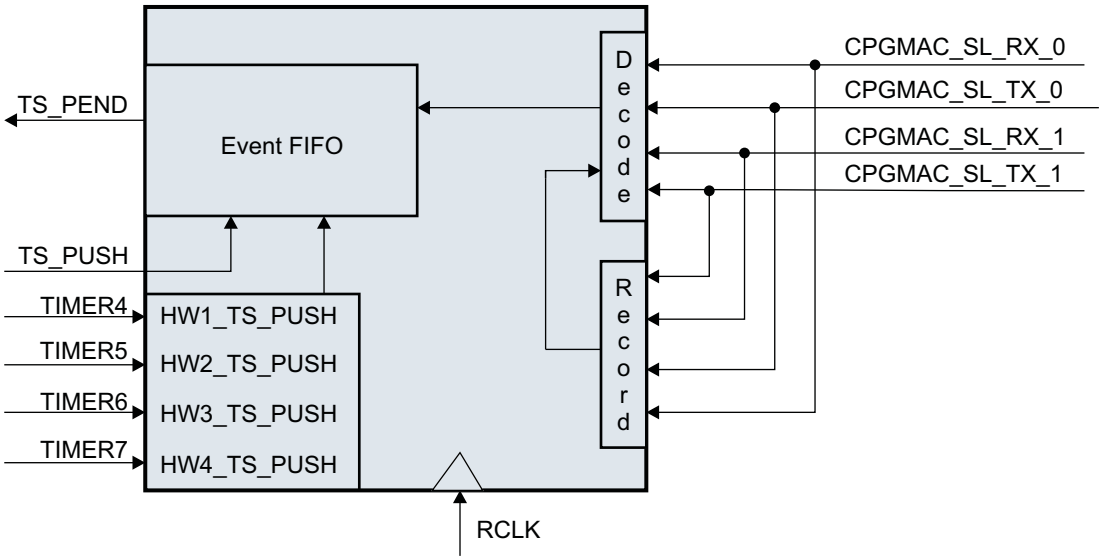
The Common Platform Time Sync (CPTS) module is used to facilitate host control of time sync operations. It enables compliance with the IEEE 1588-2008 standard for a precision clock synchronization protocol.

#### 9.2.2.1 Architecture

Figure 9-7 shows the architecture of the CPTS module inside the 3PSW Ethernet Subsystem. Time stamp values for every packet transmitted or received on either port of the 3PSW are recorded. At the same time, each packet is decoded to determine if it is a valid time sync event. If so, an event is loaded into the Event FIFO for processing containing the recorded time stamp value when the packet was transmitted or received.

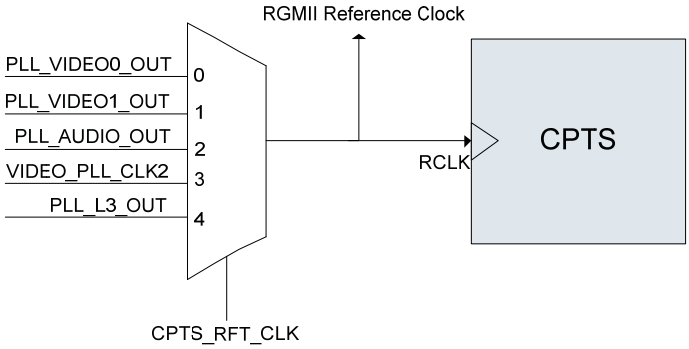
In addition, both hardware (HWx_TS_PUSH) and software (TS_PUSH) can be used to read the current time stamp value though the Event FIFO

**Figure 9-7. CPTS Block Diagram**




The reference clock used for the time stamp (RCLK) is sourced from one of the five sources, as shown in Figure 9-8. The PLL used for sourcing RCLK can be selected by configuring the RMII _REFCLK_SRC register in the *Control Module* chapter. Note that if RGMII mode is used, RCLK must be exactly 250 MHz.

**Figure 9-8. CPTS RCLK Source**




#### 9.2.2.2 Common Platform Time Sync Initialization

The CPTS module should be configured as:


Copyright © 2011–2016, Texas Instruments Incorporated

1. Reset the CPTS module.
2. Clear the CPTS_EN bit in the CPTS_CONTROL register.
3. Write the CPTS_RFT_CLK value in the RMII_RFTCLK_SEL register in the Control Module with the desired reference clock selection.
4. Set the CPTS_EN bit in the CPTS_CONTROL register.
5. If using interrupts and not polling, enable the interrupt by setting the TS_PEND_EN bit in the CPTS_INT_ENABLE register.

#### 9.2.2.3 Time Stamp Value

The time stamp value is a 32-bit value that increments on each RCLK rising edge when CPTS_EN is set to 1. When CPTS_EN is cleared to 0, the time stamp value is reset to 0.

If more than 32-bits of time stamp are required by the application, the host software must maintain the necessary number of upper bits. The upper time stamp value should be incremented by the host when the rollover event is detected.

For test purposes, the time stamp can be written via the time stamp load function (CPTS_TS_LOAD_VAL and CPTS_TS_LOAD_EN registers).

#### 9.2.2.4 Event FIFO

All time sync events are push onto the Event FIFO. There are 16 locations in the event FIFO with no overrun indication supported. Software must service the event FIFO in a timely manner to prevent FIFO overrun.

#### 9.2.2.5 Time Sync Events

Time Sync events are 64-bit values that are pushed onto the event FIFO and read in two 32-bit reads. The two 32-bit registers, CPTS_EVENT_HIGH and CPTS_EVENT_LOW, are described in Section 9.4. There are five types of sync events:

- Time stamp push event
- Time stamp counter rollover event
- Time stamp counter half-rollover event
- Ethernet receive event
- Ethernet transmit event

##### *9.2.2.5.1 Time Stamp Push Event*

Software can obtain the current time stamp value (at the time of the write) by initiating a time stamp push event. The push event is initiated by setting the TS_PUSH bit of the CPTS_TS_PUSH register. The time stamp value is returned in the event, along with a time stamp push event code. Software should not push a second time stamp event on to the FIFO until the first time stamp value has been read from the event FIFO.

##### *9.2.2.5.2 Time Stamp Counter Rollover Event*

The CPTS module contains a 32-bit time stamp value. The counter upper bits are maintained by host software. The rollover event indicates to software that the time stamp counter has rolled over from FFFF FFFFh to 0000 0000h, and the software maintained upper count value should be incremented.

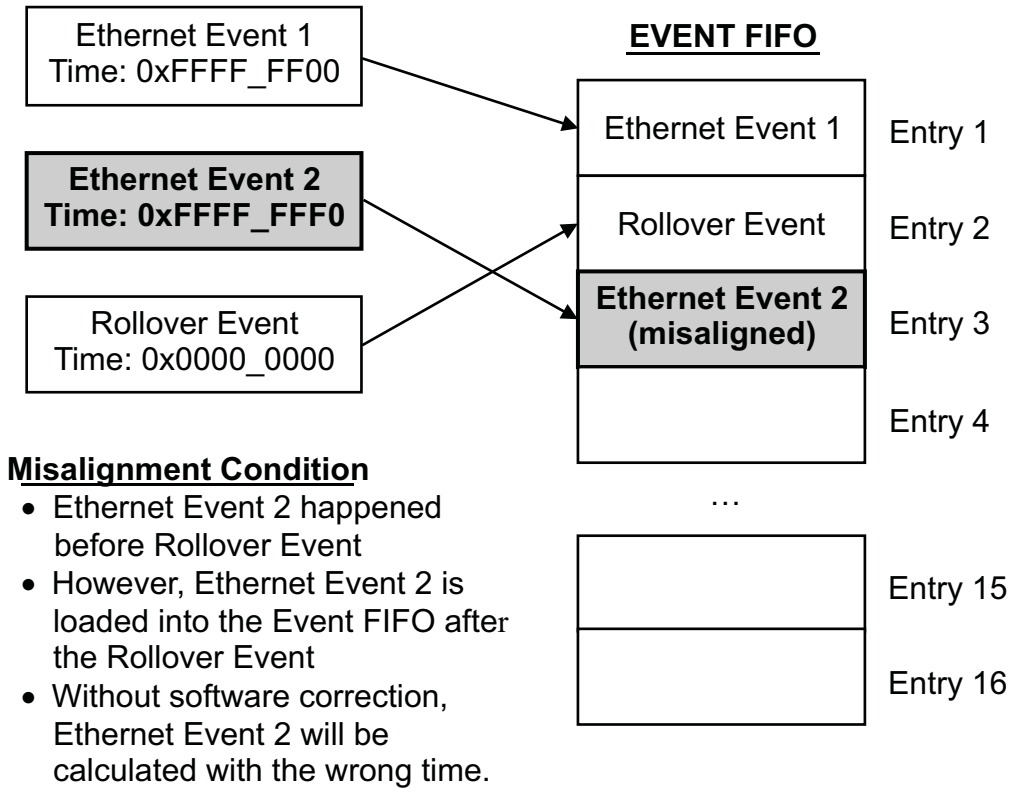

Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.2.5.3 Time Stamp Counter Half-rollover Event

The CPTS includes a time stamp counter half-rollover event. The half-rollover event indicates to software that the time stamp value has incremented from 7FFF FFFFh to 8000 0000h. The half-rollover event is included to enable software to correct a misaligned event condition. The half-rollover event is included to enable software to determine the correct time for each event that contains a valid time stamp value, such as an Ethernet event. If an Ethernet event occurs around a counter rollover (full rollover), the rollover event could possibly be loaded into the event FIFO before the Ethernet event, even though the Ethernet event time was actually taken before the rollover. Figure 9-9 shows a misalignment condition.

Host software must detect and correct for misaligned event conditions. For every event after a rollover and before a half-rollover, software must examine the time stamp most significant bit. If bit 31 of the time stamp value is low (0000 0000h through 7FFF FFFFh), then the event time stamp was taken after the rollover and no correction is required. If the value is high (8000 0000h through FFFF FFFFh), the time stamp value was taken before the rollover and a misalignment is detected. The misaligned case indicates to software that it must subtract one from the upper count value stored in software to calculate the correct time for the misaligned event. The misaligned event occurs only on the rollover boundary and not on the half-rollover boundary. Software only needs to check for misalignment from a rollover event to a half-rollover event.

**Figure 9-9. Event FIFO Misalignment Condition**

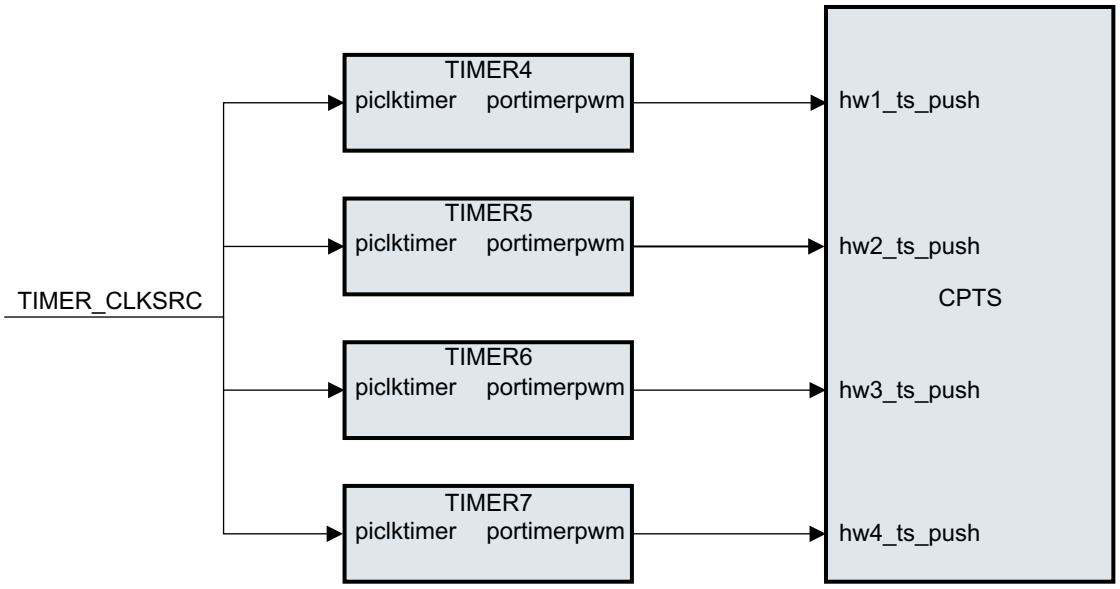



#### 9.2.2.5.4 Hardware Time Stamp Push Event

There are four hardware time stamp inputs (HW1/4_TS_PUSH) that can cause hardware time stamp push events to be loaded into the Event FIFO. Each hardware time stamp input is internally connected to the PORTIMERPWM output of each timer as shown in Figure 9-10.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 9-10. HW1/4_TSP_PUSH Connection**




The event is loaded into the event FIFO on the rising edge of the timer, and the PORT_NUMBER field in the EVENT_HIGH register indicates the hardware time stamp input that caused the event.

Each hardware time stamp input must be asserted for at least 10 periods of the selected RCLK clock. Each input can be enabled or disabled by setting the respective bits in the CPTS_CONTROL register.

Hardware time stamps are intended to be an extremely low frequency signals, such that the event FIFO does not overrun. Software must keep up with the event FIFO and ensure that there is no overrun, or events will be lost.

#### *9.2.2.5.5 Ethernet Port Events*

Packets transmitted or received on each Ethernet port can generate Ethernet Transmit Events or Ethernet Receive Events, respectively. The CPTS hardware will decode each packet to determine if it is a valid CPTS time sync event.

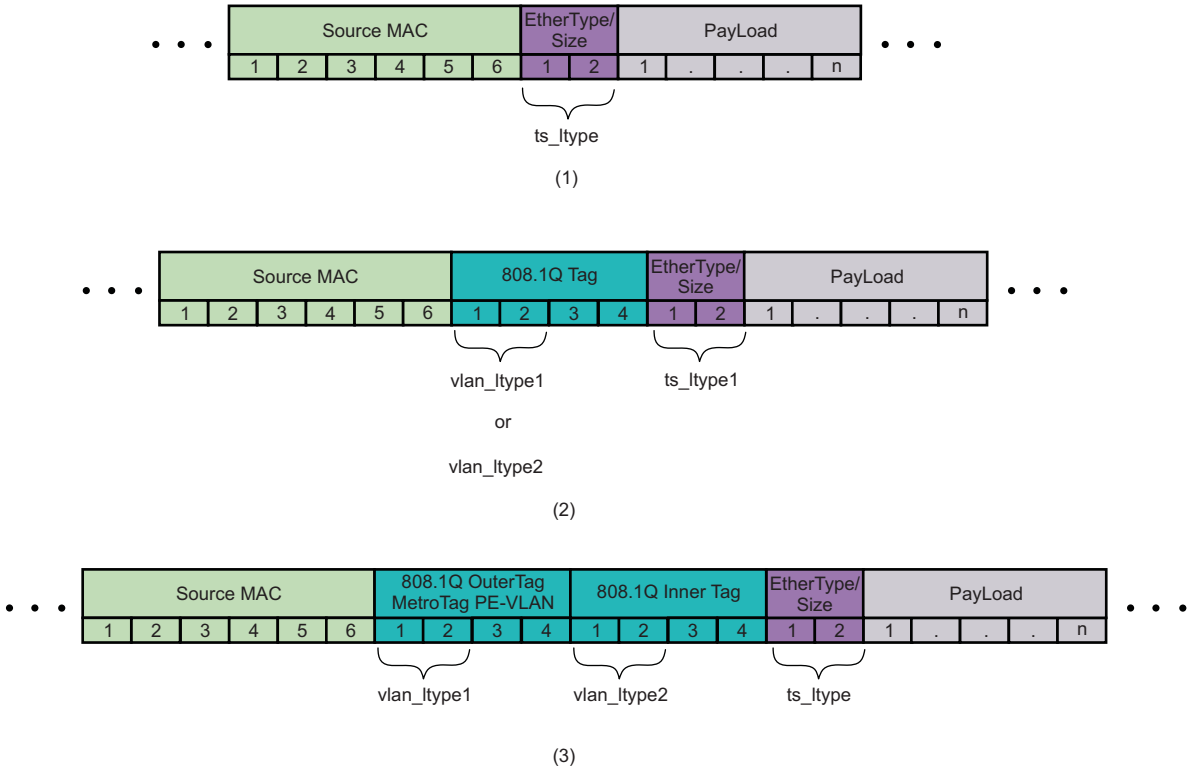
According to the IEEE 802.3 Ethernet standard, each Ethernet frame contains a 2-octet EtherType field to indicate which protocol is encapsulated in the PayLoad field, as shown in Figure 9-11. For standard time sync packets, this will contain the EtherType for the Precision Time Protocol (IEEE 1588), which is defined as 0x88F7. The CPTS hardware will compare this field to the Pn_TS_LTYPE field in the Pn_TS_SEQ_LTYPE register, which should also be programmed to 88F7h.

When a virtual LAN is used, an additional 4-octet 802.1Q tag is inserted in the Ethernet frame before the EtherType field, as shown in Figure 9-11. To indicate to the CPTS hardware that a virtual LAN is in use, the Pn_TS_TX_VLAN_LTYPE1_EN/Pn_TS_RX_VLAN_LTYPE1_EN (or Pn_TS_TX_VLAN_LTYPE2_EN/Pn_TS_RX_VLAN_LTYPE2_EN) enable bit must be set in the Pn_TS_CTL register. The EtherType for the 802.1Q tag is defined as 0x8100, and the CTPS hardware will compare this value to the Pn_TS_VLAN_LTYPE1 (or Pn_TS_VLAN_LTYPE2 depending on which enable bit was set) field in the Pn_TS_VLAN register, which should also be programmed to 0x8100.

When two stacked VLANs are used, two additional 4-octet 801.Q tags are inserted in the Ethernet frame before the EtherType field, as shown in Figure 9-11. In this case, both Pn_TS_VLAN_LTYPE1 and Pn_TS_VLAN_LTYPE2 must be enabled. The outer tag must match the value of the Pn_TS_VLAN_LTYPE1 field, and the inner tag must match the value of the Pn_TS_VLAN_LTYPE2 field.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 9-11. Partial Ethernet-II Frames Showing Register Mapping of EtherTypes for a Simple Frame (1), a Single 1Q Tag Added (2), and Two 1Q Tags Added (3)**




To enable transmit/receive event packets on a given Ethernet port, perform the following steps, where n is the port number:

1. Set the Pn_TS_TX_EN/Pn_TS_RX_EN bit in the Pn_TS_CTL register to enable transmit/receive event packets
2. Configure the Pn_TS_SEQ_LTYPE register:
   (a) Set the Pn_TS_LTYPE field to 88F7h, which corresponds to the Precision Time Protocol (IEEE 1588) EtherType.
   (b) Set the Pn_TS_SEQ_ID_OFFSET field to 1Eh, which is the sequence ID offset in the common message header given in the IEEE 1588 specification.
3. To enable support for VLAN tagging (IEEE 802.1Q):
   (a) Set the Pn_TS_TX_VLAN_LTYPE1_EN/Pn_TS_RX_VLAN_LTYPE1_EN or Pn_TS_TX_VLAN_LTYPE2_EN/Pn_TS_RX_VLAN_LTYPE2_EN bit in the Pn_TS_CTL register.
   (b) Set the Pn_TS_VLAN_LTYPE1 or Pn_TS_VLAN_LTYPE2 field (matching what was used in step 3a) in the Pn_TS_VLAN register to 8100h, which corresponds to the VLAN-tagged frame (IEEE 802.1Q) EtherType.
4. To enable support for up to two stacked VLANs (IEEE 802.1ad):
   (a) Set the Pn_TS_TX_VLAN_LTYPE1_EN/Pn_TS_RX_VLAN_LTYPE1_EN and Pn_TS_TX_VLAN_LTYPE2_EN/Pn_TS_RX_VLAN_LTYPE2_EN bit in the Pn_TS_CTL register.
   (b) Set the Pn_TS_VLAN_LTYPE1 field in the Pn_TS_VLAN register to match the EtherType of the outer tag.
   (c) Set the Pn_TS_VLAN_LTYPE2 field in the Pn_TS_VLAN register to 8100h, which corresponds to the VLAN-tagged frame (IEEE 802.1Q) EtherType of the inner tag.
5. Set the Pn_TS_MSG_TYPE_EN field in the Pn_TS_CTL register to choose which types of time stamp messages will push events onto the Event FIFO. Table 9-23 lists the message types defined in the IEEE 1588-2008 specification.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-23. Values of Message Type Field**

| Message Type | Value (hex) |
|---|---|
| Sync | 0 |
| Delay_Req | 1 |
| Pdelay_Req | 2 |
| Pdelay_Resp | 3 |
| Reserved | 4-7 |
| Follow_Up | 8 |
| Delay_Resp | 9 |
| Pdelay_Resp_Follow_Up | A |
| Announce | B |
| Signaling | C |
| Management | D |
| Reserved | E-F |

Once a transmitted or received packet is determined to be a valid time sync packet, the Ethernet Transmit Event or Ethernet Receive Event is loaded onto the Event FIFO. The CPTS_EVENT_HIGH register contains the Message Type and Sequence ID values from the original time sync packet. The CPTS_EVENT_LOW register contains the time stamp value when the packet arrived at the corresponding port.

#### 9.2.2.6 Interrupt Handling

When an event is push onto the Event FIFO, an interrupt can be generated to indicate to software that a time sync event occurred. The following steps should be taken to process time sync events using interrupts:

1. Enable the TS_PEND interrupt by setting the TS_PEND_EN bit of the CPTS_TS_INT_ENABLE register.
2. Upon interrupt, read the CPTS_EVENT_LOW and CPTS_EVENT_HIGH registers values.
3. Set the EVENT_POP field (bit 0) of the CPTS_EVENT_POP register to pop the previously read value off of the event FIFO.
4. Process the interrupt as required by the application software.

Software has the option of processing more than a single event from the event FIFO in the interrupt service routine in the following way:

1. Enable the TS_PEND interrupt by setting the TS_PEND_EN bit of the CPTS_TS_INT_ENABLE register.
2. Upon interrupt, read the CPTS_EVENT_LOW and CPTS_EVENT_HIGH registers values.
3. Set the EVENT_POP bit of the CPTS_EVENT_POP register to pop the previously read value off of the event FIFO.
4. Wait for an amount of time greater than four RCLK periods plus four VBUSP_CLK periods.
5. Read the TS_PEND_RAW bit in the CPTS_INTSTAT_RAW register to determine if another valid event is in the event FIFO. If it is asserted, go to step 2; otherwise, go to step 6.
6. Process the interrupt(s) as required by the application software.

Software also has the option of disabling the interrupt and polling the TS_PEND_RAW bit of the CPTS_INTSTAT_RAW register to determine if a valid event is on the event FIFO.

### *9.2.3 CPPI Buffer Descriptors*

The buffer descriptor is a central part of the 3PSW Ethernet Subsystem and is how the application software describes Ethernet packets to be sent and empty buffers to be filled with incoming packet data.


Copyright © 2011–2016, Texas Instruments Incorporated

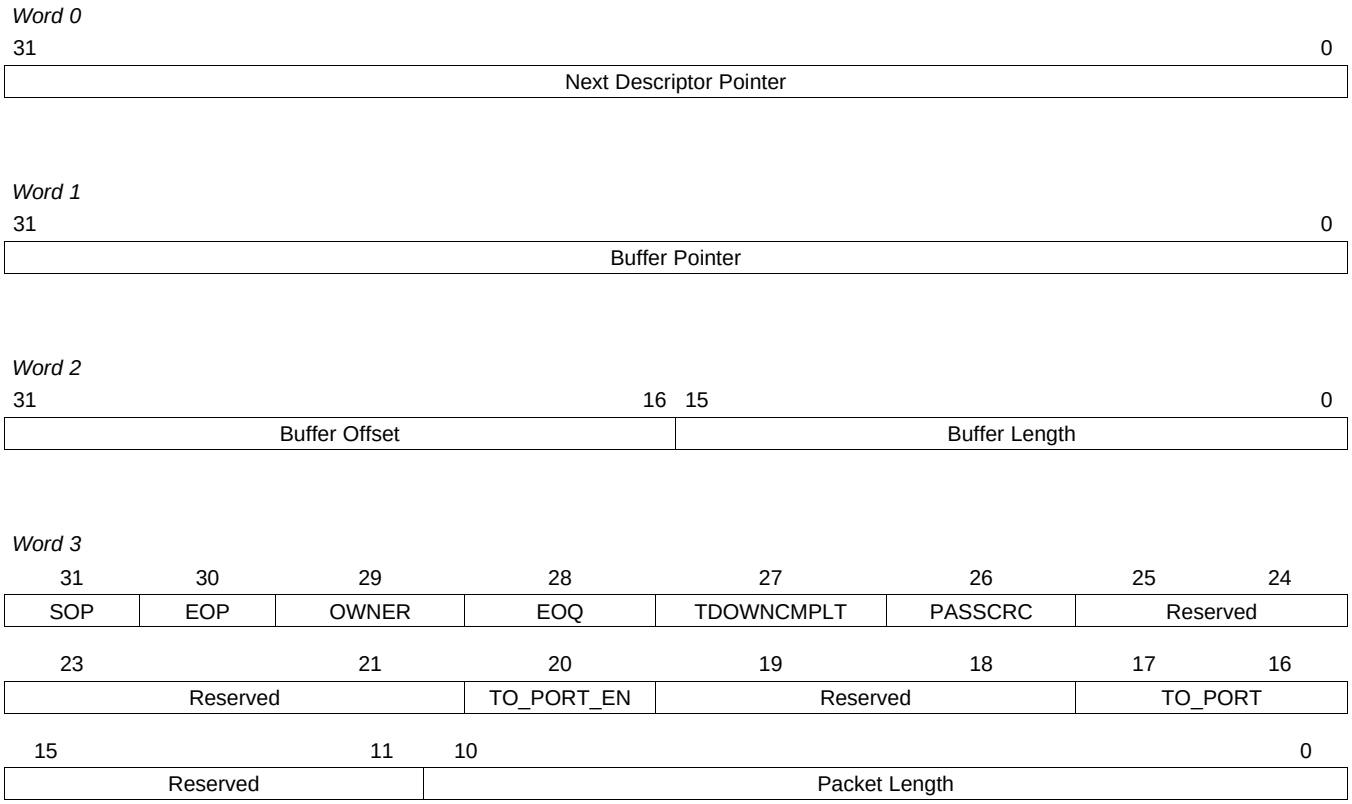
Host Software sends and receives network frames via the CPPI 3.0 compliant host interface. The host interface includes module registers and host memory data structures. The host memory data structures are buffer descriptors and data buffers. Buffer descriptors are data structures that contain information about a single data buffer. Buffer descriptors may be linked together to describe frames or queues of frames for transmission of data and free buffer queues available for received data.

**NOTE:** The 8K bytes of Ethernet Subsystem CPPI RAM begin at address 4A10 2000h and end at 4A10 3FFFh from the 3PSW perspective. The buffer descriptors programmed to access the CPPI RAM memory should use the address range from 4A10 2000h.

#### 9.2.3.1 TX Buffer Descriptors

A TX buffer descriptor (Figure 9-12) is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

**Figure 9-12. TX Buffer Descriptor Format**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.3.1.1 CPPI TX Data Word 0

**Next Descriptor Pointer**

The next descriptor pointer points to the 32-bit word aligned memory address of the next buffer descriptor in the transmit queue. This pointer is used to create a linked list of buffer descriptors. If the value of this pointer is zero, then the current buffer is the last buffer in the queue. The software application must set this value prior to adding the descriptor to the active transmit list. This pointer is not altered by the EMAC. The value of pNext should never be altered once the descriptor is in an active transmit queue, unless its current value is NULL. If the pNext pointer is initially NULL, and more packets need to be queued for transmit, the software application may alter this pointer to point to a newly appended descriptor. The EMAC will use the new pointer value and proceed to the next descriptor unless the pNext value has already been read. In this latter case, the transmitter will halt on the transmit channel in question, and the software application may restart it at that time. The software can detect this case by checking for an end of queue (EOQ) condition flag on the updated packet descriptor when it is returned by the EMAC

#### 9.2.3.1.2 CPPI TX Data Word 1

**Buffer Pointer**

The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the buffer_pointer. The software application must set this value prior to adding the descriptor to the active transmit list. This pointer is not altered by the EMAC.

#### 9.2.3.1.3 CPPI TX Data Word 2

**Buffer Offset**

Indicates how many unused bytes are at the start of the buffer. A value of 0000h indicates that no unused bytes are at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 000Fh (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The host sets the Buffer_Offset value (which may be zero to the buffer length minus 1). Valid only on SOP.

**Buffer Length**

Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host sets the Buffer_Length. The Buffer_Length must be greater than zero.

#### 9.2.3.1.4 CPPI TX Data Word 3

**Packet Length**

Specifies the number of bytes in the entire packet. Offset bytes are not included. The sum of the Buffer_Length fields should equal the Packet_Length. Valid only on SOP. The packet length must be greater than zero. The packet data will be truncated to the packet length if the packet length is shorter than the sum of the packet buffer descriptor buffer lengths. A host error occurs if the packet length is greater than the sum of the packet buffer descriptor buffer lengths.

**Start of Packet (SOP) Flag**

When set, this flag indicates that the descriptor points to a packet buffer that is the start of a new packet. In the case of a single fragment packet, both the SOP and end of packet (EOP) flags are set. Otherwise,,the descriptor pointing to the last packet buffer for the packet sets the EOP flag. This bit is set by the software application and is not altered by the EMAC.

0 - Not start of packet buffer.

1 - Start of packet buffer.

**End of Packet (EOP) Flag**


Copyright © 2011–2016, Texas Instruments Incorporated

When set, this flag indicates that the descriptor points to a packet buffer that is last for a given packet. In the case of a single fragment packet, both the start of packet (SOP) and EOP flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet sets the EOP flag. This bit is set by the software application and is not altered by the EMAC.

0 - Not end of packet buffer.

1 - End of packet buffer.

**Ownership (OWNER) Flag**

When set this flag indicates that all the descriptors for the given packet (from SOP to EOP) are currently owned by the EMAC. This flag is set by the software application on the SOP packet descriptor before adding the descriptor to the transmit descriptor queue. For a single fragment packet, the SOP, EOP, and OWNER flags are all set. The OWNER flag is cleared by the EMAC once it is finished with all the descriptors for the given packet. Note that this flag is valid on SOP descriptors only.

0 - The packet is owned by the host

1 - The packet is owned by the port

**End of Queue (EOQ) Flag**

When set, this flag indicates that the descriptor in question was the last descriptor in the transmit queue for a given transmit channel, and that the transmitter has halted. This flag is initially cleared by the software application prior to adding the descriptor to the transmit queue. This bit is set by the EMAC when the EMAC identifies that a descriptor is the last for a given packet (the EOP flag is set), and there are no more descriptors in the transmit list (next descriptor pointer is NULL).

The software application can use this bit to detect when the EMAC transmitter for the corresponding channel has halted. This is useful when the application appends additional packet descriptors to a transmit queue list that is already owned by the EMAC. Note that this flag is valid on EOP descriptors only.

0 - The TX queue has more packets to transfer.

1 - The Descriptor buffer is the last buffer in the last packet in the queue.

**Teardown Complete (TDOWNCMPLT) Flag**

This flag is used when a transmit queue is being torn down, or aborted, instead of allowing it to be transmitted. This would happen under device driver reset or shutdown conditions. The EMAC sets this bit in the SOP descriptor of each packet as it is aborted from transmission. Note that this flag is valid on SOP descriptors only. Also note that only the first packet in an unsent list has the TDOWNCMPLT flag set. Subsequent descriptors are not processed by the EMAC.

0 - The port has not completed the teardown process.

1 - The port has completed the commanded teardown process.

**Pass CRC (PASSCRC) Flag**

This flag is set by the software application in the SOP packet descriptor before it adds the descriptor to the transmit queue. Setting this bit indicates to the EMAC that the 4 byte Ethernet CRC is already present in the packet data, and that the EMAC should not generate its own version of the CRC. When the CRC flag is cleared, the EMAC generates and appends the 4-byte CRC. The buffer length and packet length fields do not include the CRC bytes. When the CRC flag is set, the 4-byte CRC is supplied by the software application and is already appended to the end of the packet data. The buffer length and packet length fields include the CRC bytes, as they are part of the valid packet data. Note that this flag is valid on SOP descriptors only.

0 - The CRC is not included with the packet data and packet length.

1 - The CRC is included with the packet data and packet length.

**TO_PORT**

Port number to send the directed packet to. This field is set by the host. This field is valid on SOP. Directed packets go to the directed port, but an ALE lookup is performed to determine untagged egress in VLAN_AWARE mode.


Copyright © 2011–2016, Texas Instruments Incorporated

1 - Send the packet to port 1 if TO_PORT_EN is asserted.

2 - Send the packet to port 2 if TO_PORT_EN is asserted.

**TO_PORT_ENABLE**

Indicates when set that the packet is a directed packet to be sent to the TO_PORT field port number. This field is set by the host. The packet is sent to one port only (index not mask). This bit is valid on SOP.

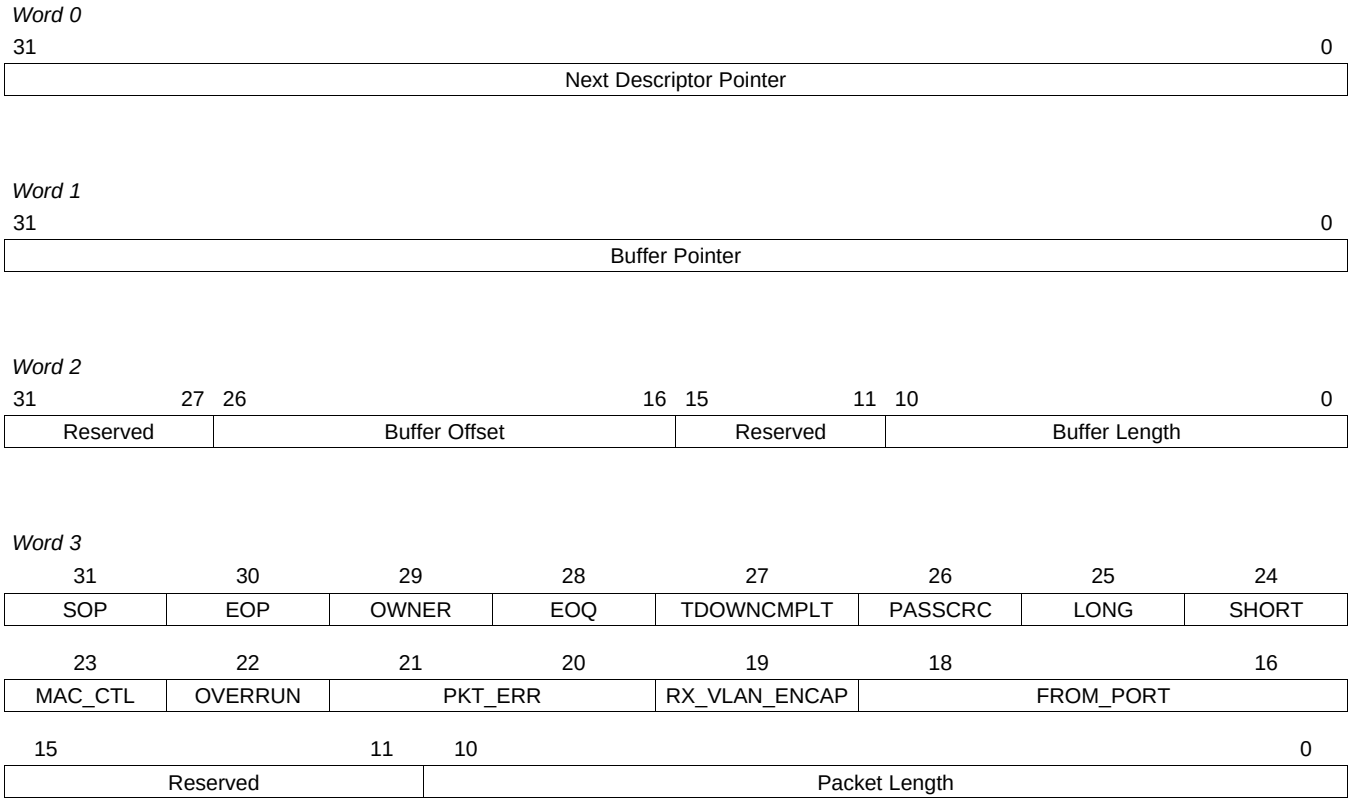
0 - not a directed packet

1 - directed packet

#### 9.2.3.2 RX Buffer Descriptors

A RX buffer descriptor (Figure 9-13) is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

**Figure 9-13. Receive Buffer Descriptor Format**




##### *9.2.3.2.1 CPPI RX Data Word 0*

**Next Descriptor Pointer**

The 32-bit word aligned memory address of the next buffer descriptor in the RX queue. This is the mechanism used to reference the next buffer descriptor from the current buffer descriptor. If the value of this pointer is zero then the current buffer is the last buffer in the queue. The host sets the Next_Descriptor_Pointer.

##### *9.2.3.2.2 CPPI RX Data Word 1*

**Buffer Pointer**

The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the Buffer_Pointer.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.3.2.3 CPPI RX Data Word 2

**Buffer Offset**

Indicates how many unused bytes are at the start of the buffer. A value of 0000h indicates that there are no unused bytes at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 000Fh (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The port writes the Buffer_Offset with the value from the RX_BUFFER_OFFSET register value. The host initializes the Buffer_Offset to zero for free buffers. The Buffer_Length must be greater than the RX_BUFFER_OFFSET register value. The buffer offset is valid only on SOP.

**Buffer Length**

Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host initializes the Buffer_Length, but the port may overwrite the host initiated value with the actual buffer length value on SOP and/or EOP buffer descriptors. SOP buffer length values will be overwritten if the packet size is less than the size of the buffer or if the offset is nonzero. EOP buffer length values will be overwritten if the entire buffer is not filled up with data. The Buffer_Length must be greater than zero.

#### 9.2.3.2.4 CPPI RX Data Word 3

**Packet Length**

Specifies the number of bytes in the entire packet. Offset bytes are not included. The sum of the Buffer_Length fields should equal the Packet_Length. Valid only on SOP.

**Start of Packet (SOP) Flag**

When set, this flag indicates that the descriptor points to a packet buffer that is the start of a new packet. In the case of a single fragment packet, both the SOP and end of packet (EOP) flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet has the EOP flag set. This flag is initially cleared by the software application before adding the descriptor to the receive queue. This bit is set by the EMAC on SOP descriptors.

**End of Packet (EOP) Flag**

When set, this flag indicates that the descriptor points to a packet buffer that is last for a given packet. In the case of a single fragment packet, both the start of packet (SOP) and EOP flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet has the EOP flag set. This flag is initially cleared by the software application before adding the descriptor to the receive queue. This bit is set by the EMAC on EOP descriptors.

**Ownership (OWNER) Flag**

When set, this flag indicates that the descriptor is currently owned by the EMAC. This flag is set by the software application before adding the descriptor to the receive descriptor queue. This flag is cleared by the EMAC once it is finished with a given set of descriptors, associated with a received packet. The flag is updated by the EMAC on SOP descriptor only. So when the application identifies that the OWNER flag is cleared on an SOP descriptor, it may assume that all descriptors up to and including the first with the EOP flag set have been released by the EMAC. (Note that in the case of single buffer packets, the same descriptor will have both the SOP and EOP flags set.)

**End of Queue (EOQ) Flag**

When set, this flag indicates that the descriptor in question was the last descriptor in the receive queue for a given receive channel, and that the corresponding receiver channel has halted. This flag is initially cleared by the software application prior to adding the descriptor to the receive queue. This bit is set by the EMAC when the EMAC identifies that a descriptor is the last for a given packet received (also sets the EOP flag), and there are no more descriptors in the receive list (next descriptor pointer is NULL).The software application can use this bit to detect when the EMAC receiver for the corresponding channel has halted. This is useful when the application appends additional free buffer descriptors to an active receive queue. Note that this flag is valid on EOP descriptors only.

**Teardown Complete (TDOWNCMPLT) Flag**


Copyright © 2011–2016, Texas Instruments Incorporated

This flag is used when a receive queue is being torn down, or aborted, instead of being filled with received data. This would happen under device driver reset or shutdown conditions. The EMAC sets this bit in the descriptor of the first free buffer when the tear down occurs. No additional queue processing is performed.

**Pass CRC (PASSCRC) Flag**

This flag is set by the EMAC in the SOP buffer descriptor if the received packet includes the 4-byte CRC. This flag should be cleared by the software application before submitting the descriptor to the receive queue.

**Long (Jabber) Flag**

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is a jabber frame and was not discarded because the RX_CEF_EN bit was set in the MACCONTROL register. Jabber frames are frames that exceed the RXMAXLEN in length, and have CRC, code, or alignment errors.

**Short (Fragment) Flag**

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is only a packet fragment and was not discarded because the RX_CSF_EN bit was set in the MACCONTROL register.

**Control Flag**

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is an EMAC control frame and was not discarded because the RX_CMF_EN bit was set in the MACCONTROL register.

**Overrun Flag**

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet was aborted due to a receive overrun.

**Packet Error (PKT_ERR) Flag**

Packet Contained Error on Ingress:

00 - no error

01 - CRC error on ingress

10 - Code error on ingress

11 - Align error on ingress

**VLAN Encapsulated Packet (RX_VLAN_ENCAP)**

Indicates when set that the packet data contains a 32-bit VLAN header word that is included in the packet byte count. This field is set by the port to be the value of the CPSW control register RX_VLAN_ENCAP bit.

**FROM_PORT**

Indicates the port number that the packet was received on (ingress to the switch).


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.2.4 MDIO

The MII Management I/F module implements the 802.3 serial management interface to interrogate and control two Ethernet PHYs simultaneously using a shared two-wire bus. Two user access registers control and monitor up to two PHYs simultaneously.

#### 9.2.4.1 MII Management Interface Frame Formats

Table 9-24 shows the read format and Table 9-25 shows the write format of the 32-bit MII Management interface frames.

**Table 9-24. MDIO Read Frame Format**

| Pre-amble | Start Delimiter | Operation Code | PHY Address | Register Address | Turnaround | Data |
|---|---|---|---|---|---|---|
| FFFF FFFFh | 01 | 10 | AAAAA | RRRRR | Z0 | DDDD.DDDD.DDDD.DDDD |

**Table 9-25. MDIO Write Frame Format**

| Pre-amble | Start Delimiter | Operation Code | PHY Address | Register Address | Turnaround | Data |
|---|---|---|---|---|---|---|
| FFFF FFFFh | 01 | 01 | AAAAA | RRRRR | 10 | DDDD.DDDD.DDDD.DDDD |

The default or idle state of the two wire serial interface is a logic one. All tri-state drivers should be disabled and the PHY's pull-up resistor will pull the MDIO line to a logic 1. Prior to initiating any other transaction, the station management entity shall send a preamble sequence of 32 contiguous logic 1 bits on the MDIO line with 32 corresponding cycles on MDCLK to provide the PHY with a pattern that it can use to establish synchronization. A PHY shall observe a sequence of 32 contiguous logic one bits on MDIO with 32 corresponding MDCLK cycles before it responds to any other transaction.

**Preamble**

The start of a frame is indicated by a preamble, which consists of a sequence of 32 contiguous bits all of which are a 1. This sequence provides the PHY a pattern to use to establish synchronization.

**Start Delimiter**

The preamble is followed by the start delimiter which is indicated by a 01 pattern. The pattern assures transitions from the default logic 1 state to logic 0, and back to logic 1.

**Operation Code**

The operation code for a read is 10, while the operation code for a write is a 01.

**PHY Address**

The PHY address is 5 bits allowing 32 unique values. The first bit transmitted is the MSB of the PHY address.

**Register Address**

The Register address is 5 bits allowing 32 registers to be addressed within each PHY. Refer to the 10/100 PHY address map for addresses of individual registers.

**Turnaround**

An idle bit time during which no device actively drives the MDIO signal shall be inserted between the register address field and the data field of a read frame in order to avoid contention. During a read frame, the PHY shall drive a zero bit onto MDIO for the first bit time following the idle bit and preceding the Data field. During a write frame, this field shall consist of a one bit followed by a zero bit.

**Data**

The Data field is 16 bits. The first bit transmitted and received is the MSB of the data word.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.4.2 Functional Description

The MII Management I/F will remain idle until enabled by setting the enable bit in the MDIO CONTROL register. The MII Management I/F will then continuously poll the link status from within the Generic Status Register of all possible 32 PHY addresses in turn recording the results in the MDIO LINK register. The LINKSEL bit in the MDIO USERPHYSEL register determines the status input that is used. A change in the link status of the two PHYs being monitored will set the appropriate bit in the MDIO LINKINTRAW register and the MDIO LINKINTMASKED register, if enabled by the LINKINT_ENABLE bit in the MDIO USERPHYSEL register.

The MDIO ALIVE register is updated by the MII Management I/F module if the PHY acknowledged the read of the generic status register. In addition, any PHY register read transactions initiated by the host also cause the MDIO ALIVE register to be updated.

At any time, the host can define a transaction for the MII Management interface module to undertake using the DATA, PHYADR, REGADR, and WRITE fields in a MDIO USERACCESS register. When the host sets the GO bit in this register, the MII Management interface module will begin the transaction without any further intervention from the host. Upon completion, the MII Management interface will clear the GO bit and set the USERINTRAW bit in the MDIO USERINTRAW register corresponding to the MDIO USERACCESS register being used. The corresponding bit in the MDIO USERINTMASKED register may also be set depending on the mask setting in the MDIO USERINTMASKSET and MDIO USERINTMASKCLR registers. A round-robin arbitration scheme is used to schedule transactions that may be queued by the host in different MDIO USERACCESS registers. The host should check the status of the GO bit in the MDIO USERACCESS register before initiating a new transaction to ensure that the previous transaction has completed. The host can use the ACK bit in the MDIO USERACCESS register to determine the status of a read transaction.

It is necessary for software to use the MII Management interface module to setup the auto-negotiation parameters of each PHY attached to a MAC port, retrieve the negotiation results, and setup the MACCONTROL register in the corresponding MAC.


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.2.5 Interrupts

The 3Port Switch Ethernet Subsystem generates four interrupt events.

#### 9.2.5.1 Receive Packet Completion Pulse Interrupt (RX_PULSE)

The RX_PULSE interrupt is a pulse interrupt selected from the 3PSW RX_PEND[7:0] interrupts. The receive DMA controller has eight channels with each channel having a corresponding (RX_PEND[7:0]).

The following steps will enable the receive packet completion interrupt:

1. Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the RX_INTMASK_SET register.
2. The receive completion interrupt(s) to be routed to RX_PULSE is selected by setting one or more bits in the receive interrupt enable register (RX_EN) . The masked interrupt status can be read in the address location of RX_STAT bit in the receive status register.

When the 3PSW completes a packet reception, the subsystem issues an interrupt to the CPU by writing the packet's last buffer descriptor address to the appropriate channel queue's receive completion pointer located in the state RAM block. The interrupt is generated by the write when enabled by the interrupt mask, regardless of the value written.

Upon interrupt reception, the CPU processes one or more packets from the buffer chain and then acknowledges one or more interrupt(s) by writing the address of the last buffer descriptor processed to the queue's associated receive completion pointer (RX *n*_CP) in the receive DMA state RAM.

Upon reception of an interrupt, software should perform the following:

1. Read the RX_STAT bit address location to determine which channel(s) caused the interrupt.
2. Process received packets for the interrupting channel(s).
3. Write the 3PSW completion pointer(s) (RX *n*_CP). The data written by the host (buffer descriptor address of the last processed buffer) is compared to the data in the register written by the subsystem (address of last buffer descriptor used by the subsystem). If the two values are not equal (which means that the 3PSW has received more packets than the CPU has processed), the receive packet completion interrupt signal remains asserted. If the two values are equal (which means that the host has processed all packets that the system has received), the pending interrupt is de-asserted. The value that the 3PSW is expecting is found by reading the receive channel *n* completion pointer register (RX *n*_CP).
4. Write the value 1h to the CPDMA_EOI_VECTOR register.

To disable the interrupt:

1. The eight channel interrupts may be individually disabled by writing to 1 the appropriate bit in the RX_INTMASK_CLEAR register.
2. The receive completion pulse interrupt could be disabled by clearing to 0 all the bits in the RX_EN register.

The software could still poll for the RX_INTSTAT_RAW and RX_INTSTAT_MASKED registers if the corresponding interrupts are enabled.

#### 9.2.5.2 Transmit Packet Completion Pulse Interrupt (TX_PULSE)

The TX_PULSE interrupt is a pulse interrupt selected from the 3PSW TX_PEND [7:0] interrupts. The transmit DMA controller has eight channels with each channel having a corresponding (TX_PEND[7:0]).

To enable the transmit packet completion interrupt:

1. Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the TX_INTMASK_SET register.
2. The transmit completion interrupt(s) to be routed to TX_PULSE is selected by setting one or more bits in the transmit interrupt enable register TX_EN .The masked interrupt status can be read in the address location of TX_STAT bit in the transmit status register.


Copyright © 2011–2016, Texas Instruments Incorporated

When the 3PSW completes the transmission of a packet, the 3PSW subsystem issues an interrupt to the CPU by writing the packet's last buffer descriptor address to the appropriate channel queue's transmit completion pointer located in the state RAM block. The interrupt is generated by the write when enabled by the interrupt mask, regardless of the value written.

Upon reception of an interrupt, software should perform the following:

1. Read the TX_STAT bit address location to determine which channel(s) caused the interrupt
2. Process received packets for the interrupting channel(s).
3. Write the 3PSW completion pointer(s) (TX *n*_CP). The data written by the host (buffer descriptor address of the last processed buffer) is compared to the data in the register written by the 3PSW (address of last buffer descriptor used by the 3PSW). If the two values are not equal (which means that the 3PSW has transmitted more packets than the CPU has processed), the transmit packet completion interrupt signal remains asserted. If the two values are equal (which means that the host has processed all packets that the subsystem has transferred), the pending interrupt is cleared. The value that the 3PSW is expecting is found by reading the transmit channel *n*completion pointer register (TX *n*_CP).
4. Write the 2h to the CPDMA_EOI_VECTOR register.

To disable the interrupt:

1. The eight channel interrupts may be individually disabled by writing to 1 the appropriate bit in the TX_INTMASK_CLEAR register.
2. The receive completion pulse interrupt could be disabled by clearing to 0 all the bits in the TX_EN register. The software could still poll for the TX_INTSTAT_RAW and TX_INTSTAT_MASKED registers, if the corresponding interrupts are enabled.

#### 9.2.5.3 Receive Threshold Pulse Interrupt (RX_THRESH_PULSE)

The RX_THRESH_PULSE interrupt is an immediate (non-paced) pulse interrupt selected from the CPSW_3G RX_THRESH_PEND[7:0] interrupts. The receive DMA controller has eight channels with each channel having a corresponding threshold pulse interrupt (RX_THRESH_PEND[7:0]).

To enable the receive threshold pulse Interrupt:

1. Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the RX_INTMASK_SET register.
2. he receive threshold interrupt(s) to be routed to RX_THRESH_PULSE is selected by setting one or more bits in the receive threshold interrupt enable register RX_THRESH_EN. The masked interrupt status can be read in the address location of RX_THRESH_STAT bit in the receive threshold status register.

The RX_THRESH_PULSE is asserted when enabled when the channel's associated free buffer count RX *n*_FREEBUFFER is less than or equal to the corresponding RX *n*_PENDTHRESH register.

Upon reception of an interrupt, software should perform the following:

1. Read the RX_THRESH_STAT bit address location to determine which channel(s) caused the interrupt.
2. Process the received packets in order to add more buffers to any channel that is below the threshold value.
3. Write the CPSW_3G completion pointer(s).
4. Write the value 0h to the CPDMA_EOI_VECTOR register.

The threshold pulse interrupt is an immediate interrupt intended to indicate that software should immediately process packets to preclude an overrun condition from occurring for the particular channels.

To disable the interrupt:

1. The eight channel receive threshold interrupts may be individually disabled by writing to 1 the appropriate bit in the RX_INTMASK_CLEAR register.
2. The receive threshold pulse interrupt could be disabled by clearing to Zero the corresponding bits in the RX_THRESH_EN register. The software could still poll for the RX_INTSTAT_RAW and INTSTAT_MASKED registers, if the corresponding interrupts are enabled.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.5.4 Miscellaneous Pulse Interrupt (MISC_PULSE)

The MISC_PULSE interrupt is an immediate pulse interrupt selected from the miscellaneous interrupts (EVNT_PEND,STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT). The miscellaneous interrupt(s) is selected by setting one or more bits in the MISC_EN register. The masked interrupt status can be read in the MISC_STAT address location. Upon reception of an interrupt, software should perform the following:

1. Read the MISC_STAT address location to determine which caused the interrupt.
2. Process the interrupt.
3. Write the appropriate value (3h) to the CPDMA_EOI_VECTOR register.

**NOTE:** The MISC_STAT register's MDIO_LINKINT and MDIO_USERINT bitfields represent the Port0/Phy0 status. MDIO_LINKINT[1] and MDIO_USERINT[1] are not provided in MISC_STAT register. As such, MDIO Link and User interrupts can only be generated for Port0. For Port1, software must poll the status, visible in the LINKINTMASKED or USERINTMASKED registers.

##### *9.2.5.4.1 EVNT_PEND( CPTS_PEND) Interrupt*

See Section 9.2.2 for more details on this interrupt.

##### *9.2.5.4.2 Statistics Interrupt*

The statistics level interrupt (STAT_PEND) will be asserted, if enabled when any statistics value is greater than or equal to 8000 0000h. The statistics interrupt is cleared by writing to decrement all statistics values greater than 8000 0000h (such that their new values are less than 8000 0000h). The raw and masked statistics interrupt status may be read by reading the TX_INTSTAT_RAW and TX_INTSTAT_MASKED registers, respectively.

##### *9.2.5.4.3 Host Error interrupt*

The host error interrupt (HOST_PEND) will be asserted, if enabled when a host error is detected during transmit or receive CPDMA transactions. The host error interrupt is intended for software debug, and is cleared by a warm reset or a system reset. The raw and masked statistics interrupt status can be read by reading the TX_INTSTAT_RAW and TXINTSTAT_MASKED registers, respectively.

The transmit host error conditions are:

- SOP error
- OWNERSHIP bit not set in SOP buffer
- next buffer descriptor pointer without EOP cleared to 0
- buffer pointer cleared to 0
- buffer length cleared to 0
- packet length error

The receive host error conditions are:

- OWNERSHIP bit not set in input buffer
- Zero buffer pointer
- Zero buffer Length on non-SOP descriptor
- SOP buffer length not greater than offset

The host error interrupt is disabled by clearing to 0 the appropriate bit in the CPDMA_INTMASK_CLEAR register.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.2.5.4.4 MDIO Interrupts

MDIO_LINKINT is set if there is a change in the link state of the PHY corresponding to the address in the PHYADR_MON field of the USERPHYSEL*n* register and the corresponding LINKINT_ENABLE bit is set. The MDIO_LINKINT event is also captured in the LINKINTMASKED register. When the GO bit in the USERACCESS*n* register transitions from 1 to 0, indicating the completion of a user access, and the corresponding USERINTMASKSET bit in the USERINTMASKSET register is set, the MDIO_USERINT signal is asserted. The MDIO_USERINT event is also captured in the USERINTMASKED register.

To enable the miscellaneous pulse interrupt:

The miscellaneous interrupt(s) is selected by setting one or more bits in the miscellaneous interrupt enable register (MISC_EN).

- The Statistics interrupt is enabled by setting to 1 the STAT_INT_MASK bit in the DMA_INTMASK_SET register.
- The HOST_PEND is enabled by setting to 1 the HOST_ERR_INTMASK in the DMA_INTMASK_SET register.

Upon reception of an interrupt, software should perform the following:

- Read the MISC_STAT bit address location to determine the source of the interrupt.
- Process the interrupt.
- Write the value 3h to the CPDMA_EOI_VECTOR register.


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.2.6 Interrupt Pacing

The receive and transmit pulse interrupts can be paced. The receive threshold and miscellaneous interrupts are not paced. The Interrupt pacing feature limits the number of interrupts that occur during a given period of time. For heavily loaded systems in which interrupts can occur at a very high rate (for example, 148,800 packets per second for Ethernet), the performance benefit is significant due to minimizing the overhead associated with servicing each interrupt. Interrupt pacing increases the CPU cache hit ratio by minimizing the number of times that large interrupt service routines are moved to and from the CPU instruction cache. Each CPU receive and transmit pulse interrupt contains an interrupt pacing sub-block (six total). Each sub-block is disabled by default allowing the selected interrupt inputs to pass through unaffected. The interrupt pacing module counts the number of interrupts that occur over a 1 ms interval of time. At the end of each 1 ms interval, the current number of interrupts is compared with a target number of interrupts (specified by the associated maximum number of interrupts register). Based on the results of the comparison, the length of time during which interrupts are blocked is dynamically adjusted. The 1 ms interval is derived from a 4 µs pulse that is created from a prescale counter whose value is set in the INT_PRESCALE field in the INT_CONTROL register. The INT_PRESCALE value should be written with the number of VBUSP_CLK periods in 4 µs. The pacing timer determines the interval during which interrupts are blocked and decrements every 4 µs. It is reloaded each time a zero count is reached. The value loaded into the pacing timer is calculated by hardware every 1 ms according to the following algorithm:

```
if (intr_count > 2*intr_max)
    pace_timer = 255;
  else if (intr_count > 1.5*intr_max)
    pace_timer = last_pace_timer*2 + 1;
  else if (intr_count > 1.0*intr_max)
    pace_timer = last_pace_timer + 1;
  else if (intr_count > 0.5*intr_max)
    pace_timer = last_pace_timer - 1;
  else if (intr_count != 0)
    pace_timer = last_pace_timer/2;
else
pace_timer = 0;
```

If the rate of interrupt inputs is much less than the target interrupt rate specified in the associated maximum interrupts register, then the interrupt is not blocked. If the interrupt rate is greater than the target rate, the interrupt will be "paced" at the rate specified in the interrupt maximum register. The interrupt maximum register should be written with a value between 2 and 63 inclusive, indicating the target number of interrupts per millisecond.


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.2.7 Reset Isolation

Reset isolation for the Ethernet switch on Device is that the switch function of the ethernet IP remains active even in case of all device resets except for POR pin reset and ICEPICK COLD reset. Packet traffic to/from the 3PSW host will be flushed/dropped, but the ethernet switch will remain operational for all traffic between external devices on the switch even though the device is undergoing a device reset. Pin mux configuration for ethernet related IO and reference clocks needed by the Ethernet switch IP to be active is controlled by a protected control module bit. If reset isolation is enabled, then only a POR pin or ICEPICK COLD reset event should fully reset the Ethernet switch IP including the actual switch and also the reference clocks and pin mux control specifically associated with the Ethernet IP.

#### 9.2.7.1 Modes of Operation

The device has two modes of operation concerning the reset of the 3PSW Ethernet switch. The mode is controlled by the ISO_CONTROL bit in the RESET_ISO register of the device Control Module. This bit defaults to 0. Any modification of this bit first requires writing an unlock pattern to lock register in device control module. After modification, the bit should again be locked by writing appropriate value to the lock register. Writes to the ISO_CONTROL bit and to the lock register must all be supervisor mode writes.

**Mode1: ISO_CONTROL = 0 (reset isolation disabled)**

1. This mode is selected when the ISO_CONTROL bit = 0. This should be the default state of the bit after control module reset.
2. Upon any device level resets, the entire CPSW_3GSS_R IP, SATA_SS, SATA SERDES PLL, L3/L4, control module (including all pin mux control and the ISO_CONTROL bit) is immediately reset.

**Mode2: ISO_CONTROL = 1 (reset isolation enabled)**

1. This mode is selected when the ISO_CONTROL bit = 1.
2. Upon any device reset source other than POR pin or ICEPICK cold (so this includes SW global cold, any watchdog reset, warm RESETn pin, ICEPICK warm, SW global warm or security violation), the following should be true:
   (a) The CPSW_3GSS_R is put into "isolate" mode and non-switch related portions of the IP are reset.
   (b) The 50 MHz and 125 MHz reference clocks to the 3PSW Ethernet Subsystem remains active throughout the entire reset condition.
   (c) The control for pin multiplexing for all of the signals should maintain their current configuration throughout the entire reset condition.
   (d) The reset isolated logic inside 3PSW Ethernet Subsystem IP which maintains the switch functionality
3. Upon any cold reset sources, the entire 3PSW Ethernet Subsystem, SATA SERDES PLL, control module (including all pin mux control and the ISO_CONTROL bit itself) is reset.

For more details on the register configuration, see the Control Module RESET_ISO register.

### 9.2.8 CPSW_3G Network Statistics

The CPSW_3G has a set of statistics that record events associated with frame traffic on selected switch ports. The statistics values are cleared to zero 38 clocks after the rising edge of VBUSP_RST_N. When one or more port enable (Pn_STAT_EN) bits in the CPSW_STAT_PORT_EN register are set, all statistics registers are write to decrement. The value written will be subtracted from the register value with the result being stored in the register. If a value greater than the statistics value is written, then zero will be written to the register (writing FFFF FFFFh clears a statistics location). When all port enable bits are cleared to zero, all statistics registers are read/write (normal write direct, so writing 0000 0000h clears a statistics location). All write accesses must be 32-bit accesses. In the below statistics descriptions, "the port" refers to any enabled port (with a corresponding set Pn_STAT_EN bit).

The statistics interrupt (STAT_PEND) will be issued if enabled when any statistics value is greater than or equal to 8000 0000h. The statistics interrupt is removed by writing to decrement any statistics value greater than 8000 0000h. The statistics are mapped into internal memory space and are 32-bits wide. All statistics rollover from FFFF FFFFh to 0000 0000h.

See Section 9.4 for more details on the registers.


Copyright © 2011–2016, Texas Instruments Incorporated

## 9.3 Software Operation

### 9.3.1 Transmit Operation

After reset, the host must write zeroes to all TX DMA State head descriptor pointers. The TX port may then be enabled. To initiate packet transmission the host constructs transmit queues in memory (one or more packets for transmission) and then writes the appropriate TX DMA state head descriptor pointers. For each buffer added to a transmit queue, the host must initialize the TX buffer descriptor values as follows:

1. Write the Next Descriptor Pointer with the 32-bit aligned address of the next descriptor in the queue (zero if last descriptor)
2. Write the Buffer Pointer with the byte aligned address of the buffer data
3. Write the Buffer Length with the number of bytes in the buffer
4. Write the Buffer Offset with the number of bytes in the offset to the data (nonzero with SOP only)
5. Set the SOP, EOP, and Ownership bits as appropriate
6. Clear the End Of Queue bit

The port begins TX packet transmission on a given channel when the host writes the channel's TX queue head descriptor pointer with the address of the first buffer descriptor in the queue (nonzero value). Each channel may have one or more queues, so each channel may have one or more head descriptor pointers. The first buffer descriptor for each TX packet must have the Start of Packet (SOP) bit and the Ownership bit set to one by the host. The last buffer descriptor for each TX packet must have the End of Packet (EOP) bit set to one by the host. The port will transmit packets until all queued packets have been transmitted and the queue(s) are empty. When each packet transmission is complete, the port will clear the Ownership bit in the packet's SOP buffer descriptor and issue an interrupt to the host by writing the packet's last buffer descriptor address to the queue's TX DMA State Completion Pointer. The interrupt is generated by the write, regardless of the value written. When the last packet in a queue has been transmitted, the port sets the End Of Queue bit in the EOP buffer descriptor, clears the Ownership bit in the SOP Descriptor, zeroes the appropriate DMA state head descriptor pointer, and then issues a TX interrupt to the host by writing to the queue's associated TX completion pointer (address of the last buffer descriptor processed by the port). The port issues a maskable level interrupt (which may then be routed through external interrupt control logic to the host).

On interrupt from the port, the host processes the buffer queue, detecting transmitted packets by the status of the Ownership bit in the SOP buffer descriptor. If the Ownership bit is cleared to zero, then the packet has been transmitted and the host may reclaim the buffers associated with the packet. The host continues queue processing until the end of the queue or until a SOP buffer descriptor is read that contains a set Ownership bit indicating that the packet transmission is not complete. The host determines that all packets in the queue have been transmitted when the last packet in the queue has a cleared Ownership bit in the SOP buffer descriptor, the End of Queue bit is set in the last packet EOP buffer descriptor, and the Next Descriptor Pointer of the last packet EOP buffer descriptor is zero. The host acknowledges an interrupt by writing the address of the last buffer descriptor to the queue's associated TX Completion Pointer in the TX DMA State. If the host written buffer address value is different from the buffer address written by the port, then the level interrupt remains asserted. If the host written buffer address value is equal to the port written value, then the level interrupt is de-asserted. The port write to the completion pointer actually stores the value in the state register (ram). The host written value is actually not written to the register location. The host written value is compared to the register contents (which was written by the port) and if the two values are equal, the interrupt is removed, otherwise the interrupt remains asserted. The host may process multiple packets previous to acknowledging an interrupt, or the host may acknowledge interrupts for every packet.

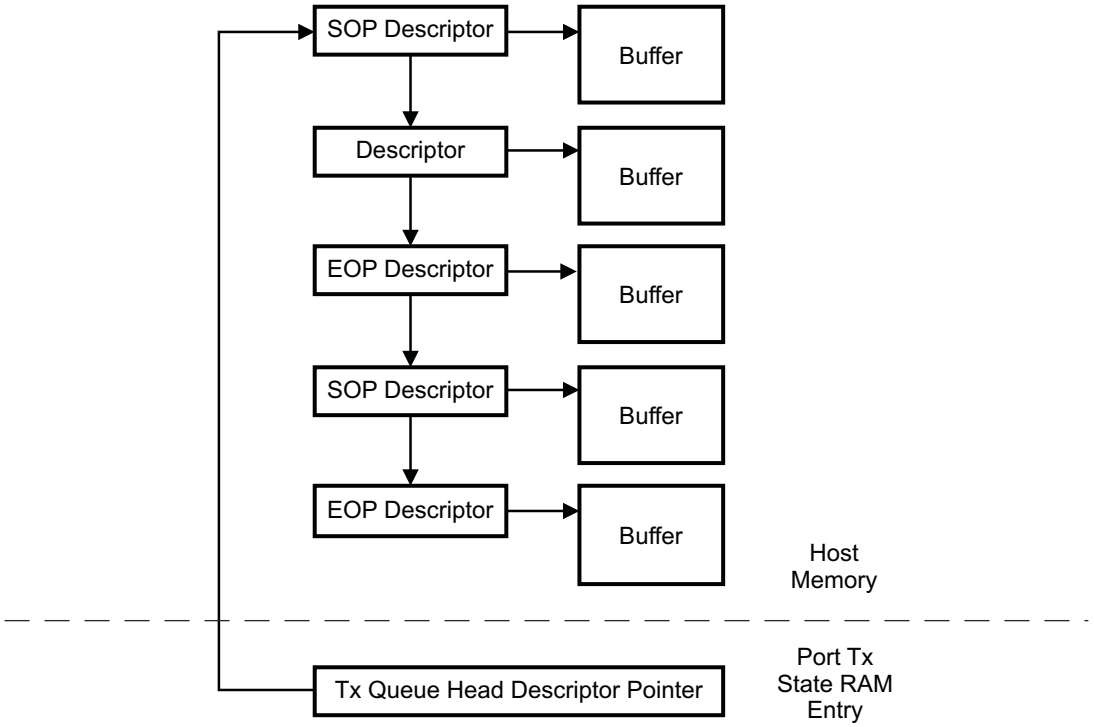
A mis-queued packet condition may occur when the host adds a packet to a queue for transmission as the port finishes transmitting the previous last packet in the queue. The mis-queued packet is detected by the host when queue processing detects a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and a nonzero Next Descriptor Pointer in the EOP buffer descriptor. A mis-queued packet means that the port read the last EOP buffer descriptor before the host added the new last packet to the queue, so the port determined queue empty just before the last packet was added. The host corrects the mis-queued packet condition by initiating a new packet transfer for the mis-queued packet by writing the mis-queued packet's SOP buffer descriptor address to the appropriate DMA State TX Queue head Descriptor Pointer.


Copyright © 2011–2016, Texas Instruments Incorporated

The host may add packets to the tail end of an active TX queue at any time by writing the Next Descriptor Pointer to the current last descriptor in the queue. If a TX queue is empty (inactive), the host may initiate packet transmission at any time by writing the appropriate TX DMA State head descriptor pointer. The host software should always check for and reinitiate transmission for mis-queued packets during queue processing on interrupt from the port. In order to preclude software underrun, the host should avoid adding buffers to an active queue for any TX packet that is not complete and ready for transmission.

The port determines that a packet is the last packet in the queue by detecting the End of Packet bit set with a zero Next Descriptor Pointer in the packet buffer descriptor. If the End of Packet bit is set and the Next Descriptor Pointer is nonzero, then the queue still contains one or more packets to be transmitted. If the EOP bit is set with a zero Next Descriptor Pointer, then the port will set the EOQ bit in the packet's EOP buffer descriptor and then zero the appropriate head descriptor pointer previous to interrupting the port (by writing the completion pointer) when the packet transmission is complete.

**Figure 9-14. TX Queue Head Descriptor**





Copyright © 2011–2016, Texas Instruments Incorporated

### 9.3.2 Receive Operation

After reset, the host must write zeroes to all RX DMA State head descriptor pointers. The RX port may then be enabled. To initiate packet reception, the host constructs receive queues in memory and then writes the appropriate RX DMA state head descriptor pointer. For each RX buffer descriptor added to the queue, the host must initialize the RX buffer descriptor values as follows:

1. Write the Next Descriptor Pointer with the 32-bit aligned address of the next descriptor in the queue (zero if last descriptor)
2. Write the Buffer Pointer with the byte aligned address of the buffer data
3. Clear the Offset field
4. Write the Buffer Length with the number of bytes in the buffer
5. Clear the SOP, EOP, and EOQ bits
6. Set the Ownership bit

The host enables packet reception on a given channel by writing the address of the first buffer descriptor in the queue (nonzero value) to the channel's head descriptor pointer in the channel's RX DMA state. When packet reception begins on a given channel, the port fills each RX buffer with data in order starting with the first buffer and proceeding through the RX queue. If the Buffer Offset in the RX DMA State is nonzero, then the port will begin writing data after the offset number of bytes in the SOP buffer. The port performs the following operations at the end of each packet reception:

1. Overwrite the buffer length in the packet's EOP buffer descriptor with the number of bytes actually received in the packet's last buffer. The host initialized value is the buffer size. The overwritten value will be less than or equal to the host initialized value.
2. Set the EOP bit in the packet's EOP buffer descriptor.
3. Set the EOQ bit in the packet's EOP buffer descriptor if the current packet is the last packet in the queue.
4. Overwrite the packet's SOP buffer descriptor Buffer Offset with the RX DMA state value (the host initialized the buffer descriptor Buffer Offset value to zero). All non SOP buffer descriptors must have a zero Buffer Offset initialized by the host.
5. Overwrite the packet's SOP buffer descriptor buffer length with the number of valid data bytes in the buffer. If the buffer is filled up, the buffer length will be the buffer size minus buffer offset.
6. Set the SOP bit in the packet's SOP buffer descriptor.
7. Write the SOP buffer descriptor Packet Length field.
8. Clear the Ownership bit in the packet's SOP buffer descriptor.
9. Issue an RX host interrupt by writing the address of the packet's last buffer descriptor to the queue's RX DMA State Completion Pointer. The interrupt is generated by the write to the RX DMA State Completion Pointer address location, regardless of the value written.

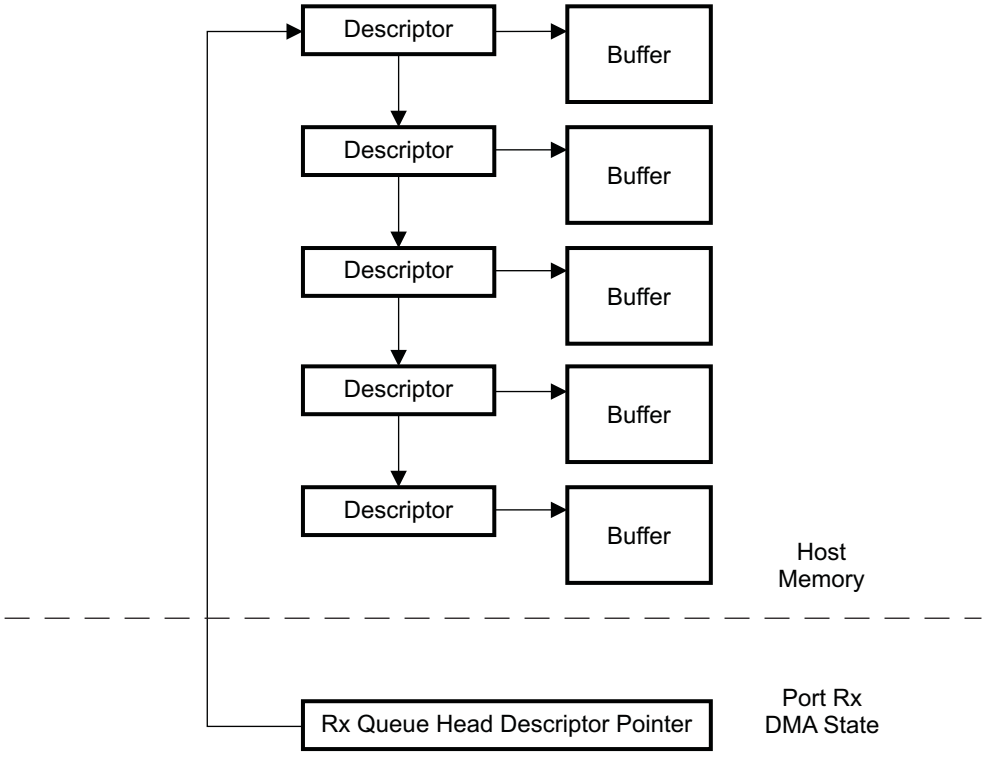
On interrupt the host processes the RX buffer queue detecting received packets by the status of the Ownership bit in each packet's SOP buffer descriptor. If the Ownership bit is cleared then the packet has been completely received and is available to be processed by the host. The host may continue RX queue processing until the end of the queue or until a buffer descriptor is read that contains a set Ownership bit indicating that the next packet's reception is not complete. The host determines that the RX queue is empty when the last packet in the queue has a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and the Next Descriptor Pointer in the EOP buffer descriptor is zero.

A mis-queued buffer may occur when the host adds buffers to a queue as the port finishes the reception of the previous last packet in the queue. The mis-queued buffer is detected by the host when queue processing detects a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and a nonzero Next Descriptor Pointer in the EOP buffer descriptor. A mis-queued buffer means that the port read the last EOP buffer descriptor before the host added buffer descriptor(s) to the queue, so the port determined queue empty just before the host added more buffer descriptor(s). In the transmit case, the packet transmission is delayed by the time required for the host to determine the condition and reinitiate the transaction, but the packet is not actually lost. In the receive case, receive overrun condition may occur in the mis-queued buffer case. If a new packet reception is begun during the time that the port has determined the end of queue condition, then the received packet will overrun (start


Copyright © 2011–2016, Texas Instruments Incorporated

of packet overrun). If the mis-queued buffer occurs during the middle of a packet reception then middle of packet overrun may occur. If the mis-queued buffer occurs after the last packet has completed, and is corrected before the next packet reception begins, then overrun will not occur. The host acts on the mis-queued buffer condition by writing the added buffer descriptor address to the appropriate RX DMA State Head Descriptor Pointer.

**Figure 9-15. RX Queue Head Descriptor**





Copyright © 2011–2016, Texas Instruments Incorporated

### 9.3.3 MDIO Software Interface

#### 9.3.3.1 Initializing the MDIO Module

The following steps are performed by the application software or device driver to initialize the MDIO device:

1. Configure the PREAMBLE and CLKDIV bits in the MDIO Control register (CONTROL).
2. Enable the MDIO module by setting the ENABLE bit in CONTROL.
3. The MDIO PHY alive status register (ALIVE) can be read in polling fashion until a PHY connected to the system responded, and the MDIO PHY link status register (LINK) can determine whether this PHY already has a link.
4. Setup the appropriate PHY addresses in the MDIO user PHY select register (USERPHYSEL*n*), and set the LINKINTENB bit to enable a link change event interrupt if desirable.
5. If an interrupt on general MDIO register access is desired, set the corresponding bit in the MDIO user command complete interrupt mask set register (USERINTMASKSET) to use the MDIO user access register (USERACCESS*n*). Since only one PHY is used in this device, the application software can use one USERACCESS*n* to trigger a completion interrupt; the other USERACCESS*n* is not setup.

#### 9.3.3.2 Writing Data To a PHY Register

The MDIO module includes a user access register (USERACCESS*n*) to directly access a specified PHY device.To write a PHY register, perform the following:

1. Check to ensure that the GO bit in the MDIO user access register (USERACCESS*n*) is cleared.
2. Write to the GO, WRITE, REGADR, PHYADR, and DATA bits in USERACCESS*n* corresponding to the PHY and PHY register you want to write.
3. The write operation to the PHY is scheduled and completed by the MDIO module. Completion of the write operation can be determined by polling the GO bit in USERACCESS *n* for a 0.
4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (USERINTRAW) corresponding to USERACCESS *n* used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (USERINTMASKSET), then the bit is also set in the MDIO user command complete interrupt register (USERINTMASKED) and an interrupt is triggered on the CPU.

#### 9.3.3.3 Reading Data From a PHY Register

The MDIO module includes a user access register (USERACCESS *n*) to directly access a specified PHY device. To read a PHY register, perform the following:

1. Check to ensure that the GO bit in the MDIO user access register (USERACCESS*n*) is cleared.
2. Write to the GO, REGADR, and PHYADR bits in USERACCESS*n* corresponding to the PHY and PHY register you want to read.
3. The read data value is available in the DATA bits in USERACCESS*n* after the module completes the read operation on the serial bus. Completion of the read operation can be determined by polling the GO and ACK bits in USERACCESS*n*. After the GO bit has cleared, the ACK bit is set on a successful read.
4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (USERINTRAW) corresponding to USERACCESS*n* used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (USERINTMASKSET), then the bit is also set in the MDIO user command complete interrupt register (USERINTMASKED) and an interrupt is triggered on the CPU.


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.3.4 Initialization and Configuration of CPSW

To configure the 3PSW Ethernet Subsystem for operation the host must perform the following:

1. Select the Interface (G/MII, RGMII, RMII) Mode
2. Configure pads (PIN muxing), as per the interface selected.
3. Enable the 3PSW Ethernet Subsystem Clocks
4. Configure the PRCM registers CM_ETHERNET_CLKSTCTRL, CM_ALWON_ETHERNET_0_CLKCTRL to enable power and clocks to 3PSW Ethernet Subsystem.
5. Apply Soft Reset to 3PSW Subsystem, CPSW_3G, CPGMAC_SL1/2, and CPDMA
6. Initialize the HDPs (Header Description Pointer) and CPs (Completion Pointer) to NULL
7. Configure the Interrupts
8. Configure the CPSW_3G CONTROL register
9. Configure the Statistics Port Enable register
10. Configure the ALE
11. Configure the MDIO
12. Configure the CPDMA receive DMA controller
13. Configure the CPDMA transmit DMA controller
14. Configure the CPPI TX and RX Descriptors
15. Configure CPGMAC_SL1 and CPGMAC_SL2, as per the desired mode of operations.
16. Start up RX and TX DMA (Write to HDP of RX and TX)
17. Wait for the completion of Transfer (HDP cleared to 0)


Copyright © 2011–2016, Texas Instruments Incorporated

## 9.4 Registers

### 9.4.1 CPSW_3G Module

This section describes the memory-mapped registers for the CPSW_3G module. The base address of the CPSW_3G registers is 4A10 0000h. Table 9-26 lists the memory-mapped registers for the CPSW_3G module.

**Table 9-26. CPSW_3G Module Registers**

| Offset | Acronym | Register Name |
|---|---|---|
| 000h | CPSW_ID_VER | CPSW ID Version Register |
| 004h | CPSW_CONTROL | CPSW Switch Control Register |
| 008h | CPSW_SOFT_RESET | CPSW Soft Reset Register |
| 00Ch | CPSW_STAT_PORT_EN | CPSW Statistics Port Enable Register |
| 010h | CPSW_PTYPE | CPSW Transmit Priority Type Register |
| 014h | CPSW_SOFT_IDLE | CPSW Software Idle Register |
| 018h | CPSW_THRU_RATE | CPSW Throughput Rate Register |
| 01Ch | CPSW_GAP_THRESH | CPSW CPGMAC_SL Short Gap Threshold Register |
| 020h | CPSW_TX_START_WDS | CPSW Transmit Start Words Register |
| 024h | CPSW_FLOW_CONTROL | CPSW Flow Control Register |
| 028h | P0_MAX_BLKS | CPSW Port 0 Maximum FIFO Blocks Register |
| 02Ch | P0_BLK_CNT | CPSW Port 0 FIFO Block Usage Count Register (read only) |
| 030h | P0_TX_IN_CTL | CPSW Port 0 Transmit FIFO Control Register |
| 034h | P0_PORT_VLAN | CPSW Port 0 VLAN Register |
| 038h | P0_TX_PRI_MAP | CPSW Port 0 TX Header Priority to Switch Priority Mapping Register |
| 03Ch | CPDMA_TX_PRI_MAP | CPSW CPDMA TX (Port 0 RX) Packet Priority to Header Priority Mapping Register |
| 040h | CPDMA_RX_CH_MAP | CPSW CPDMA RX (Port 0 TX) Switch Priority to DMA channel Mapping Register |
| 050h | P1_MAX_BLKS | CPSW Port 1 Maximum FIFO Blocks Register |
| 054h | P1_BLK_CNT | CPSW Port 1 FIFO Block Usage Count Register (read only) |
| 058h | P1_TX_IN_CTL | CPSW Port 1 Transmit FIFO Control Register |
| 05Ch | P1_PORT_VLAN | CPSW Port 1 VLAN Register |
| 060h | P1_TX_PRI_MAP | CPSW Port 1 TX Header Priority to Switch Priority Mapping Register |
| 064h | P1_TS_CTL | CPSW_3G Port 1 Time Sync Control Register |
| 068h | P1_TS_SEQ_LTYPE | CPSW_3G Port 1 Time Sync LTYPE Register (and SEQ_ID_OFFSET) |
| 06Ch | P1_TS_VLAN | CPSW_3G Port 1 Time Sync VLAN2 and VLAN2 Register |
| 070h | SL1_SA_LO | CPSW CPGMAC_SL1 Source Address Low Register |
| 074h | SL1_SA_HI | CPSW CPGMAC_SL1 Source Address High Register |
| 078h | P1_SEND_PERCENT | CPSW Port 1 Transmit Queue Send Percentages Register |
| 090h | P2_MAX_BLKS | CPSW Port 2 Maximum FIFO Blocks Register |
| 094h | P2_BLK_CNT | CPSW Port 2 FIFO Block Usage Count Register (read only) |
| 098h | P2_TX_IN_CTL | CPSW Port 2 Transmit FIFO Control Register |
| 09Ch | P2_PORT_VLAN | CPSW Port 2 VLAN Register |
| 0A0h | P2_TX_PRI_MAP | CPSW Port 2 TX Header Priority to Switch Priority Mapping Register |
| 0A4h | P2_TS_CTL | CPSW_3GF Port 2 Time Sync Control Register |
| 0A8h | P2_TS_SEQ_LTYPE | CPSW_3GF Port 2 Time Sync LTYPE Register (and SEQ_ID_OFFSET) |
| 0ACh | P2_TS_VLAN | CPSW_3GF Port 2 Time Sync VLAN2 and VLAN2 Register |
| 0B0h | SL2_SA_LO | CPSW CPGMAC_SL2 Source Address Low Register |
| 0B4h | SL2_SA_HI | CPSW CPGMAC_SL2 Source Address High Register |
| 0B8h | P2_SEND_PERCENT | CPSW Port 2 Transmit Queue Send Percentages |
| 100h | TX_IDVER | CPDMA_REGS TX Identification and Version Register |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-26. CPSW_3G Module Registers (continued)**

| Offset | Acronym | Register Name |
|---|---|---|
| 104h | TX_CONTROL | CPDMA_REGS TX Control Register |
| 108h | TX_TEARDOWN | CPDMA_REGS TX Teardown Register |
| 110h | RX_IDVER | CPDMA_REGS RX Identification and Version Register |
| 114h | RX_CONTROL | CPDMA_REGS RX Control Register |
| 118h | RX_TEARDOWN | CPDMA_REGS RX Teardown Register |
| 11Ch | SOFT_RESET | CPDMA_REGS Soft Reset Register |
| 120h | DMACONTROL | CPDMA_REGS CPDMA Control Register |
| 124h | DMASTATUS | CPDMA_REGS CPDMA Status Register |
| 128h | RX_BUFFER_OFFSET | CPDMA_REGS Receive Buffer Offset Register |
| 12Ch | EMCONTROL | CPDMA_REGS Emulation Control Register |
| 130h | TX_PRI0_RATE | CPDMA_REGS Transmit (ingress) Priority 0 Rate Register |
| 134h | TX_PRI1_RATE | CPDMA_REGS Transmit (ingress) Priority 1 Rate Register |
| 138h | TX_PRI2_RATE | CPDMA_REGS Transmit (ingress) Priority 2 Rate Register |
| 13Ch | TX_PRI3_RATE | CPDMA_REGS Transmit (ingress) Priority 3 Rate Register |
| 140h | TX_PRI4_RATE | CPDMA_REGS Transmit (ingress) Priority 4 Rate Register |
| 144h | TX_PRI5_RATE | CPDMA_REGS Transmit (ingress) Priority 5 Rate Register |
| 148h | TX_PRI6_RATE | CPDMA_REGS Transmit (ingress) Priority 6 Rate Register |
| 14Ch | TX_PRI7_RATE | CPDMA_REGS Transmit (ingress) Priority 7 Rate Register |
| 180h | TX_INTSTAT_RAW | CPDMA_INT Transmit Interrupt Status Register (raw value) |
| 184h | TX_INTSTAT_MASKED | CPDMA_INT Transmit Interrupt Status Register (masked value) |
| 188h | TX_INTMASK_SET | CPDMA_INT Transmit Interrupt Mask Set Register |
| 18Ch | TX_INTMASK_CLEAR | CPDMA_INT Transmit Interrupt Mask Clear Register |
| 190h | CPDMA_IN_VECTOR | CPDMA_INT Input Vector (read only) |
| 194h | CPDMA_EOI_VECTOR | CPDMA_INT End Of Interrupt Vector |
| 1A0h | RX_INTSTAT_RAW | CPDMA_INT Receive Interrupt Status Register (raw value) |
| 1A4h | RX_INTSTAT_MASKED | CPDMA_INT Receive Interrupt Status Register (masked value) |
| 1A8h | RX_INTMASK_SET | CPDMA_INT Receive Interrupt Mask Set Register |
| 1ACh | RX_INTMASK_CLEAR | CPDMA_INT Receive Interrupt Mask Clear Register |
| 1B0h | DMA_INTSTAT_RAW | CPDMA_INT DMA Interrupt Status Register (raw value) |
| 1B4h | DMA_INTSTAT_MASKED | CPDMA_INT DMA Interrupt Status Register (masked value) |
| 1B8h | DMA_INTMASK_SET | CPDMA_INT DMA Interrupt Mask Set Register |
| 1BCh | DMA_INTMASK_CLEAR | CPDMA_INT DMA Interrupt Mask Clear Register |
| 1C0h | RX0_PENDTHRESH | CPDMA_INT Receive Channel 0 Threshold Pending Register |
| 1C4h | RX1_PENDTHRESH | CPDMA_INT Receive Channel 1 Threshold Pending Register |
| 1C8h | RX2_PENDTHRESH | CPDMA_INT Receive Channel 2 Threshold Pending Register |
| 1CCh | RX3_PENDTHRESH | CPDMA_INT Receive Channel 3 Threshold Pending Register |
| 1D0h | RX4_PENDTHRESH | CPDMA_INT Receive Channel 4 Threshold Pending Register |
| 1D4h | RX5_PENDTHRESH | CPDMA_INT Receive Channel 5 Threshold Pending Register |
| 1D8h | RX6_PENDTHRESH | CPDMA_INT Receive Channel 6 Threshold Pending Register |
| 1DCh | RX7_PENDTHRESH | CPDMA_INT Receive Channel 7 Threshold Pending Register |
| 1E0h | RX0_FREEBUFFER | CPDMA_INT Receive Channel 0 Free Buffer Register |
| 1E4h | RX1_FREEBUFFER | CPDMA_INT Receive Channel 1 Free Buffer Register |
| 1E8h | RX2_FREEBUFFER | CPDMA_INT Receive Channel 2 Free Buffer Register |
| 1ECh | RX3_FREEBUFFER | CPDMA_INT Receive Channel 3 Free Buffer Register |
| 1F0h | RX4_FREEBUFFER | CPDMA_INT Receive Channel 4 Free Buffer Register |
| 1F4h | RX5_FREEBUFFER | CPDMA_INT Receive Channel 5 Free Buffer Register |
| 1F8h | RX6_FREEBUFFER | CPDMA_INT Receive Channel 6 Free Buffer Register |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-26. CPSW_3G Module Registers (continued)**

| Offset | Acronym | Register Name |
|---|---|---|
| 1FCh | RX7_FREEBUFFER | CPDMA_INT Receive Channel 7 Free Buffer Register |
| 200h | TX0_HDP | CPDMA_STATERAM TX Channel 0 Head Descriptor Pointer |
| 204h | TX1_HDP | CPDMA_STATERAM TX Channel 1 Head Descriptor Pointer |
| 208h | TX2_HDP | CPDMA_STATERAM TX Channel 2 Head Descriptor Pointer |
| 20Ch | TX3_HDP | CPDMA_STATERAM TX Channel 3 Head Descriptor Pointer |
| 210h | TX4_HDP | CPDMA_STATERAM TX Channel 4 Head Descriptor Pointer |
| 214h | TX5_HDP | CPDMA_STATERAM TX Channel 5 Head Descriptor Pointer |
| 218h | TX6_HDP | CPDMA_STATERAM TX Channel 6 Head Descriptor Pointer |
| 21Ch | TX7_HDP | CPDMA_STATERAM TX Channel 7 Head Descriptor Pointer |
| 220h | RX0_HDP | CPDMA_STATERAM RX 0 Channel 0 Head Descriptor Pointer |
| 224h | RX1_HDP | CPDMA_STATERAM RX 1 Channel 1 Head Descriptor Pointer |
| 228h | RX2_HDP | CPDMA_STATERAM RX 2 Channel 2 Head Descriptor Pointer |
| 22Ch | RX3_HDP | CPDMA_STATERAM RX 3 Channel 3 Head Descriptor Pointer |
| 230h | RX4_HDP | CPDMA_STATERAM RX 4 Channel 4 Head Descriptor Pointer |
| 234h | RX5_HDP | CPDMA_STATERAM RX 5 Channel 5 Head Descriptor Pointer |
| 238h | RX6_HDP | CPDMA_STATERAM RX 6 Channel 6 Head Descriptor Pointer |
| 23Ch | RX7_HDP | CPDMA_STATERAM RX 7 Channel 7 Head Descriptor Pointer |
| 240h | TX0_CP | CPDMA_STATERAM TX Channel 0 Completion Pointer Register |
| 244h | TX1_CP | CPDMA_STATERAM TX Channel 1 Completion Pointer Register |
| 248h | TX2_CP | CPDMA_STATERAM TX Channel 2 Completion Pointer Register |
| 24Ch | TX3_CP | CPDMA_STATERAM TX Channel 3 Completion Pointer Register |
| 250h | TX4_CP | CPDMA_STATERAM TX Channel 4 Completion Pointer Register |
| 254h | TX5_CP | CPDMA_STATERAM TX Channel 5 Completion Pointer Register |
| 258h | TX6_CP | CPDMA_STATERAM TX Channel 6 Completion Pointer Register |
| 25Ch | TX7_CP | CPDMA_STATERAM TX Channel 7 Completion Pointer Register |
| 260h | RX0_CP | CPDMA_STATERAM RX Channel 0 Completion Pointer Register |
| 264h | RX1_CP | CPDMA_STATERAM RX Channel 1 Completion Pointer Register |
| 268h | RX2_CP | CPDMA_STATERAM RX Channel 2 Completion Pointer Register |
| 26Ch | RX3_CP | CPDMA_STATERAM RX Channel 3 Completion Pointer Register |
| 270h | RX4_CP | CPDMA_STATERAM RX Channel 4 Completion Pointer Register |
| 274h | RX5_CP | CPDMA_STATERAM RX Channel 5 Completion Pointer Register |
| 278h | RX6_CP | CPDMA_STATERAM RX Channel 6 Completion Pointer Register |
| 27Ch | RX7_CP | CPDMA_STATERAM RX Channel 7 Completion Pointer Register |
| 400h | RXGOODFRAMES | CPSW_STATS Total number of good frames received |
| 404h | RXBROADCASTFRAMES | CPSW_STATS Total number of good broadcast frames received |
| 408h | RXMULTICASTFRAMES | CPSW_STATS Total number of good multicast frames received |
| 40Ch | RXPAUSEFRAMES | CPSW_STATS Pause R xFrames |
| 410h | RXCRCERRORS | CPSW_STATS Total number of CRC errors frames received |
| 414h | RXALIGNCODEERRORS | CPSW_STATS Total number of alignment/code errors received |
| 418h | RXOVERSIZEDFRAMES | CPSW_STATS Total number of oversized frames received |
| 41Ch | RXJABBERFRAMES | CPSW_STATS Total number of jabber frames received |
| 420h | RXUNDERSIZEDFRAMES | CPSW_STATS Total number of undersized frames received |
| 424h | RXFRAGMENTS | CPSW_STATS RX Fragments received |
| 430h | RXOCTETS | CPSW_STATS Total number of received bytes in good frames |
| 434h | TXGOODFRAMES | CPSW_STATS Good TX Frames |
| 438h | TXBROADCASTFRAMES | CPSW_STATS Broadcast TX Frames |
| 43Ch | TXMULTICASTFRAMES | CPSW_STATS Multicast TX Frames |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-26. CPSW_3G Module Registers (continued)**

| Offset | Acronym | Register Name |
|---|---|---|
| 440h | TXPAUSEFRAMES | CPSW_STATS Pause TX Frames |
| 444h | TXDEFERREDFRAMES | CPSW_STATS Deferred Frames |
| 448h | TXCOLLISIONFRAMES | CPSW_STATS Collisions |
| 44Ch | TXSINGLECOLLFRAMES | CPSW_STATS Single Collision TX Frames |
| 450h | TXMULTCOLLFRAMES | CPSW_STATS Multiple Collision TX Frames |
| 454h | TXEXCESSIVECOLLISIONS | CPSW_STATS Excessive Collisions |
| 458h | TXLATECOLLISIONS | CPSW_STATS Late Collisions |
| 45Ch | TXUNDERRUN | CPSW_STATS Transmit Underrun Error |
| 460h | TXCARRIERSENSEERRORS | CPSW_STATS Carrier Sense Errors |
| 464h | TXOCTETS | CPSW_STATS TX Octets |
| 468h | 64OCTETFRAMES | CPSW_STATS 64 Octet Frames |
| 46Ch | 65T127OCTETFRAMES | CPSW_STATS 65-127 Octet Frames |
| 470h | 128T255OCTETFRAMES | CPSW_STATS 128-255 Octet Frames |
| 474h | 256T511OCTETFRAMES | CPSW_STATS 256-511 Octet Frames |
| 478h | 512T1023OCTETFRAMES | CPSW_STATS 512-1023 Octet Frames |
| 47Ch | 1024TUPOCTETFRAMES | CPSW_STATS 1023-1518 Octet Frames |
| 480h | NETOCTETS | CPSW_STATS Net Octets |
| 484h | RXSOFOVERRUNS | CPSW_STATS Receive FIFO or DMA Start of Frame Overruns |
| 488h | RXMOFOVERRUNS | CPSW_STATS Receive FIFO or DMA Mid of Frame Overruns |
| 48Ch | RXDMAOVERRUNS | CPSW_STATS Receive DMA Start of Frame and Middle of Frame Overruns |
| 500h | CPTS_IDVER | Identification and Version Register |
| 504h | CPTS_CONTROL | Time Sync Control Register |
| 508h | CPTS_RFTCLK_SEL | Reference Clock Select Register |
| 50Ch | CPTS_TS_PUSH | Time Stamp Event Push Register |
| 510h | CPTS_TS_LOAD_VAL | Time Stamp Load Value Register |
| 514h | CPTS_TS_LOAD_EN | Time Stamp Load Enable Register |
| 520h | CPTS_INTSTAT_RAW | Time Sync Interrupt Status Raw Register |
| 524h | CPTS_INTSTAT_MASKED | Time Sync Interrupt Status Masked Register |
| 528h | CPTS_INT_ENABLE | Time Sync Interrupt Enable Register |
| 530h | CPTS_EVENT_POP | Event Interrupt Pop Register |
| 534h | CPTS_EVENT_LOW | Lower 32-bits of the event value |
| 538h | CPTS_EVENT_HIGH | Upper 32-bits of the event value |
| 600h | ALE_IDVER | Address Lookup Engine ID/Version Register |
| 608h | ALE_CONTROL | Address Lookup Engine Control Register |
| 610h | ALE_PRESCALE | Address Lookup Engine Prescale Register |
| 618h | ALE_UNKNOWN_VLAN | Address Lookup Engine Unknown VLAN Register |
| 620h | ALE_TBLCTL | Address Lookup Engine Table Control Register |
| 634h | ALE_TBLW2 | Address Lookup Engine Table Word 2 Register |
| 638h | ALE_TBLW1 | Address Lookup Engine Table Word 1 Register |
| 63Ch | ALE_TBLW0 | Address Lookup Engine Table Word 0 Register |
| 640h | ALE_PORTCTL0 | Address Lookup Engine Port 0 Control Register |
| 644h | ALE_PORTCTL1 | Address Lookup Engine Port 1 Control Register |
| 648h | ALE_PORTCTL2 | Address Lookup Engine Port 2 Control Register |
| 700h | SL1_IDVER | CPGMAC_SL1 ID/Version Register |
| 704h | SL1_MACCONTROL | CPGMAC_SL1 Mac Control Register |
| 708h | SL1_MACSTATUS | CPGMAC_SL1 Mac Status Register |
| 70Ch | SL1_SOFT_RESET | CPGMAC_SL1 Soft Reset Register |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-26. CPSW_3G Module Registers (continued)**

| Offset | Acronym | Register Name |
|---|---|---|
| 710h | SL1_RX_MAXLEN | CPGMAC_SL1 RX Maximum Length Register |
| 714h | SL1_BOFFTEST | CPGMAC_SL1 Backoff Test Register |
| 718h | SL1_RX_PAUSE | CPGMAC_SL1 Receive Pause Timer Register |
| 71Ch | SL1_TX_PAUSE | CPGMAC_SL1 Transmit Pause Timer Register |
| 720h | SL1_EMCONTROL | CPGMAC_SL1 Emulation Control Register |
| 724h | SL1_RX_PRI_MAP | CPGMAC_SL1 RX Packet Priority to Header Priority Mapping Register |
| 740h | SL2_IDVER | CPGMAC_SL2 ID/Version Register |
| 744h | SL2_MACCONTROL | CPGMAC_SL2 Mac Control Register |
| 748h | SL2_MACSTATUS | CPGMAC_SL2 Mac Status Register |
| 74Ch | SL2_SOFT_RESET | CPGMAC_SL2 Soft Reset Register |
| 750h | SL2_RX_MAXLEN | CPGMAC_SL2 RX Maximum Length Register |
| 754h | SL2_BOFFTEST | CPGMAC_SL2 Backoff Test Register |
| 758h | SL2_RX_PAUSE | CPGMAC_SL2 Receive Pause Timer Register |
| 75Ch | SL2_TX_PAUSE | CPGMAC_SL2 Transmit Pause Timer Register |
| 760h | SL2_EMCONTROL | CPGMAC_SL2 Emulation Control Register |
| 764h | SL2_RX_PRI_MAP | CPGMAC_SL2 RX Packet Priority to Header Priority Mapping Register |

**Table 9-27. Register Bit Definitions**

| Term | Definition |
|---|---|
| w | A writeable bit or field |
| wc | A Write-to-clear bit. Writing a bit of this type with a one will clear the bit to zero. Writing a zero to a bit of this type has no effect. |
| ws | A Write-to-set bit. Writing a bit of this type with a one will set the bit to one. Writing a zero to a bit of this type has no effect. |
| wi | A Write-to-increment field. Writing a value to this field increments the field value by the value written. Reads are permitted and do not alter the field value. |
| wd | A Write-to-decrement field. Writing a value to this field decrements the field by the value written. Reads are permitted and do not alter the field value. |
| r | A readable bit or field |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.1 CPSW ID Version Register (CPSW_ID_VER) (0x000)

**Figure 9-16. CPSW ID Version Register (CPSW_ID_VER)**

| 31 | 16 |
|---|---|
| CPSW_3G_IDENT | |
| R-19h | |

| 15 | 11 | 10 | 8 | 7 | 0 |
|---|---|---|---|---|---|
| CPSW_3G_RTL_VER | | CPSW_3G_MAJ_VER | | CPSW_3G_MINOR_VER | |
| R-0 | | R-1 | | R-Ah | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-28. CPSW ID Version Register (CPSW_ID_VER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | CPSW_3G_IDENT | 0-FFFFh | CPSW_3G Identification Value. |
| 15-11 | CPSW_3G_RTL_VER | 0-1Fh | CPSW_3G RTL Version Value. |
| 10-8 | CPSW_3G_MAJ_VER | 0-7h | CPSW_3G Major Version Value. |
| 7-0 | CPSW_3G_MINOR_VER | 0-FFh | CPSW_3G Minor Version Value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.2 CPSW Control Register (CPSW_CONTROL) (0x004)

**Figure 9-17. CPSW Control Register (CPSW_CONTROL)**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | P2_PASS_PRI_TAGGED | P1_PASS_PRI_TAGGED | P0_PASS_PRI_TAGGED | RX_VLAN_ENCAP | VLAN_AWARE | FIFO_LOOPBACK |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-29. CPSW Control Register (CPSW_CONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Read as zero. |
| 5 | P2_PASS_PRI_TAGGED | | Port 2 Pass Priority Tagged |
| | | 0 | Priority tagged packets have the zero VID replaced with the input port P2_PORT_VLAN[11:0] |
| | | 1 | Priority tagged packets are processed unchanged. |
| 4 | P1_PASS_PRI_TAGGED | | Port 1 Pass Priority Tagged |
| | | 0 | Priority tagged packets have the zero VID replaced with the inputport P1_PORT_VLAN[11:0] |
| | | 1 | Priority tagged packets are processed unchanged. |
| 3 | P0_PASS_PRI_TAGGED | | Port 0 Pass Priority Tagged |
| | | 0 | Priority tagged packets have the zero VID replaced with the inputport P0_PORT_VLAN[11:0] |
| | | 1 | Priority tagged packets are processed unchanged. |
| 2 | RX_VLAN_ENCAP | | Port 0 VLAN Encapsulation (egress) |
| | | 0 | Port 0 receive packets (from CPSW_3G) are not VLAN encapsulated. |
| | | 1 | Port 0 receive packets (from CPSW_3G) are VLAN encapsulated. |
| 1 | VLAN_AWARE | | VLAN Aware Mode |
| | | 0 | CPSW_3G is in the VLAN unaware mode. |
| | | 1 | CPSW_3G is in the VLAN aware mode. |
| 0 | FIFO_LOOPBACK | | FIFO Loopback Mode |
| | | 0 | Loopback is disabled |
| | | 1 | FIFO Loopback mode enabled. Each packet received is turned around and sent out on the same port's transmit path. Port 2 receive is fixed on channel zero. The RXSOFOVERRUN statistic will increment for every packet sent in FIFO loopback mode. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.3 CPSW Software Reset Register (CPSW_SOFT_RESET) (0x008)

**Figure 9-18. CPSW Software Reset Register (CPSW_SOFT_RESET)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | SOFT_RESET |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-30. CPSW Software Reset Register (CPSW_SOFT_RESET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Read as zero. |
| 0 | SOFT_RESET | | Software reset |
| | | 0 | If a zero is read then reset has occurred. |
| | | 1 | Writing a one to this bit causes the CPSW_3G logic to be reset. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. |

### 9.4.1.4 CPSW Statistics Port Enable Register (CPSW_STAT_PORT_EN) (0x00C)

**Figure 9-19. CPSW Statistics Port Enable Register (CPSW_STAT_PORT_EN)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | P2_STAT_EN | P1_STAT_EN | P0_STAT_EN |
| R-0 | | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-31. CPSW Statistics Port Enable Register (CPSW_STAT_PORT_EN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Read as zero. |
| 2 | P2_STAT_EN | | Port 2 (EMAC1 and Port 2 FIFO) Statistics Enable |
| | | 0 | Port 2 statistics are not enabled. |
| | | 1 | Port 2 statistics are enabled. |
| 1 | P1_STAT_EN | | Port 1 (EMAC0 and Port 1 FIFO) Statistics Enable |
| | | 0 | Port 1 statistics are not enabled. |
| | | 1 | Port 1 statistics are enabled. |
| 0 | P0_STAT_EN | | Port 0 Statistics Enable |
| | | 0 | Port 0 statistics are not enabled |
| | | 1 | Port 0 statistics are enabled. FIFO overruns (SOFOVERRUNS) are the only port 0 statistics that are enabled to be kept. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.5 CPSW Transmit Priority Type Register (CPSW_PTYPE) (0x010)

**Figure 9-20. CPSW Transmit Priority Type Register (CPSW_PTYPE)**

| 31 | 24 |
|---|---|
| Reserved | |
| R-0 | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | P2_PRI3_SHAPE_EN | P2_PRI2_SHAPE_EN | P2_PRI1_SHAPE_EN | P1_PRI3_SHAPE_EN | P1_PRI2_SHAPE_EN | P1_PRI1_SHAPE_EN |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|
| Reserved | | P2_PTYPE_ESC | P1_PTYPE_ESC | P0_PTYPE_ESC |
| R-0 | | R/W-0 | R/W-0 | R/W-0 |

| 7 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | ESC_PRI_LD_VAL | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

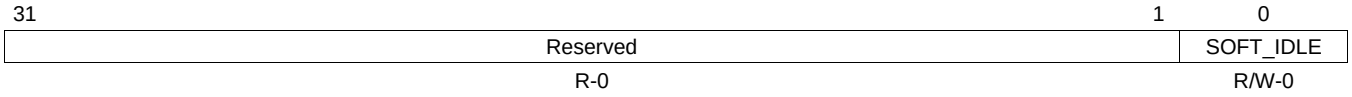
**Table 9-32. CPSW Transmit Priority Type Register (CPSW_PTYPE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-22 | Reserved | 0 | Read as zero. |
| 21 | P2_PRI3_SHAPE_EN | 0-1 | Port 2 Queue Priority 3 Transmit Shape Enable<br>If there is only one shaping queue then it must be priority 3. |
| 20 | P2_PRI2_SHAPE_EN | 0-1 | Port 2 Queue Priority 2 Transmit Shape Enable<br>If there are two shaping queues then they must be priorities 3 and 2. |
| 19 | P2_PRI1_SHAPE_EN | 0-1 | Port 2 Queue Priority 1 Transmit Shape Enable<br>If there are three shaping queues all three bits should be set. |
| 18 | P1_PRI3_SHAPE_EN | 0-1 | Port 1 Queue Priority 3 Transmit Shape Enable<br>If there is only one shaping queue then it must be priority 3. |
| 17 | P1_PRI2_SHAPE_EN | 0-1 | Port 1 Queue Priority 2 Transmit Shape Enable<br>If there are two shaping queues then they must be priorities 3 and 2. |
| 16 | P1_PRI1_SHAPE_EN | 0-1 | Port 1 Queue Priority 1 Transmit Shape Enable<br>If there are three shaping queues all three bits should be set. |
| 15-11 | Reserved | 0 | Read as zero. |
| 10 | P2_PTYPE_ESC | | Port 2 Priority Type Escalate |
| | | 0 | Port 2 priority type fixed |
| | | 1 | Port 2 priority type escalate. Escalate should not be used with queue shaping. |
| 9 | P1_PTYPE_ESC | | Port 1 Priority Type Escalate |
| | | 0 | Port 1 priority type fixed |
| | | 1 | Port 1 priority type escalate. Escalate should not be used with queue shaping. |
| 8 | P0_PTYPE_ESC | | Port 0 Priority Type Escalate |
| | | 0 | Port 0 priority type fixed |
| | | 1 | Port 0 priority type escalate. Escalate should not be used with queue shaping. |
| 7-5 | Reserved | 0 | Read as zero. |
| 4-0 | ESC_PRI_LD_VAL | 0-1Fh | Escalate Priority Load Value When a port is in escalate priority, this is the number of higher priority packets sent before the next lower priority is allowed to send a packet. Escalate priority allows lower priority packets to be sent at a fixed rate relative to the next higher priority. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.6 CPSW Software Idle Register (CPSW_SOFT_IDLE) (0x014)

**Figure 9-21. CPSW Software Idle Register (CPSW_SOFT_IDLE)**



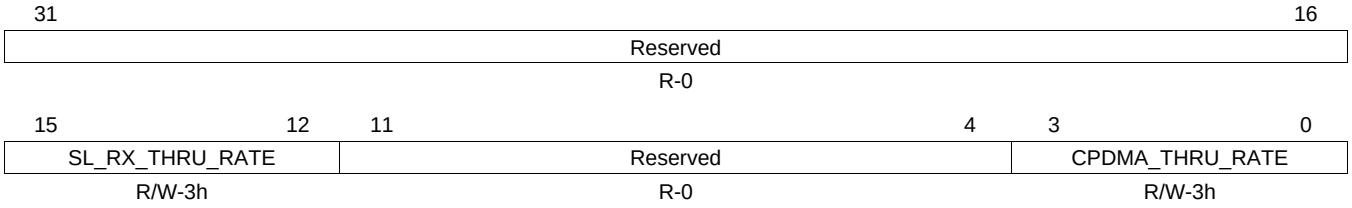

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-33. CPSW Software Idle Register (CPSW_SOFT_IDLE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Read as zero. |
| 0 | SOFT_IDLE | | Software Idle<br>Setting this bit causes the switch fabric to stop forwarding packets at the next start of packet. |

### 9.4.1.7 CPSW Throughput Rate Register (CPSW_THRU_RATE) (0x018)

**Figure 9-22. CPSW Throughput Rate Register (CPSW_THRU_RATE)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-34. CPSW Throughput Rate Register (CPSW_THRU_RATE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero. |
| 15-12 | SL_RX_THRU_RATE | 0-Fh | CPGMAC_SL Switch FIFO receive through rate.<br>This register value is the maximum throughput of the ethernet ports to the crossbar SCR. The default is one 8-byte word for every 3 CLK periods maximum. |
| 11-4 | Reserved | 0 | Read as zero. |
| 3-0 | CPDMA_THRU_RATE | 0-Fh | CPDMA Switch FIFO receive through rate.<br>This register value is the maximum throughput of the CPDMA host port to the crossbar SCR. The default is one 8-byte word for every 3 CLK periods maximum. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.8 CPSW CPGMAC_SL Short Gap Threshold Register (CPSW_GAP_THRESH) (0x01C)

**Figure 9-23. CPSW CPGMAC_SL Short Gap Threshold Register (CPSW_GAP_THRESH)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | GAP_THRESH | |
| R-0 | | R/W-Bh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-35. CPSW CPGMAC_SL Short Gap Threshold Register (CPSW_GAP_THRESH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Read as zero. |
| 4-0 | GAP_THRESH | 0-1Fh | CPGMAC_SL Short Gap Threshold<br>This is the CPGMAC_SL associated FIFO transmit block usage value for triggering TX_SHORT_GAP |

### 9.4.1.9 CPSW Transmit Start Words Register (CPSW_TX_START_WDS) (0x020)

**Figure 9-24. CPSW Transmit Start Words Register (CPSW_TX_START_WDS)**

| 31 | 11 | 10 | 0 |
|---|---|---|---|
| Reserved | | TX_START_WDS | |
| R-0 | | R/W-20h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-36. CPSW Transmit Start Words Register (CPSW_TX_START_WDS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Reserved | 0 | Read as zero. |
| 10-0 | TX_START_WDS | 0-7FFh | FIFO Packet Transmit (egress) Start Words.<br>This value is the number of required packet words in the transmit FIFO before the packet egress will begin. This value is non-zero to preclude underrun. 20h (decimal 32) is the recommended value, it should not be increased unnecessarily to prevent adding to the switch latency. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.10 CPSW Flow Control Register (CPSW_FLOW_CONTROL) (0x024)

**Figure 9-25. CPSW Flow Control Register (CPSW_FLOW_CONTROL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | P2_FLOW_EN | P1_FLOW_EN | P0_FLOW_EN |
| R-0 | | R/W-0 | R/W-0 | R/W-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-37. CPSW Flow Control Register (CPSW_FLOW_CONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Read as zero. |
| 2 | P2_FLOW_EN | 0-1 | Port 2 Receive flow control enable. |
| 1 | P1_FLOW_EN | 0-1 | Port 1 Receive flow control enable. |
| 0 | P0_FLOW_EN | 0-1 | Port 0 Receive flow control enable. |

### 9.4.1.11 CPSW Port 0 Maximum FIFO Blocks Register (P0_MAX_BLKS) (0x028)

**Figure 9-26. CPSW Port 0 Maximum FIFO Blocks Register (P0_MAX_BLKS)**

| 31 | 9 | 8 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | P0_TX_MAX_BLKS | | P0_RX_MAX_BLKS | |
| R-0 | | R/W-10h | | R/W-4h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

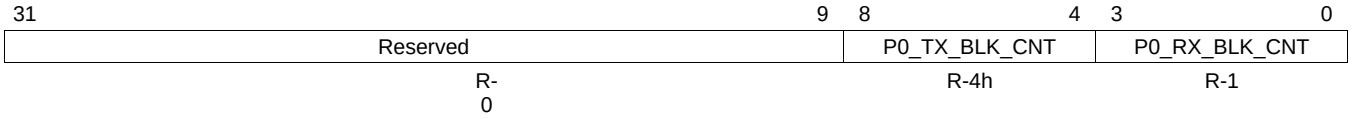
**Table 9-38. CPSW Port 0 Maximum FIFO Blocks Register (P0_MAX_BLKS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Read as zero. |
| 8-4 | P0_TX_MAX_BLKS | 0-1Fh | Transmit FIFO Maximum Blocks<br>This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 10h is the recommended value. Port 0 should remain in flow control mode. Eh is the minimum value. |
| 3-0 | P0_RX_MAX_BLKS | 0-Fh | Receive FIFO Maximum Blocks<br>This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. 4h is the recommended value, 3h is the minimum value, and 6h is the maximum value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.12 CPSW Port 0 FIFO Block Usage Count (Read Only) Register (P0_BLK_CNT) (0x02C)

**Figure 9-27. CPSW Port 0 FIFO Block Usage Count (Read Only) Register (P0_BLK_CNT)**



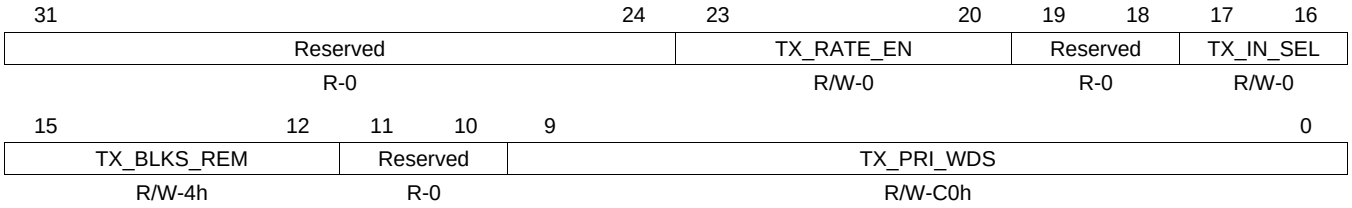

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-39. CPSW Port 0 FIFO Block Usage Count (Read Only) Register (P0_BLK_CNT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Read as zero. |
| 8-4 | P0_TX_BLK_CNT | 0-1Fh | Port 0 Transmit Block Count Usage<br>This value is the number of blocks allocated to the FIFO logical transmit queues. |
| 3-0 | P0_RX_BLK_CNT | 0-Fh | Port 0 Receive Block Count Usage<br>This value is the number of blocks allocated to the FIFO logical receive queues. |

### 9.4.1.13 CPSW Port 0 Transmit FIFO Control Register (P0_TX_IN_CTL) (0x030)

**Figure 9-28. CPSW Port 0 Transmit FIFO Control Register (P0_TX_IN_CTL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-40. CPSW Port 0 Transmit FIFO Control Register (P0_TX_IN_CTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Reserved | 0 | Read as zero. |
| 23-20 | TX_RATE_EN | 0-Fh | Transmit FIFO Input Rate Enable |
| 19-18 | Reserved | 0 | Read as zero. |
| 17-16 | TX_IN_SEL | | Transmit FIFO Input Queue Type Select |
| | | 0 | Normal priority mode |
| | | 1h | Dual MAC mode |
| | | 2h | Rate Limit mode |
| | | 3h | Reserved |
| | | | **Note:** Dual MAC mode is not compatible with escalation or shaping because dual mac mode forces round robin priority on FIFO egress. |
| 15-12 | TX_BLKS_REM | 0-Fh | Transmit FIFO Input Blocks to subtract in dual mac mode. |
| 11-10 | Reserved | 0 | Read as zero. |
| 9-0 | TX_PRI_WDS | 0-3FFh | Transmit FIFO Words in queue. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.14 CPSW Port 0 VLAN Register (P0_PORT_VLAN) (0x034)

**Figure 9-29. CPSW Port 0 VLAN Register (P0_PORT_VLAN)**

| 31 16 | 15 13 | 12 | 11 0 |
|---|---|---|---|
| Reserved | PORT_PRI | PORT_CFI | PORT_VID |
| R-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-41. CPSW Port 0 VLAN Register (P0_PORT_VLAN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero. |
| 15-13 | PORT_PRI | 0-7h | Port VLAN Priority (7 is highest priority). |
| 12 | PORT_CFI | 0-1 | Port CFI bit. |
| 11-0 | PORT_VID | 0-FFFh | Port VLAN ID. |

### 9.4.1.15 CPSW Port 0 TX Header Priority to Switch Priority Mapping Register (P0_TX_PRI_MAP) (0x038)

**Figure 9-30. CPSW Port 0 TX Header Priority to Switch Priority Mapping Register (P0_TX_PRI_MAP)**

| 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|
| Reserved | PRI7 | Reserved | PRI6 | Reserved | PRI5 | Reserved | PRI4 |
| R-0 | R/W-3h | R-0 | R/W-3h | R-0 | R/W-2h | R-0 | R/W-2h |

| 15 14 | 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
|---|---|---|---|---|---|---|---|
| Reserved | PRI3 | Reserved | PRI2 | Reserved | PRI1 | Reserved | PRI0 |
| R-0 | R/W-1 | R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-42. CPSW Port 0 TX Header Priority to Switch Priority Mapping Register (P0_TX_PRI_MAP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Read as zero. |
| 29-28 | PRI7 | 0-3h | Priority 7. A packet header priority of 7 is given this switch queue priority. |
| 27-26 | Reserved | 0 | Read as zero. |
| 25-24 | PRI6 | 0-3h | Priority 6. A packet header priority of 6 is given this switch queue priority. |
| 23-22 | Reserved | 0 | Read as zero. |
| 21-20 | PRI5 | 0-3h | Priority 5. A packet header priority of 5 is given this switch queue priority. |
| 19-18 | Reserved | 0 | Read as zero. |
| 17-16 | PRI4 | 0-3h | Priority 4. A packet header priority of 4 is given this switch queue priority. |
| 15-14 | Reserved | 0 | Read as zero. |
| 13-12 | PRI3 | 0-3h | Priority 3. A packet header priority of 03 is given this switch queue priority. |
| 11-10 | Reserved | 0 | Read as zero. |
| 9-8 | PRI2 | 0-3h | Priority 2. A packet header priority of 2 is given this switch queue priority. |
| 7-6 | Reserved | 0 | Read as zero. |
| 5-4 | PRI1 | 0-3h | Priority 1. A packet header priority of 1 is given this switch queue priority. |
| 3-2 | Reserved | 0 | Read as zero. |
| 1-0 | PRI0 | 0-3h | Priority 0. A packet header priority of 0 is given this switch queue priority. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.16 CPSW CPDMA TX (Port 0 RX) Packet Priority to Header Priority Mapping Register (CPDMA_TX_PRI_MAP) (0x03C)

**Figure 9-31. CPSW CPDMA TX (Port 0 RX) Packet Priority to Header Priority Mapping Register (CPDMA_TX_PRI_MAP)**

| 31 | 30 28 | 27 | 26 24 | 23 | 22 20 | 19 | 18 16 |
|---|---|---|---|---|---|---|---|
| Reserved | PRI7 | Reserved | PRI6 | Reserved | PRI5 | Reserved | PRI4 |
| R-0 | R/W-7h | R-0 | R/W-6h | R-0 | R/W-5h | R-0 | R/W-4h |
| 15 | 14 12 | 11 | 10 8 | 7 | 6 4 | 3 | 2 0 |
| Reserved | PRI3 | Reserved | PRI2 | Reserved | PRI1 | Reserved | PRI0 |
| R-0 | R/W-3h | R-0 | R/W-2h | R-0 | R/W-1h | R-0 | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-43. CPSW CPDMA TX (Port 0 RX) Packet Priority to Header Priority Mapping Register (CPDMA_TX_PRI_MAP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Read as zero. |
| 30-28 | PRI7 | 0-3h | Priority 7. A packet priority of 7 is mapped (changed) to this header packet priority. |
| 27 | Reserved | 0 | Read as zero. |
| 26-24 | PRI6 | 0-3h | Priority 6. A packet priority of 6 is mapped (changed) to this header packet priority |
| 23 | Reserved | 0 | Read as zero. |
| 22-20 | PRI5 | 0-3h | Priority 5. A packet priority of 5 is mapped (changed) to this header packet priority. |
| 19 | Reserved | 0 | Read as zero. |
| 18-16 | PRI4 | 0-3h | Priority 4. A packet priority of 4 is mapped (changed) to this header packet priority. |
| 15 | Reserved | 0 | Read as zero. |
| 14-12 | PRI3 | 0-3h | Priority 3. A packet priority of 3 is mapped (changed) to this header packet priority. |
| 11 | Reserved | 0 | Read as zero. |
| 10-8 | PRI2 | 0-3h | Priority 2. A packet priority of 2 is mapped (changed) to this header packet priority. |
| 7 | Reserved | 0 | Read as zero. |
| 6-4 | PRI1 | 0-3h | Priority 1. A packet priority of 1 is mapped (changed) to this header packet priority. |
| 3 | Reserved | 0 | Read as zero. |
| 2-0 | PRI0 | 0-3h | Priority 0. A packet priority of 0 is mapped (changed) to this header packet priority. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.17 CPSW CPDMA RX (Port 0 TX) Switch Priority to DMA Channel Mapping Register (CPDMA_RX_CH_MAP) (0x040)

**Figure 9-32. CPSW CPDMA RX (Port 0 TX) Switch Priority to DMA Channel Mapping Register (CPDMA_RX_CH_MAP)**

| 31 | 30 28 | 27 | 26 24 | 23 | 22 20 | 19 | 18 16 |
|---|---|---|---|---|---|---|---|
| Reserved | P2_PRI3 | Reserved | P2_PRI2 | Reserved | P2_PRI1 | Reserved | P2_PRI0 |
| R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 |

| 15 | 14 12 | 11 | 10 8 | 7 | 6 4 | 3 | 2 0 |
|---|---|---|---|---|---|---|---|
| Reserved | P1_PRI3 | Reserved | P1_PRI2 | Reserved | P1_PRI1 | Reserved | P1_PRI0 |
| R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-44. CPSW CPDMA RX (Port 0 TX) Switch Priority to DMA Channel Mapping Register (CPDMA_RX_CH_MAP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Read as zero. |
| 30-28 | P2_PRI3 | 0-7h | Port 2 Priority 3 packets go to this CPDMA RX Channel |
| 27 | Reserved | 0 | Read as zero. |
| 26-24 | P2_PRI2 | 0-7h | Port 2 Priority 2 packets go to this CPDMA RX Channel |
| 23 | Reserved | 0 | Read as zero. |
| 22-20 | P2_PRI1 | 0-7h | Port 2 Priority 1 packets go to this CPDMA RX Channel |
| 19 | Reserved | 0 | Read as zero. |
| 18-16 | P2_PRI0 | 0-7h | Port 2 Priority 0 packets go to this CPDMA RX Channel |
| 15 | Reserved | 0 | Read as zero. |
| 14-12 | P1_PRI3 | 0-7h | Port 1 Priority 3 packets go to this CPDMA RX Channel |
| 11 | Reserved | 0 | Read as zero. |
| 10-8 | P1_PRI2 | 0-7h | Port 1 Priority 2 packets go to this CPDMA RX Channel |
| 7 | Reserved | 0 | Read as zero. |
| 6-4 | P1_PRI1 | 0-7h | Port 1 Priority 1 packets go to this CPDMA RX Channel |
| 3 | Reserved | 0 | Read as zero. |
| 2-0 | P1_PRI0 | 0-7h | Port 1 Priority 0 packets go to this CPDMA RX Channel |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.18 CPSW Port 1 Maximum FIFO Blocks Register (P1_MAX_BLKS) (0x050)

**Figure 9-33. CPSW Port 1 Maximum FIFO Blocks Register (P1_MAX_BLKS)**

| 31 9 | 8 4 | 3 0 |
|---|---|---|
| Reserved | P1_TX_MAX_BLKS | P1_RX_MAX_BLKS |
| R-0 | R/W-11h | R/W-3h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-45. CPSW Port 1 Maximum FIFO Blocks Register (P1_MAX_BLKS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Read as zero. |
| 8-4 | P1_TX_MAX_BLKS | 0-1Fh | Transmit FIFO Maximum Blocks.<br>This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 11h is the recommended value of P1_TX_MAX_BLKS, unless the port is in fullduplex flow control mode. In flow control mode, the P1_RX_MAX_BLKS will need to increase in order to accept the required run out in fullduplex mode. This value will need to decrease by the amount of increase in P1_RX_MAX_BLKS. Eh is the minimum value TX_MAX_BLKS. |
| 3-0 | P1_RX_MAX_BLKS | 0-Fh | Receive FIFO Maximum Blocks.<br>This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. This value must be greater than or equal to 3h. It should be increased in fullduplex flow control mode to 5h or 6h, depending on the required run out space. The P1_TX_MAX_BLKS value must be decreased by the amount of increase in P1_RX_MAX_BLKS. 3h is the minimum value and 6h is the maximum value. |

### 9.4.1.19 CPSW Port 1 FIFO Block Usage Count (Read Only) Register (P1_BLK_CNT) (0x054)

**Figure 9-34. CPSW Port 1 FIFO Block Usage Count (Read Only) Register (P1_BLK_CNT)**

| 31 9 | 8 4 | 3 0 |
|---|---|---|
| Reserved | P1_TX_BLK_CNT | P1_RX_BLK_CNT |
| R-0 | R-4h | R-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-46. CPSW Port 1 FIFO Block Usage Count (Read Only) Register (P1_BLK_CNT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Read as zero. |
| 8-4 | P1_TX_BLK_CNT | 0-1Fh | Port 1 Transmit Block Count Usage.<br>This value is the number of blocks allocated to the FIFO logical transmit queues. |
| 3-0 | P1_RX_BLK_CNT | 0-Fh | Port 1 Receive Block Count Usage.<br>This value is the number of blocks allocated to the FIFO logical receive queues. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.20 CPSW Port 1 Transmit FIFO Control Register (P1_TX_IN_CTL) (0x058)

**Figure 9-35. CPSW Port 1 Transmit FIFO Control Register (P1_TX_IN_CTL)**

| 31–28 | 27–24 | 23–20 | 19–18 | 17–16 |
|---|---|---|---|---|
| Reserved | HOST_BLKS_REM | TX_RATE_EN | Reserved | TX_IN_SEL |
| R-0 | R/W-8h | R/W-0 | R-0 | R/W-0 |

| 15–12 | 11–10 | 9–0 |
|---|---|---|
| TX_BLKS_REM | Reserved | TX_PRI_WDS |
| R/W-4h | R-0 | R/W-C0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-47. CPSW Port 1 Transmit FIFO Control Register (P1_TX_IN_CTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | Reserved | 0 | Read as zero. |
| 27-24 | HOST_BLKS_REM | 0-Fh | Transmit FIFO Blocks that must be free before a non rate-limited CPDMA channel can begin sending a packet to the FIFO. |
| 23-20 | TX_RATE_EN | 0-Fh | Transmit FIFO Input Rate Enable |
| 19-18 | Reserved | 0 | Read as zero. |
| 17-16 | TX_IN_SEL | | Transmit FIFO Input Queue Type Select |
| | | 0 | Normal priority mode |
| | | 1h | Reserved |
| | | 2h | Rate Limit mode |
| | | 3h | Reserved |
| 15-12 | TX_BLKS_REM | 0-Fh | Transmit FIFO Input Blocks to subtract in dual mac mode and blocks to subtract on non rate-limited traffic in rate-limit mode. |
| 11-10 | Reserved | 0 | Read as zero. |
| 9-0 | TX_PRI_WDS | 0-3FFh | Transmit FIFO Words in queue. |

### 9.4.1.21 CPSW Port 1 VLAN Register (P1_PORT_VLAN) (0x05C)

**Figure 9-36. CPSW Port 1 VLAN Register (P1_PORT_VLAN)**

| 31–16 | 15–13 | 12 | 11–0 |
|---|---|---|---|
| Reserved | PORT_PRI | PORT_CFI | PORT_VID |
| R-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-48. CPSW Port 1 VLAN Register (P1_PORT_VLAN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero. |
| 15-13 | PORT_PRI | 0-7h | Port VLAN Priority (7 is highest priority). |
| 12 | PORT_CFI | 0-1 | Port CFI bit. |
| 11-0 | PORT_VID | 0-FFFh | Port VLAN ID. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.22 CPSW Port 1 TX Header Priority to Switch Pri Mapping Register (P1_TX_PRI_MAP) (0x060)

**Figure 9-37. CPSW Port 1 TX Header Priority to Switch Pri Mapping Register (P1_TX_PRI_MAP)**

| 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|
| Reserved | PRI7 | Reserved | PRI6 | Reserved | PRI5 | Reserved | PRI4 |
| R-0 | R/W-3h | R-0 | R/W-3h | R-0 | R/W-2h | R-0 | R/W-2h |

| 15 14 | 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
|---|---|---|---|---|---|---|---|
| Reserved | PRI3 | Reserved | PRI2 | Reserved | PRI1 | Reserved | PRI0 |
| R-0 | R/W-1 | R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-49. CPSW Port 1 TX Header Priority to Switch Pri Mapping Register (P1_TX_PRI_MAP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Read as zero. |
| 29-28 | PRI7 | 0-3h | Priority 7. A packet header priority of 0x7 is given this switch queue prorityi. |
| 27-26 | Reserved | 0 | Read as zero. |
| 25-24 | PRI6 | 0-3h | Priority 6. A packet header priority of 0x6 is given this switch queue priority. |
| 23-22 | Reserved | 0 | Read as zero. |
| 21-20 | PRI5 | 0-3h | Priority 5. A packet header priority of 0x5 is given this switch queue priority. |
| 19-18 | Reserved | 0 | Read as zero. |
| 17-16 | PRI4 | 0-3h | Priority 4. A packet header priority of 0x4 is given this switch queue priority. |
| 15-14 | Reserved | 0 | Read as zero. |
| 13-12 | PRI3 | 0-3h | Priority 3. A packet header priority of 0x3 is given this switch queue priority. |
| 11-10 | Reserved | 0 | Read as zero. |
| 9-8 | PRI2 | 0-3h | Priority 2. A packet header priority of 0x2 is given this switch queue priority. |
| 7-6 | Reserved | 0 | Read as zero. |
| 5-4 | PRI1 | 0-3h | Priority 1. A packet header priority of 0x1 is given this switch queue priority. |
| 3-2 | Reserved | 0 | Read as zero. |
| 1-0 | PRI0 | 0-3h | Priority 0. A packet header priority of 0x0 is given this switch queue priority. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.23 CPSW_3G Port 1 Time Sync Control Register (P1_TS_CTL) (0x064)

**Figure 9-38. CPSW_3G Port 1 Time Sync Control Register (P1_TS_CTL)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1_TX_MSG_TYPE_EN | | | | | | | | | | | | | | | |
| R/W-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | P1_TS_TX_ VLAN_LTYPE2_EN | P1_TS_TX_ VLAN_LTYPE1_EN | P1_TS_TX_EN | Rsvd | P1_TS_RX_ VLAN_LTYPE2_EN | P1_TS_RX_ VLAN_LTYPE1_EN | P1_TS_RX_EN |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-50. CPSW_3G Port 1 Time Sync Control Register (P1_TS_CTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | P1_TX_MSG_TYPE_EN | 0-FFFFh | Port 1 Time Sync Message Type Enable.<br>Each bit in this field enables the corresponding message type in receive and transmit time sync messages (Bit 0 enables message type 0 etc.). |
| 15-7 | Reserved | 0 | Read as zero. |
| 6 | P1_TS_TX_VLAN_LTYPE2_EN | 0-1 | Port 1 Time Sync Transmit VLAN LTYPE 2 enable. |
| 5 | P1_TS_TX_VLAN_LTYPE1_EN | 0-1 | Port 1 Time Sync Transmit VLAN LTYPE 1 enable. |
| 4 | P1_TS_TX_EN | | Port 1 Time Sync Transmit Enable. |
| | | 0 | Port 1 Transmit Time Sync disabled. |
| | | 1 | Port 1 Transmit Time Sync enabled. |
| 3 | Reserved | 0 | Read as zero. |
| 2 | P1_TS_RX_VLAN_LTYPE2_EN | 0-1 | Port 1 Time Sync Receive VLAN LTYPE 2 enable. |
| 1 | P1_TS_RX_VLAN_LTYPE1_EN | 0-1 | Port 1 Time Sync Receive VLAN LTYPE 1 enable. |
| 0 | P1_TS_RX_EN | | Port 1 Time Sync Receive Enable. |
| | | 0 | Port 1 Receive Time Sync disabled. |
| | | 1 | Port 1 Receive Time Sync enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.24 CPSW_3G Port 1 Time Sync Sequence ID and LTYPE Register (P1_TS_SEQ_LTYPE) (0x068)

**Figure 9-39. CPSW_3G Port 1 Time Sync Sequence ID and LTYPE Register (P1_TS_SEQ_LTYPE)**

| 31 22 | 21 16 | 15 0 |
|---|---|---|
| Reserved | P1_TS_SEQ_ID_OFFSET | P1_TS_LTYPE |
| R-0 | R/W-1Eh | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-51. CPSW_3G Port 1 Time Sync Sequence ID and LTYPE Register (P1_TS_SEQ_LTYPE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-22 | Reserved | 0 | Read as zero. |
| 21-16 | P1_TS_SEQ_ID_OFFSET | 0-3Fh | Port 1 Time Sync Sequence ID Offset.<br>This is the number of octets that the sequence ID is offset in the tx and rx time sync message header. The minimum value is 6h. |
| 15-0 | P1_TS_LTYPE | 0-FFFFh | Port 1 Time Sync LTYPE.<br>This is the LTYPE value to match for tx and rx time sync messages. |

### 9.4.1.25 CPSW_3G Port 1 Time Sync VLAN2 and VLAN2 Register (P1_TS_VLAN) (0x06C)

**Figure 9-40. CPSW_3G Port 1 Time Sync VLAN2 and VLAN2 Register (P1_TS_VLAN)**

| 31 16 | 15 0 |
|---|---|
| P1_TS_VLAN_LTYPE2 | P1_TS_VLAN_LTYPE1 |
| R/W-8100h | R/W-8100h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-52. CPSW_3G Port 1 Time Sync VLAN2 and VLAN2 Register (P1_TS_VLAN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | P1_TS_VLAN_LTYPE2 | 0-FFFFh | Port 1 Time Sync VLAN LTYPE2. This VLAN LTYPE value is used for port 1 tx and rx time sync decode. |
| 15-0 | P1_TS_VLAN_LTYPE1 | 0-FFFFh | Port 1 Time Sync VLAN LTYPE1. This VLAN LTYPE value is used for port 1 tx and rx time sync decode. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.26 CPSW CPGMAC_SL1 Source Address Low Register (SL1_SA_LO) (0x070)

**Figure 9-41. CPSW CPGMAC_SL1 Source Address Low Register (SL1_SA_LO)**

| 31 16 | 15 8 | 7 0 |
|---|---|---|
| Reserved | MACSRCADDR(7:0) | MACSRCADDR(15:8) |
| R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-53. CPSW CPGMAC_SL1 Source Address Low Register (SL1_SA_LO) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero. |
| 15-8 | MACSRCADDR(7:0) | 0-FFh | Source Address Lower 8 bits (byte 0) |
| 7-0 | MACSRCADDR(15:8) | 0-FFh | Source Address bits 15:8 (byte 1) |

### 9.4.1.27 CPSW CPGMAC_SL1 Source Address High Register (SL1_SA_HI) (0x074)

**Figure 9-42. CPSW CPGMAC_SL1 Source Address High Register (SL1_SA_HI)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| MACSRCADDR(23:16) | MACSRCADDR(31:24) | MACSRCADDR(39:32) | MACSRCADDR(47:40) |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

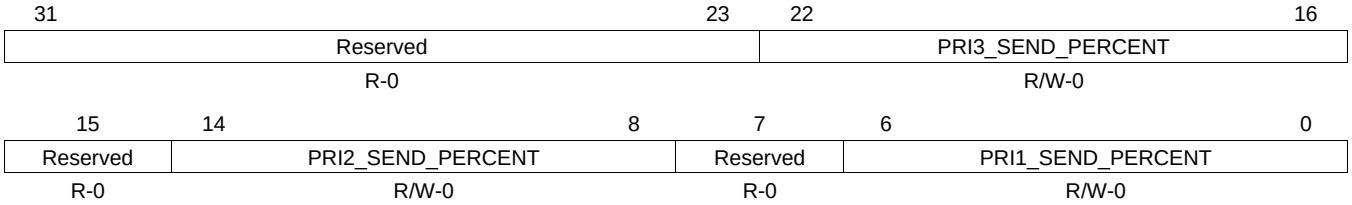
**Table 9-54. CPSW CPGMAC_SL1 Source Address High Register (SL1_SA_HI) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | MACSRCADDR(23:16) | 0-FFh | Source Address bits 23:16 (byte 2). |
| 23-16 | MACSRCADDR(31:24) | 0-FFh | Source Address bits 31:24 (byte 3). |
| 15-8 | MACSRCADDR(39:32) | 0-FFh | Source Address bits 39:32 (byte 4). |
| 7-0 | MACSRCADDR(47:40) | 0-FFh | Source Address bits 47:40 (byte 5). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.28 CPSW Port 1 Transmit Queue Send Percentages Register (P1_SEND_PERCENT) (0x078)

**Figure 9-43. CPSW Port 1 Transmit Queue Send Percentages Register (P1_SEND_PERCENT)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-55. CPSW Port 1 Transmit Queue Send Percentages Register (P1_SEND_PERCENT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-23 | Reserved | 0 | Read as zero. |
| 22-16 | PRI3_SEND_PERCENT | 0-7Fh | Priority 3 Transmit Percentage<br>This percentage value is sent from FIFO priority 3 (maximum) when the P1_PRI3_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 3 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue, then this value must be between 0 and 64h (decimal 100 (not inclusive)). |
| 15 | Reserved | 0 | Read as zero. |
| 14-8 | PRI2_SEND_PERCENT | 0-7Fh | Priority 2 Transmit Percentage<br>This percentage value is sent from FIFO priority 2 (maximum) when the P1_PRI2_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 2 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue, then this value must be between 0 and 64h (decimal 100 (not inclusive)). |
| 7 | Reserved | 0 | Read as zero. |
| 6-0 | PRI1_SEND_PERCENT | 0-7Fh | Priority 1 Transmit Percentage<br>This percentage value is sent from FIFO priority 1 (maximum) when the P1_PRI1_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 1 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue, then this value must be between 0 and 64h (decimal 100 (not inclusive)). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.29 CPSW Port 2 Maximum FIFO Blocks Register (P2_MAX_BLKS) (0x090)

**Figure 9-44. CPSW Port 2 Maximum FIFO Blocks Register (P2_MAX_BLKS)**

| 31 9 | 8 4 | 3 0 |
|---|---|---|
| Reserved | P2_TX_MAX_BLKS | P2_RX_MAX_BLKS |
| R-0 | R/W-11h | R/W-3h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-56. CPSW Port 2 Maximum FIFO Blocks Register (P2_MAX_BLKS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Read as zero. |
| 8-4 | P2_TX_MAX_BLKS | 0-1Fh | Transmit FIFO Maximum Blocks<br>This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 11h is the recommended value of P2_TX_MAX_BLKS unless the port is in fullduplex flow control mode. In flow control mode, the P2_RX_MAX_BLKS will need to increase in order to accept the required run out in fullduplex mode. This value will need to decrease by the amount of increase in P2_RX_MAX_BLKS. Eh is the minimum value TX_MAX_BLKS. |
| 3-0 | P2_RX_MAX_BLKS | 0-Fh | Receive FIFO Maximum Blocks<br>This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. This value must be greater than or equal to 3h. It should be increased In fullduplex flow control mode to 5h or 6h, depending on the required runout space. The P2_TX_MAX_BLKS value must be decreased by the amount of increase in P2_RX_MAX_BLKS. 3h is the minimum value RX_MAX_BLKS and 6h is the maximum value. |

### 9.4.1.30 CPSW Port 2 FIFO Block Usage Count (Read Only) Register (P2_BLK_CNT) (0x094)

**Figure 9-45. CPSW Port 2 FIFO Block Usage Count (Read Only) Register (P2_BLK_CNT)**

| 31 9 | 8 4 | 3 0 |
|---|---|---|
| Reserved | P2_TX_BLK_CNT | P2_RX_BLK_CNT |
| R-0 | R-4h | R-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-57. CPSW Port 2 FIFO Block Usage Count (Read Only) Register (P2_BLK_CNT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Read as zero. |
| 8-4 | P2_TX_BLK_CNT | 0-1Fh | Port 2 Transmit Block Count Usage. This value is the number of blocks allocated to the FIFO logical transmit queues. |
| 3-0 | P2_RX_BLK_CNT | 0-Fh | Port 2 Receive Block Count Usage. This value is the number of blocks allocated to the FIFO logical receive queues. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.31 CPSW Port 2 Transmit FIFO Control Register (P2_TX_IN_CTL) (0x098)

**Figure 9-46. CPSW Port 2 Transmit FIFO Control Register (P2_TX_IN_CTL)**

| 31–28 | 27–24 | 23–20 | 19–18 | 17–16 |
|---|---|---|---|---|
| Reserved | HOST_BLKS_REM | TX_RATE_EN | Reserved | TX_IN_SEL |
| R-0 | R/W-8h | R/W-0 | R-0 | R/W-0 |

| 15–12 | 11–10 | 9–0 |
|---|---|---|
| TX_BLKS_REM | Reserved | TX_PRI_WDS |
| R/W-4h | R-0 | R/W-C0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-58. CPSW Port 2 Transmit FIFO Control Register (P2_TX_IN_CTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | Reserved | 0 | Read as zero. |
| 27-24 | HOST_BLKS_REM | 0-Fh | Transmit FIFO Blocks that must be free before a non rate-limited CPDMA channel can begin sending a packet to the FIFO. |
| 23-20 | TX_RATE_EN | 0-Fh | Transmit FIFO Input Rate Enable |
| 19-18 | Reserved | 0 | Read as zero. |
| 17-16 | TX_IN_SEL | | Transmit FIFO Input Queue Type Select |
| | | 0 | Normal priority mode |
| | | 1h | Reserved |
| | | 2h | Rate Limit mode |
| | | 3h | Reserved |
| 15-12 | TX_BLKS_REM | 0-Fh | Transmit FIFO Input Blocks to subtract in dual mac mode and blocks to subtract on non rate-limited traffic in rate-limit mode. |
| 11-10 | Reserved | 0 | Read as zero. |
| 9-0 | TX_PRI_WDS | 0-3FFh | Transmit FIFO Words in queue. |

### 9.4.1.32 CPSW Port 2 VLAN Register (P2_PORT_VLAN) (0x09C)

**Figure 9-47. CPSW Port 2 VLAN Register (P2_PORT_VLAN)**

| 31–16 | 15–13 | 12 | 11–0 |
|---|---|---|---|
| Reserved | PORT_PRI | PORT_CFI | PORT_VID |
| R-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-59. CPSW Port 2 VLAN Register (P2_PORT_VLAN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero. |
| 15-13 | PORT_PRI | 0-7h | Port VLAN Priority (7 is highest priority). |
| 12 | PORT_CFI | 0-1 | Port CFI bit. |
| 11-0 | PORT_VID | 0-FFFh | Port VLAN ID. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.33 CPSW Port 2 TX Header Priority to Switch Priority Mapping Register (P2_TX_PRI_MAP) (0x0A0)

**Figure 9-48. CPSW Port 2 TX Header Priority to Switch Priority Mapping Register (P2_TX_PRI_MAP)**

| 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|
| Reserved | PRI7 | Reserved | PRI6 | Reserved | PRI5 | Reserved | PRI4 |
| R-0 | R/W-3h | R-0 | R/W-3h | R-0 | R/W-2h | R-0 | R/W-2h |

| 15 14 | 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
|---|---|---|---|---|---|---|---|
| Reserved | PRI3 | Reserved | PRI2 | Reserved | PRI1 | Reserved | PRI0 |
| R-0 | R/W-1 | R-0 | R/W-0 | R-0 | R/W-0 | R-0 | R/W-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

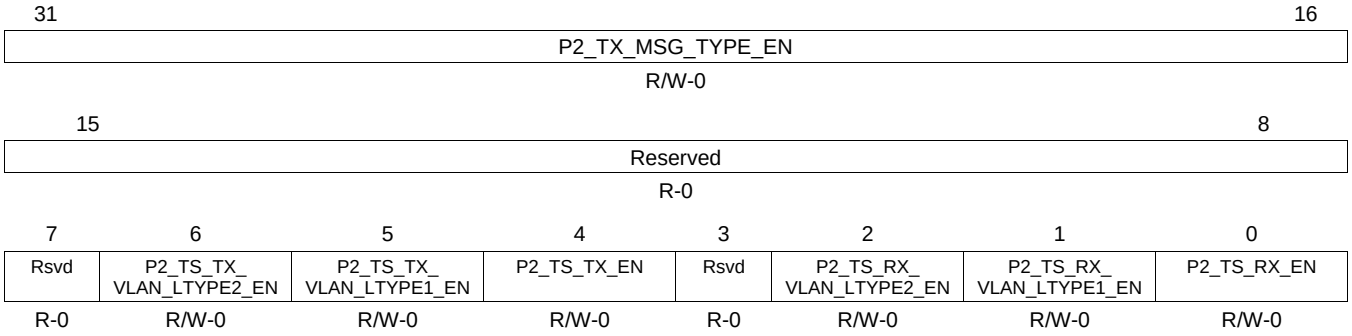
**Table 9-60. CPSW Port 2 TX Header Priority to Switch Priority Mapping Register (P2_TX_PRI_MAP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Read as zero. |
| 29-28 | PRI7 | 0-3h | Priority 7. A packet header priority of 7 is given this switch queue priority. |
| 27-26 | Reserved | 0 | Read as zero. |
| 25-24 | PRI6 | 0-3h | Priority 6. A packet header priority of 6 is given this switch queue priority. |
| 23-22 | Reserved | 0 | Read as zero. |
| 21-20 | PRI5 | 0-3h | Priority 5. A packet header priority of 5 is given this switch queue priority. |
| 19-18 | Reserved | 0 | Read as zero. |
| 17-16 | PRI4 | 0-3h | Priority 4. A packet header priority of 4 is given this switch queue priority. |
| 15-14 | Reserved | 0 | Read as zero. |
| 13-12 | PRI3 | 0-3h | Priority 3. A packet header priority of 3 is given this switch queue priority. |
| 11-10 | Reserved | 0 | Read as zero. |
| 9-8 | PRI2 | 0-3h | Priority 2. A packet header priority of 2 is given this switch queue priority. |
| 7-6 | Reserved | 0 | Read as zero. |
| 5-4 | PRI1 | 0-3h | Priority 1. A packet header priority of 1 is given this switch queue priority. |
| 3-2 | Reserved | 0 | Read as zero. |
| 1-0 | PRI0 | 0-3h | Priority 0. A packet header priority of 0 is given this switch queue priority. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.34 CPSW_3GF Port 2 Time Sync Control Register (P2_TS_CTL) (0x0A4)

**Figure 9-49. CPSW_3GF Port 2 Time Sync Control Register (P2_TS_CTL)**




LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 9-61. CPSW_3GF Port 2 Time Sync Control Register (P2_TS_CTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | P2_TX_MSG_TYPE_EN | 0-FFFFh | Port 2 Time Sync Message Type Enable. Each bit in this field enables the corresponding message type in receive and transmit time sync messages (Bit 0 enables message type 0 etc.). |
| 15-7 | Reserved | 0 | Read as zero. |
| 6 | P2_TS_TX_VLAN_LTYPE2_EN | 0-1 | Port 2 Time Sync Transmit VLAN LTYPE 2 enable. |
| 5 | P2_TS_TX_VLAN_LTYPE1_EN | 0-1 | Port 2Time Sync Transmit VLAN LTYPE 1 enable. |
| 4 | P2_TS_TX_EN | | Port 2 Time Sync Transmit Enable. |
| | | 0 | Port 2 Transmit Time Sync disabled. |
| | | 1 | Port 2 Transmit Time Sync enabled. |
| 3 | Reserved | 0 | Read as zero. |
| 2 | P2_TS_RX_VLAN_LTYPE2_EN | 0-1 | Port 2 Time Sync Receive VLAN LTYPE 2 enable. |
| 1 | P2_TS_RX_VLAN_LTYPE1_EN | 0-1 | Port 2 Time Sync Receive VLAN LTYPE 1 enable. |
| 0 | P2_TS_RX_EN | | Port 2 Time Sync Receive Enable. |
| | | 0 | Port 2 Receive Time Sync disabled. |
| | | 1 | Port 2 Receive Time Sync enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.35 CPSW_3GF Port 2 Time Sync Sequence ID and LTYPE Register (P2_TS_SEQ_LTYPE) (0x0A8)

**Figure 9-50. Port 2 Time Sync Sequence ID and LTYPE Register (P2_TS_SEQ_LTYPE)**

| 31 22 | 21 16 | 15 0 |
|---|---|---|
| Reserved | P2_TS_SEQ_ID_OFFSET | P2_TS_LTYPE |
| R-0 | R/W-1Eh | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-62. Port 2 Time Sync Sequence ID and LTYPE Register (P2_TS_SEQ_LTYPE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-22 | Reserved | 0 | Read as zero. |
| 21-16 | P2_TS_SEQ_ID_OFFSET | 0-3Fh | Port 2 Time Sync Sequence ID Offset. This is the number of octets that the sequence ID is offset in the tx and rx time sync message header. The minimum value is 6. |
| 15-0 | P2_TS_LTYPE | 0-FFFFh | Port 2 Time Sync LTYPE. This is the LTYPE value to match for tx and rx time sync messages. |

### 9.4.1.36 CPSW_3GF Port 2 Time Sync VLAN2 and VLAN2 Register (P2_TS_VLAN) (0x0AC)

**Figure 9-51. CPSW_3GF Port 2 Time Sync VLAN2 and VLAN2 Register (P2_TS_VLAN)**

| 31 16 | 15 0 |
|---|---|
| P2_TS_VLAN_LTYPE2 | P2_TS_VLAN_LTYPE1 |
| R/W-8100h | R/W-8100h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-63. CPSW_3GF Port 2 Time Sync VLAN2 and VLAN2 Register (P2_TS_VLAN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | P2_TS_VLAN_LTYPE2 | 0-FFFFh | Port 2 Time Sync VLAN LTYPE2. This VLAN LTYPE value is used for port 1 tx and rx time sync decode. |
| 15-0 | P2_TS_VLAN_LTYPE1 | 0-FFFFh | Port 2 Time Sync VLAN LTYPE1. This VLAN LTYPE value is used for port 1 tx and rx time sync decode. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.37 CPSW CPGMAC_SL2 Source Address Low Register (SL2_SA_LO) (0x0B0)

**Figure 9-52. CPSW CPGMAC_SL2 Source Address Low Register (SL2_SA_LO)**

| 31 16 | 15 8 | 7 0 |
|---|---|---|
| Reserved | MACSRCADDR(7:0) | MACSRCADDR(15:8) |
| R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-64. CPSW CPGMAC_SL2 Source Address Low Register (SL2_SA_LO) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Read as zero. |
| 15-8 | MACSRCADDR(7:0) | 0-FFh | Source Address Lower 8 bits (byte 0). |
| 7-0 | MACSRCADDR(15:8) | 0-FFh | Source Address bits 15:8 (byte 1). |

### 9.4.1.38 CPSW CPGMAC_SL2 Source Address High Register (SL2_SA_HI) (0x0B4)

**Figure 9-53. CPGMAC_SL2 Source Address High Register (SL2_SA_HI)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| MACSRCADDR(23:16) | MACSRCADDR(31:24) | MACSRCADDR(39:32) | MACSRCADDR(47:40) |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-65. CPGMAC_SL2 Source Address High Register (SL2_SA_HI) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | MACSRCADDR(23:16) | 0-FFh | Source Address bits 23:16 (byte 2) |
| 23-16 | MACSRCADDR(31:23) | 0-FFh | Source Address bits 31:23 (byte 3) |
| 15-8 | MACSRCADDR(39:32) | 0-FFh | Source Address bits 39:32 (byte 4) |
| 7-0 | MACSRCADDR(47:40) | 0-FFh | Source Address bits 47:40 (byte 5) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.39 Port 2 Transmit Queue Shaping Percentages Register (P2_SEND_PERCENT) (0x0B8)

**Figure 9-54. Port 2 Transmit Queue Send Percentages Register (P2_SEND_PERCENT)**

| 31 | 23 | 22 | 16 |
|---|---|---|---|
| Reserved | | PRI3_SEND_PERCENT | |
| R-0 | | R/W-0 | |

| 15 | 14 | 8 | 7 | 6 | 0 |
|---|---|---|---|---|---|
| Rsvd | PRI2_SEND_PERCENT | | Rsvd | PRI1_SEND_PERCENT | |
| R-0 | R/W-0 | | R-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-66. Port 2 Transmit Queue Send Percentages Register (P2_SEND_PERCENT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-23 | Reserved | 0 | Read as zero. |
| 22-16 | PRI3_SEND_PERCENT | 0-7Fh | Priority 3 Transmit Percentage.<br>This percentage value is sent from FIFO priority 3 (maximum) when the P2_PRI3_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 3 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue, then this value must be between 0 and 64h (decimal 100 (not inclusive)). |
| 15 | Reserved | 0 | Read as zero. |
| 14-8 | PRI2_SEND_PERCENT | 0-7Fh | Priority 2 Transmit Percentage.<br>This percentage value is sent from FIFO priority 2 (maximum) when the P2_PRI2_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 2 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue t en this value must be between 0 and 64h (decimal 100 (not inclusive)). |
| 7 | Reserved | 0 | Read as zero. |
| 6-0 | PRI1_SEND_PERCENT | 0-7Fh | Priority 1 Transmit Percentage.<br>This percentage value is sent from FIFO priority 1 (maximum) when the P2_PRI1_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 1 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between 0 and 64h (decimal 100 (not inclusive)). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.40 CPDMA_REGS TX Identification and Version Register (TX_IDVER) (0x100)

**Figure 9-55. CPDMA_REGS TX Identification and Version Register (TX_IDVER)**

| 31 16 | 15 8 | 7 0 |
|---|---|---|
| TX_IDENT | TX_MAJOR_VER | TX_MINOR_VER |
| R-18h | R-1 | R-8h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-67. CPDMA_REGS TX Identification and Version Register (TX_IDVER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | TX_IDENT | 0-FFFFh | TX Identification Value |
| 15-8 | TX_MAJOR_VER | 0-FFh | TX Major Version Value. The value read is the major version number |
| 7-0 | TX_MINOR_VER | 0-FFh | TX Minor Version Value. The value read is the minor version number |

### 9.4.1.41 TCPDMA_REGS TX Control Register (TX_CONTROL) (0x104)

**Figure 9-56. CPDMA_REGS TX Control Register (TX_CONTROL)**

| 31 1 | 0 |
|---|---|
| Reserved | TX_EN |
| R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-68. CPDMA_REGS TX Control Register (TX_CONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Read as zero |
| 0 | TX_EN | | TX Enable |
| | | 0 | Disabled |
| | | 1 | Enabled |

### 9.4.1.42 TCPDMA_REGS TX Teardown Register (TX_TEARDOWN) (0x108)

**Figure 9-57. CPDMA_REGS TX Teardown Register (TX_TEARDOWN)**

| 31 | 30 3 | 2 0 |
|---|---|---|
| TX_TDN_RDY | Reserved | TX_TDN_CH |
| R-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-69. CPDMA_REGS TX Teardown Register (TX_TEARDOWN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | TX_TDN_RDY | | TX Teardown Ready. Read as zero, but is always assumed to be one (unused). |
| 30-3 | Reserved | 0 | Read as zero |
| 2-0 | TX_TDN_RDY | | TX Teardown Channel. Transmit channel teardown is commanded by writing the encoded value of the transmit channel to be torn down. The teardown register is read as zero. |
| | | 0-7h | Teardown transmit channel 0 to teardown transmit channel 7 |

### 9.4.1.43 CPDMA_REGS RX Identification and Version Register (RX_IDVER) (0x110)


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 9-58. CPDMA_REGS RX Identification and Version Register (RX_IDVER)**

| 31 16 | 15 8 | 7 0 |
|---|---|---|
| RX_IDENT | RX_MAJOR_VER | RX_MINOR_VER |
| R-Ch | R-1 | R-7h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-70. CPDMA_REGS RX Identification and Version Register (RX_IDVER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | RX_IDENT | 0-FFFFh | RX Identification Value |
| 15-8 | RX_MAJOR_VER | 0-FFh | RX Major Version Value |
| 7-0 | RX_MINOR_VER | 0-FFh | RX Minor Version Value |

### 9.4.1.44 CPDMA_REGS RX Control Register (RX_CONTROL) (0x114)

**Figure 9-59. CPDMA_REGS RX Control Register (RX_CONTROL)**

| 31 1 | 0 |
|---|---|
| Reserved | RX_EN |
| R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-71. CPDMA_REGS RX Control Register (RX_CONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Read as zero |
| 0 | RX_EN | | RX DMA Enable |
| | | 0 | Disabled |
| | | 1 | Enabled |

### 9.4.1.45 CPDMA_REGS RX Teardown Register (RX_TEARDOWN) (0x118)

**Figure 9-60. CPDMA_REGS RX Teardown Register (RX_TEARDOWN)**

| 31 | 30 3 | 2 0 |
|---|---|---|
| RX_TDN_RDY | Reserved | RX_TDN_CH |
| R-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-72. CPDMA_REGS RX Teardown Register (RX_TEARDOWN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | RX_TDN_RDY | | Teardown Ready. Read as zero, but is always assumed to be one (unused). |
| 30-3 | Reserved | 0 | Read as zero |
| 2-0 | RX_TDN_CH | | RX Teardown Channel. Transmit channel teardown is commanded by writing the encoded value of the transmit channel to be torn down. The teardown register is read as zero. |
| | | 0-7h | Teardown transmit channel 0 to teardown transmit channel 7 |

### 9.4.1.46 CPDMA_REGS Software Reset Register (SOFT_RESET) (0x11C)


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 9-61. CPDMA_REGS Software Reset Register (SOFT_RESET)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | SOFT_RESET |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-73. CPDMA_REGS Software Reset Register (SOFT_RESET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Read as zero |
| 0 | SOFT_RESET | | Software reset. Writing a one to this bit causes the CPDMA logic to be reset. Software reset occurs when the RX and TX DMA Controllers are in an idle state to avoid locking up the VBUSP bus. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred. |

### 9.4.1.47 CPDMA_REGS Control Register (DMACONTROL) (0x120)

**Figure 9-62. CPDMA_REGS Control Register (DMACONTROL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 |
|---|---|
| TX_RLIM | |
| R/W-0 | |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | RX_CEF | CMD_IDLE | RX_OFFLEN_BLOCK | RX_OWNERSHIP | TX_PTYPE |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-74. CPDMA_REGS Control Register (DMACONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero |
| 15-8 | TX_RLIM | | Transmit Rate Limit Channel Bus |
| | | 0 | No rate-limited channels |
| | | 80h | Channel 7 is rate-limited |
| | | C0h | Channels 7 to 6 are rate-limited |
| | | E0h | Channels 7 to 5 are rate-limited |
| | | F0h | Channels 7 to 4 are rate-limited |
| | | F8h | Channels 7 to 3 are rate-limited |
| | | FCh | Channels 7 to 2 are rate-limited |
| | | FEh | Channels 7 to 1 are rate-limited |
| | | FFh | Channels 7 to 0 are rate-limited |
| | | | All others invalid – this bus must be set MSB towards LSB. TX_PTYPE must be set, if any TX_RLIM bit is set for fixed priority. |
| 7-5 | Reserved | 0 | Read as zero |
| 4 | RX_CEF | | RX Copy Error Frames Enable. Enables DMA overrun frames to be transferred to memory (up to the point of overrun). The overrun error bit will be set in the frame EOP buffer descriptor. Overrun frame data will be filtered when RX_CEF is not set. Frames coming from the receive FIFO with other error bits set are not effected by this bit. |
| | | 0 | Frames containing overrun errors are filtered. |
| | | 1 | Frames containing overrun errors are transferred to memory. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-74. CPDMA_REGS Control Register (DMACONTROL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 3 | CMD_IDLE | | Command Idle |
| | | 0 | Idle not commanded |
| | | 1 | Idle Commanded (read idle in DMAStatus) |
| 2 | RX_OFFLEN_BLOCK | | Receive Offset/Length word write block |
| | | 0 | Do not block the DMA writes to the receive buffer descriptor offset/buffer length word. The offset/buffer length word is written as specified in CPPI 3.0. |
| | | 1 | Block all CPDMA DMA controller writes to the receive buffer descriptor offset/buffer length words during CPPI packet processing. when this bit is set, the CPDMA will never write the third word to any receive buffer descriptor. |
| 1 | RX_OWNERSHIP | | Receive Ownership Write Bit Value |
| | | 0 | The CPDMA writes the receive ownership bit to zero at the end of packet processing as specified in CPPI 3.0. |
| | | 1 | The CPDMA writes the receive ownership bit to one at the end of packet processing. Users who do not use the ownership mechanism can use this mode to preclude the necessity of software having to set this bit each time the buffer descriptor is used. |
| 0 | TX_PTYPE | | Transmit Queue Priority Type |
| | | 0 | The queue uses a round robin scheme to select the next channel for transmission. |
| | | 1 | The queue uses a fixed (channel 7 highest priority) priority scheme to select the next channel for transmission. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.48 CPDMA_REGS Status Register (DMASTATUS) (0x124)

**Figure 9-63. CPDMA_REGS Status Register (DMASTATUS)**

| 31 | 30 | 24 | 23 | 20 | 19 | 18 | 16 |
|---|---|---|---|---|---|---|---|
| IDLE | Reserved | | TX_HOST_ERR_CODE | | Rsvd | TX_ERR_CH | |
| R-0 | R-0 | | R-0 | | R-0 | R-0 | |

| 15 | 12 | 11 | 10 | 8 | 7 | 0 |
|---|---|---|---|---|---|---|
| RX_HOST_ERR_CODE | | Rsvd | RX_ERR_CH | | Reserved | |
| R-0 | | R-0 | R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

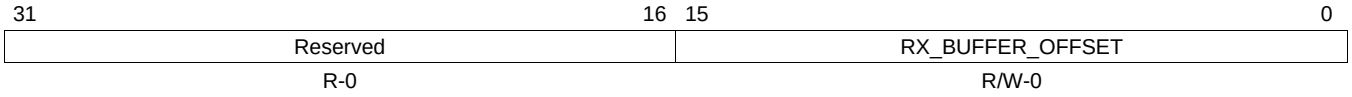
**Table 9-75. CPDMA_REGS Status Register (DMASTATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | IDLE | | Idle Status Bit – Indicates when set that the CPDMA is not transferring a packet on transmit or receive. |
| 30-24 | Reserved | 0 | Read as zero |
| 23-20 | TX_HOST_ERR_CODE | | TX Host Error Code – This field is set to indicate CPDMA detected TX DMA related host errors. The host should read this field after a HOST_ERR_INT to determine the error. Host error Interrupts require hardware reset in order to recover. A zero packet length is an error, but it is not detected. |
| | | 0 | No error |
| | | 1h | SOP error |
| | | 2h | Ownership bit not set in SOP buffer |
| | | 3h | Zero Next Buffer Descriptor Pointer Without EOP |
| | | 4h | Zero Buffer Pointer |
| | | 5h | Zero Buffer Length |
| | | 6h | Packet Length Error (sum of buffers < packet length) |
| | | 7h-Fh | Reserved |
| 19 | Reserved | 0 | Read as zero |
| 18-16 | TX_ERR_CH | | TX Host Error Channel – This field indicates which TX channel (if applicable) the host error occurred on. This field is cleared to zero on a host read. |
| | | 0-7h | The host error occurred on TX channel 0 to TX channel 7 |
| 15-12 | RX_HOST_ERR_CODE | | RX Host Error Code – This field is set to indicate CPDMA detected RX DMA related host errors. The host should read this field after a HOST_ERR_INT to determine the error. Host error Interrupts require hardware reset in order to recover. |
| | | 0 | No error |
| | | 1h | Reserved |
| | | 2h | Ownership bit not set in input buffer |
| | | 3h | Reserved |
| | | 4h | Zero Buffer Pointer |
| | | 5h | Zero buffer length on non-SOP descriptor |
| | | 6h | SOP buffer length not greater than offset |
| | | 7h-Fh | Reserved |
| 11 | Reserved | 0 | Read as zero |
| 10-8 | RX_ERR_CH | | RX Host Error Channel – This field indicates which RX channel the host error occurred on. This field is cleared to zero on a host read. |
| | | 0-7h | The host error occurred on RX channel 0 to RX channel 7 |
| 7-0 | Reserved | 0 | Read as zero |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.49 CPDMA_REGS Receive Buffer Offset Register (RX_BUFFER_OFFSET) (0x128)

**Figure 9-64. CPDMA_REGS Receive Buffer Offset Register (RX_BUFFER_OFFSET)**



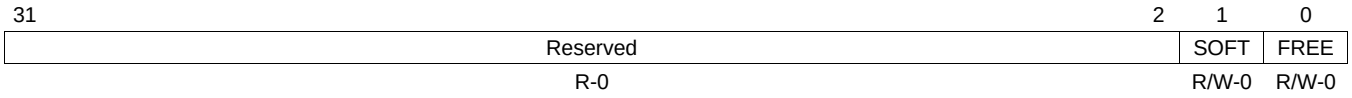

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-76. CPDMA_REGS Receive Buffer Offset Register (RX_BUFFER_OFFSET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero |
| 15-0 | RX_BUFFER_OFFSET | | Receive Buffer Offset Value. The RX_BUFFER_OFFSET will be written by the port into each frame SOP buffer descriptor BUFFER_OFFSET field. The frame data will begin after the RX_BUFFER_OFFSET value of bytes. A value of 0 indicates that there are no unused bytes at the beginning of the data and that valid data begins on the first byte of the buffer. A value of Fh (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. This value is used for all channels. |

### 9.4.1.50 CPDMA_REGS Emulation Control Register (EMCONTROL) (0x12C)

**Figure 9-65. CPDMA_REGS Emulation Control Register (EMCONTROL)**



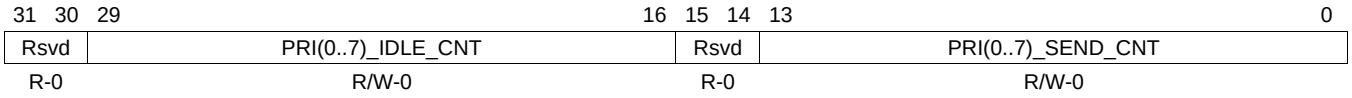

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-77. CPDMA_REGS Emulation Control Register (EMCONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Read as zero |
| 1 | SOFT | 0-1 | Emulation Soft Bit |
| 0 | FREE | 0-1 | Emulation Free Bit |

### 9.4.1.51 CPDMA_REGS Transmit Priority Rate Registers (TX_PRI0_RATE-TX_PRI7_RATE) (0x130 - 0x14C)

**Figure 9-66. CPDMA_REGS Transmit Priority Rate Registers**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-78. CPDMA_REGS Transmit Priority Rate Registers Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Read as zero |
| 29-16 | PRI(0..7)_IDLE_CNT | 0-3FFFh | Priority (7:0) idle count |
| 15-14 | Reserved | 0 | Read as zero |
| 13-0 | PRI(0..7)_SEND_CNT | 0-3FFFh | Priority (7:0) send count |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.52 CPDMA_INT TX Interrupt Status (raw value) Register (TX_INTSTAT_RAW) (0x180)

**Figure 9-67. CPDMA_INT TX Interrupt Status (raw value) Register (TX_INTSTAT_RAW)**

| 31 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX7_PEND | TX6_PEND | TX5_PEND | TX4_PEND | TX3_PEND | TX2_PEND | TX1_PEND | TX0_PEND |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-79. CPDMA_INT TX Interrupt Status (raw value) Register (TX_INTSTAT_RAW) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero |
| 7 | TX7_PEND | 0-1 | TX7_PEND raw interrupt read (before mask) |
| 6 | TX6_PEND | 0-1 | TX6_PEND raw interrupt read (before mask) |
| 5 | TX5_PEND | 0-1 | TX5_PEND raw interrupt read (before mask) |
| 4 | TX4_PEND | 0-1 | TX4_PEND raw interrupt read (before mask) |
| 3 | TX3_PEND | 0-1 | TX3_PEND raw interrupt read (before mask) |
| 2 | TX2_PEND | 0-1 | TX2_PEND raw interrupt read (before mask) |
| 1 | TX1_PEND | 0-1 | TX1_PEND raw interrupt read (before mask) |
| 0 | TX0_PEND | 0-1 | TX0_PEND raw interrupt read (before mask) |

### 9.4.1.53 CPDMA_INT TX Interrupt Status (masked value) Register (TX_INTSTAT_MASKED) (0x184)

**Figure 9-68. CPDMA_INT TX Interrupt Status (masked value) Register (TX_INTSTAT_MASKED)**

| 31 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX7_PEND | TX6_PEND | TX5_PEND | TX4_PEND | TX3_PEND | TX2_PEND | TX1_PEND | TX0_PEND |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-80. CPDMA_INT TX Interrupt Status (masked value) Register (TX_INTSTAT_MASKED) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero |
| 7 | TX7_PEND | 0-1 | TX7_PEND masked interrupt read |
| 6 | TX6_PEND | 0-1 | TX6_PEND masked interrupt read |
| 5 | TX5_PEND | 0-1 | TX5_PEND masked interrupt read |
| 4 | TX4_PEND | 0-1 | TX4_PEND masked interrupt read |
| 3 | TX3_PEND | 0-1 | TX3_PEND masked interrupt read |
| 2 | TX2_PEND | 0-1 | TX2_PEND masked interrupt read |
| 1 | TX1_PEND | 0-1 | TX1_PEND masked interrupt read |
| 0 | TX0_PEND | 0-1 | TX0_PEND masked interrupt read |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.54 CPDMA_INT TX Interrupt Mask Set Register (TX _INTMASK_SET) (0x188)

**Figure 9-69. CPDMA_INT TX Interrupt Mask Set Register (TX _INTMASK_SET)**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TX7_MASK | TX6_MASK | TX5_MASK | TX4_MASK | TX3_MASK | TX2_MASK | TX1_MASK | TX0_MASK |
| R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-81. CPDMA_INT TX Interrupt Mask Set Register (TX _INTMASK_SET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero |
| 7 | TX7_MASK | 0-1 | TX Channel 7 Mask. Write one to enable interrupt. |
| 6 | TX6_MASK | 0-1 | TX Channel 6 Mask. Write one to enable interrupt. |
| 5 | TX5_MASK | 0-1 | TX Channel 5 Mask. Write one to enable interrupt. |
| 4 | TX4_MASK | 0-1 | TX Channel 4 Mask. Write one to enable interrupt. |
| 3 | TX3_MASK | 0-1 | TX Channel 3 Mask. Write one to enable interrupt. |
| 2 | TX2_MASK | 0-1 | TX Channel 2 Mask. Write one to enable interrupt. |
| 1 | TX1_MASK | 0-1 | TX Channel 1 Mask. Write one to enable interrupt. |
| 0 | TX0_MASK | 0-1 | TX Channel 0 Mask. Write one to enable interrupt. |

### 9.4.1.55 CPDMA_INT TX Interrupt Mask Clear Register (TX_INTMASK_CLEAR) (0x18C)

**Figure 9-70. CPDMA_INT TX Interrupt Mask Clear Register (TX_INTMASK_CLEAR)**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TX7_MASK | TX6_MASK | TX5_MASK | TX4_MASK | TX3_MASK | TX2_MASK | TX1_MASK | TX0_MASK |
| R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

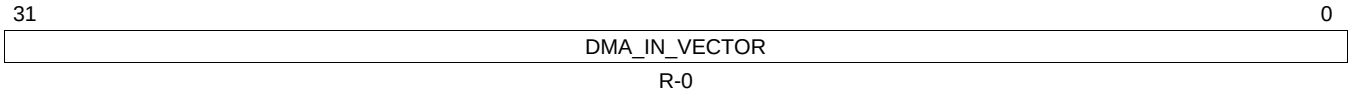
**Table 9-82. CPDMA_INT TX Interrupt Mask Clear Register (TX_INTMASK_CLEAR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero |
| 7 | TX7_MASK | 0-1 | TX Channel 7 Mask. Write one to disable interrupt. |
| 6 | TX6_MASK | 0-1 | TX Channel 6 Mask. Write one to disable interrupt. |
| 5 | TX5_MASK | 0-1 | TX Channel 5 Mask. Write one to disable interrupt. |
| 4 | TX4_MASK | 0-1 | TX Channel 4 Mask. Write one to disable interrupt. |
| 3 | TX3_MASK | 0-1 | TX Channel 3 Mask. Write one to disable interrupt. |
| 2 | TX2_MASK | 0-1 | TX Channel 2 Mask. Write one to disable interrupt. |
| 1 | TX1_MASK | 0-1 | TX Channel 1 Mask. Write one to disable interrupt. |
| 0 | TX0_MASK | 0-1 | TX Channel 0 Mask. Write one to disable interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.56 CPDMA_INT Input Vector (CPDMA_IN_VECTOR) (0x190)

**Figure 9-71. CPDMA_INT Input Vector (CPDMA_IN_VECTOR)**



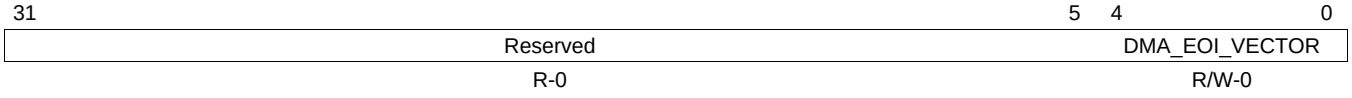

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-83. CPDMA_INT Input Vector (CPDMA_IN_VECTOR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | DMA_IN_VECTOR | 0-FFFF FFFFh | DMA Input Vector. The value of DMA_IN_VECTOR is reset to zero, but will change to the **IN_VECTOR** bus value one clock after reset is deasserted. Thereafter, this value will change to a new **IN_VECTOR** value one clock after the **IN_VECTOR** value changes. |

### 9.4.1.57 CPDMA_INT End Of Interrupt Vector (CPDMA_EOI_VECTOR) (0x194)

**Figure 9-72. CPDMA_INT End Of Interrupt Vector (CPDMA_EOI_VECTOR)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-84. CPDMA_INT End Of Interrupt Vector (CPDMA_EOI_VECTOR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Read as zero. |
| 4-0 | DMA_EOI_VECTOR | 0-1Fh | DMA End of Interrupt Vector. The **EOI_VECTOR(4:0)** pins reflect the value written to this location one **CLK** cycle after a write to this location. The **EOI_WR** signal is asserted for a single clock cycle after a latency of two **CLK** cycles when a write is performed to this location. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.58 CPDMA_INT RX Interrupt Status (raw value) Register (RX_INTSTAT_RAW) (0x1A0)

**Figure 9-73. CPDMA_INT RX Interrupt Status (raw value) Register (RX_INTSTAT_RAW)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RX7_THRESH_PEND | RX6_THRESH_PEND | RX5_THRESH_PEND | RX4_THRESH_PEND | RX3_THRESH_PEND | RX2_THRESH_PEND | RX1_THRESH_PEND | RX0_THRESH_PEND |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RX7_PEND | RX6_PEND | RX5_PEND | RX4_PEND | RX3_PEND | RX2_PEND | RX1_PEND | RX0_PEND |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 9-85. CPDMA_INT RX Interrupt Status (raw value) Register (RX_INTSTAT_RAW) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero. |
| 15 | RX7_THRESH_PEND | 0-1 | **RX7_THRESH_PEND** raw interrupt read (before mask). |
| 14 | RX6_THRESH_PEND | 0-1 | **RX6_THRESH_PEND** raw interrupt read (before mask). |
| 13 | RX5_THRESH_PEND | 0-1 | **RX5_THRESH_PEND** raw interrupt read (before mask). |
| 12 | RX4_THRESH_PEND | 0-1 | **RX4_THRESH_PEND** raw interrupt read (before mask). |
| 11 | RX3_THRESH_PEND | 0-1 | **RX3_THRESH_PEND** raw interrupt read (before mask). |
| 10 | RX2_THRESH_PEND | 0-1 | **RX2_THRESH_PEND** raw interrupt read (before mask). |
| 9 | RX1_THRESH_PEND | 0-1 | **RX1_THRESH_PEND** raw interrupt read (before mask). |
| 8 | RX0_THRESH_PEND | 0-1 | **RX0_THRESH_PEND** raw interrupt read (before mask). |
| 7 | RX7_PEND | 0-1 | **RX7_PEND** raw interrupt read (before mask). |
| 6 | RX6_PEND | 0-1 | **RX6_PEND** raw interrupt read (before mask). |
| 5 | RX5_PEND | 0-1 | **RX5_PEND** raw interrupt read (before mask). |
| 4 | RX4_PEND | 0-1 | **RX4_PEND** raw interrupt read (before mask). |
| 3 | RX3_PEND | 0-1 | **RX3_PEND** raw interrupt read (before mask). |
| 2 | RX2_PEND | 0-1 | **RX2_PEND** raw interrupt read (before mask). |
| 1 | RX1_PEND | 0-1 | **RX1_PEND** raw interrupt read (before mask). |
| 0 | RX0_PEND | 0-1 | **RX0_PEND** raw interrupt read (before mask). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.59 CPDMA_INT RX Interrupt Status (masked value) Register (RX_INTSTAT_MASKED) (0x1A4)

**Figure 9-74. CPDMA_INT RX Interrupt Status (masked value) Register (RX_INTSTAT_MASKED)**

| 31 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RX7_THRESH_PEND | RX6_THRESH_PEND | RX5_THRESH_PEND | RX4_THRESH_PEND | RX3_THRESH_PEND | RX2_THRESH_PEND | RX1_THRESH_PEND | RX0_THRESH_PEND |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RX7_PEND | RX6_PEND | RX5_PEND | RX4_PEND | RX3_PEND | RX2_PEND | RX1_PEND | RX0_PEND |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-86. CPDMA_INT RX Interrupt Status (masked value) Register (RX_INTSTAT_MASKED) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero. |
| 15 | RX7_THRESH_PEND | 0-1 | **RX7_THRESH_PEND** masked interrupt read. |
| 14 | RX6_THRESH_PEND | 0-1 | **RX6_THRESH_PEND** masked interrupt read. |
| 13 | RX5_THRESH_PEND | 0-1 | **RX5_THRESH_PEND** masked interrupt read. |
| 12 | RX4_THRESH_PEND | 0-1 | **RX4_THRESH_PEND** masked interrupt read. |
| 11 | RX3_THRESH_PEND | 0-1 | **RX3_THRESH_PEND** masked interrupt read. |
| 10 | RX2_THRESH_PEND | 0-1 | **RX2_THRESH_PEND** masked interrupt read. |
| 9 | RX1_THRESH_PEND | 0-1 | **RX1_THRESH_PEND** masked interrupt read. |
| 8 | RX0_THRESH_PEND | 0-1 | **RX0_THRESH_PEND** masked interrupt read. |
| 7 | RX7_PEND | 0-1 | **RX7_PEND** masked interrupt read. |
| 6 | RX6_PEND | 0-1 | **RX6_PEND** masked interrupt read. |
| 5 | RX5_PEND | 0-1 | **RX5_PEND** masked interrupt read. |
| 4 | RX4_PEND | 0-1 | **RX4_PEND** masked interrupt read. |
| 3 | RX3_PEND | 0-1 | **RX3_PEND** masked interrupt read. |
| 2 | RX2_PEND | 0-1 | **RX2_PEND** masked interrupt read. |
| 1 | RX1_PEND | 0-1 | **RX1_PEND** masked interrupt read. |
| 0 | RX0_PEND | 0-1 | **RX0_PEND** masked interrupt read. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.60 CPDMA_INT RX Interrupt Mask Set Register (RX_INTMASK_SET) (0x1A8)

**Figure 9-75. CPDMA_INT RX Interrupt Mask Set Register (RX_INTMASK_SET)**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RX7_THRESH_PEND_MASK | RX6_THRESH_PEND_MASK | RX5_THRESH_PEND_MASK | RX4_THRESH_PEND_MASK | RX3_THRESH_PEND_MASK | RX2_THRESH_PEND_MASK | RX1_THRESH_PEND_MASK | RX0_THRESH_PEND_MASK |
| R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RX7_PEND_MASK | RX6_PEND_MASK | RX5_PEND_MASK | RX4_PEND_MASK | RX3_PEND_MASK | RX2_PEND_MASK | RX1_PEND_MASK | RX0_PEND_MASK |
| R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 | R/WS-0 |

LEGEND: R/W = Read/Write; WS = Write/Set; R = Read only; -*n* = value after reset

**Table 9-87. CPDMA_INT RX Interrupt Mask Set Register (RX_INTMASK_SET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero. |
| 15 | RX7_THRESH_PEND_MASK | 0-1 | RX Channel 7 Threshold Pending Interrupt Mask. Write one to enable interrupt. |
| 14 | RX6_THRESH_PEND_MASK | 0-1 | RX Channel 6 Threshold Pending Interrupt Mask. Write one to enable interrupt. |
| 13 | RX5_THRESH_PEND_MASK | 0-1 | RX Channel 5 Threshold Pending Interrupt Mask. Write one to enable interrupt. |
| 12 | RX4_THRESH_PEND_MASK | 0-1 | RX Channel 4 Threshold Pending Interrupt Mask. Write one to enable interrupt. |
| 11 | RX3_THRESH_PEND_MASK | 0-1 | RX Channel 3 Threshold Pending Interrupt Mask. Write one to enable interrupt. |
| 10 | RX2_THRESH_PEND_MASK | 0-1 | RX Channel 2 Threshold Pending Interrupt Mask. Write one to enable interrupt. |
| 9 | RX1_THRESH_PEND_MASK | 0-1 | RX Channel 1 Threshold Pending Interrupt Mask. Write one to enable interrupt. |
| 8 | RX0_THRESH_PEND_MASK | 0-1 | RX Channel 0 Threshold Pending Interrupt Mask. Write one to enable interrupt. |
| 7 | RX7_PEND_MASK | 0-1 | RX Channel 7 Pending Interrupt Mask. Write one to enable interrupt. |
| 6 | RX6_PEND_MASK | 0-1 | RX Channel 6 Pending Interrupt Mask. Write one to enable interrupt. |
| 5 | RX5_PEND_MASK | 0-1 | RX Channel 5 Pending Interrupt Mask. Write one to enable interrupt. |
| 4 | RX4_PEND_MASK | 0-1 | RX Channel 4 Pending Interrupt Mask. Write one to enable interrupt. |
| 3 | RX3_PEND_MASK | 0-1 | RX Channel 3 Pending Interrupt Mask. Write one to enable interrupt. |
| 2 | RX2_PEND_MASK | 0-1 | RX Channel 2 Pending Interrupt Mask. Write one to enable interrupt. |
| 1 | RX1_PEND_MASK | 0-1 | RX Channel 1 Pending Interrupt Mask. Write one to enable interrupt. |
| 0 | RX0_PEND_MASK | 0-1 | RX Channel 0 Pending Interrupt Mask. Write one to enable interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.61 CPDMA_INT RX Interrupt Mask Clear Register (RX_INTMASK_CLEAR) (0x1AC)

**Figure 9-76. CPDMA_INT RX Interrupt Mask Clear Register (RX_INTMASK_CLEAR)**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RX7_THRESH_PEND_MASK | RX6_THRESH_PEND_MASK | RX5_THRESH_PEND_MASK | RX4_THRESH_PEND_MASK | RX3_THRESH_PEND_MASK | RX2_THRESH_PEND_MASK | RX1_THRESH_PEND_MASK | RX0_THRESH_PEND_MASK |
| R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RX7_PEND_MASK | RX6_PEND_MASK | RX5_PEND_MASK | RX4_PEND_MASK | RX3_PEND_MASK | RX2_PEND_MASK | RX1_PEND_MASK | RX0_PEND_MASK |
| R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 | R/WC-0 |

LEGEND: R/W = Read/Write; WC = Write/Clear; R = Read only; -*n* = value after reset

**Table 9-88. CPDMA_INT RX Interrupt Mask Clear Register (RX_INTMASK_CLEAR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero. |
| 15 | RX7_THRESH_PEND_MASK | 0-1 | RX Channel 7 Threshold Pending Interrupt Mask . Write one to disable interrupt. |
| 14 | RX6_THRESH_PEND_MASK | 0-1 | RX Channel 6 Threshold Pending Interrupt Mask. Write one to disable interrupt. |
| 13 | RX5_THRESH_PEND_MASK | 0-1 | RX Channel 5 Threshold Pending Interrupt Mask. Write one to disable interrupt. |
| 12 | RX4_THRESH_PEND_MASK | 0-1 | RX Channel 4 Threshold Pending Interrupt Mask. Write one to disable interrupt. |
| 11 | RX3_THRESH_PEND_MASK | 0-1 | RX Channel 3 Threshold Pending Interrupt Mask. Write one to disable interrupt. |
| 10 | RX2_THRESH_PEND_MASK | 0-1 | RX Channel 2 Threshold Pending Interrupt Mask. Write one to disable interrupt. |
| 9 | RX1_THRESH_PEND_MASK | 0-1 | RX Channel 1 Threshold Pending Interrupt Mask. Write one to disable interrupt. |
| 8 | RX0_THRESH_PEND_MASK | 0-1 | RX Channel 0 Threshold Pending Interrupt Mask. Write one to disable interrupt. |
| 7 | RX7_PEND_MASK | 0-1 | RX Channel 7 Pending Interrupt Mask. Write one to disable interrupt. |
| 6 | RX6_PEND_MASK | 0-1 | RX Channel 6 Pending Interrupt Mask. Write one to disable interrupt. |
| 5 | RX5_PEND_MASK | 0-1 | RX Channel 5 Pending Interrupt Mask. Write one to disable interrupt. |
| 4 | RX4_PEND_MASK | 0-1 | RX Channel 4 Pending Interrupt Mask. Write one to disable interrupt. |
| 3 | RX3_PEND_MASK | 0-1 | RX Channel 3 Pending Interrupt Mask. Write one to disable interrupt. |
| 2 | RX2_PEND_MASK | 0-1 | RX Channel 2 Pending Interrupt Mask. Write one to disable interrupt. |
| 1 | RX1_PEND_MASK | 0-1 | RX Channel 1 Pending Interrupt Mask. Write one to disable interrupt. |
| 0 | RX0_PEND_MASK | 0-1 | RX Channel 0 Pending Interrupt Mask. Write one to disable interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.62 CPDMA_INT DMA Interrupt Status (raw value) Register (DMA_INTSTAT_RAW) (0x1B0)

**Figure 9-77. CPDMA_INT DMA Interrupt Status (raw value) Register (DMA_INTSTAT_RAW)**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | HOST_PEND | STAT_PEND |
| R-0 | | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-89. CPDMA_INT DMA Interrupt Status (raw value) Register (DMA_INTSTAT_RAW) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Read as zero. |
| 1 | HOST_PEND | 0-1 | Host Pending Interrupt. Raw interrupt read (before mask). |
| 0 | STAT_PEND | 0-1 | Statistics Pending Interrupt. Raw interrupt read (before mask). |

### 9.4.1.63 CPDMA_INT DMA Interrupt Status (masked value) Register (DMA_INTSTAT_MASKED) (0x1B4)

**Figure 9-78. CPDMA_INT DMA Interrupt Status (masked value) Register (DMA_INTSTAT_MASKED)**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | HOST_PEND | STAT_PEND |
| R-0 | | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

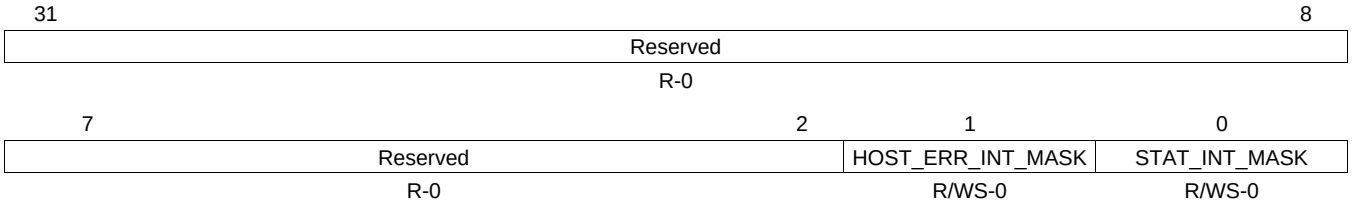
**Table 9-90. CPDMA_INT DMA Interrupt Status (masked value) Register (DMA_INTSTAT_MASKED) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Read as zero. |
| 1 | HOST_PEND | 0-1 | Host Pending Interrupt. Masked interrupt read. |
| 0 | STAT_PEND | 0-1 | Statistics Pending Interrupt. Masked interrupt read. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.64 CPDMA_INT DMA Interrupt Mask Set Register (DMA_INTMASK_SET) (0x1B8)

**Figure 9-79. CPDMA_INT DMA Interrupt Mask Set Register (DMA_INTMASK_SET)**



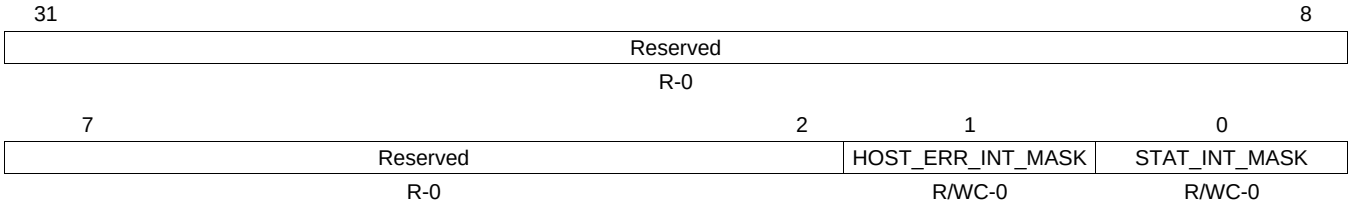

LEGEND: R/W = Read/Write; WS = Write/Set; R = Read only; -*n* = value after reset

**Table 9-91. CPDMA_INT DMA Interrupt Mask Set Register (DMA_INTMASK_SET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Read as zero. |
| 1 | HOST_ERR_INT_MASK | 0-1 | Host Error Interrupt Mask. Write one to enable interrupt. |
| 0 | STAT_INT_MASK | 0-1 | Statistics Interrupt Mask. Write one to enable interrupt. |

### 9.4.1.65 CPDMA_INT DMA Interrupt Mask Clear Register (DMA_INTMASK_CLEAR) (0x1BC)

**Figure 9-80. CPDMA_INT DMA Interrupt Mask Clear Register (DMA_INTMASK_CLEAR)**




LEGEND: R/W = Read/Write; WS = Write/Set; R = Read only; -*n* = value after reset

**Table 9-92. CPDMA_INT DMA Interrupt Mask Clear Register (DMA_INTMASK_CLEAR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Read as zero. |
| 1 | HOST_ERR_INT_MASK | 0-1 | Host Error Interrupt Mask - Write one to disable interrupt. |
| 0 | STAT_INT_MASK | 0-1 | Statistics Interrupt Mask - Write one to disable interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.66 CPDMA_INT RX Channel Threshold Pending Registers (RX0_PENDTHRESH-RX7_PENDTHRESH) (0x1C0-0x1DC)

**Figure 9-81. CPDMA_INT RX Channel Threshold Pending Channel Registers**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RX(0..7)_PENDTHRESH | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-93. CPDMA_INT RX Channel Threshold Pending Channel Registers Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero. |
| 7-0 | RX(0..7)_PENDTHRESH | 0-FFh | RX Flow Threshold. This field contains the threshold value for issuing receive threshold pending interrupts (when enabled). |

### 9.4.1.67 CPDMA_INT RX Channel Free Buffer Count Registers (RX0_FREEBUFFER-RX7_FREEBUFFER) (0x1E0-0x1FC)

**Figure 9-82. CPDMA_INT RX Channel Free Buffer Count Registers**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | RX(0..7)_FREEBUF | |
| R-0 | | WI-0 | |

LEGEND: R/W = Read/Write; WI = Write to increment field; R = Read only; -*n* = value after reset

**Table 9-94. CPDMA_INT RX Channel Free Buffer Count Registers Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Read as zero. |
| 15-0 | RX(0..7)_FREEBUF | 0-FFFFh | RX Free Buffer Count. This field contains the count of free buffers available. The RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asserted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing. |

### 9.4.1.68 CPDMA_STATERAM TX Channel Head Descriptor Pointers (TX0_HDP-TX7_HDP) (0x200-0x21C)

**Figure 9-83. CPDMA_STATERAM TX Channel Head Descriptor Pointers**

| 31 | 0 |
|---|---|
| TX(0..7)_HDP | |
| R/W-X | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; X = Unknown

**Table 9-95. CPDMA_STATERAM TX Channel Head Descriptor Pointers Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | TX(0..7)_HDP | 0-FFFF FFFFh | TX Channel (0..7) DMA Head Descriptor Pointer. Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.69 CPDMA_STATERAM RX Channel Head Descriptor Pointers (RX0_HDP-RX7_HDP) (0x220-0x23C)

**Figure 9-84. CPDMA_STATERAM RX Channel Head Descriptor Pointers**

| 31 | 0 |
|---|---|
| RX(0..7)_HDP | |
| R/W-X | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; X = Unknown

**Table 9-96. CPDMA_STATERAM RX Channel Head Descriptor Pointers Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | RX(0..7)_HDP | 0-FFFF FFFFh | RX DMA Head Descriptor Pointer. Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset. |

### 9.4.1.70 CPDMA_STATERAM TX Channel Completion Pointer Registers (TX0_CP-TX7_CP) (0x240-0x25C)

**Figure 9-85. CPDMA_STATERAM TX Channel Completion Pointer Registers**

| 31 | 0 |
|---|---|
| TX(0..7)_CP | |
| R/W-X | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; X = Unknown

**Table 9-97. CPDMA_STATERAM TX Channel Completion Pointer Registers Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | TX(0..7)_CP | 0-FFFF FFFFh | TX Completion Pointer Register. This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. |

### 9.4.1.71 CPDMA_STATERAM RX Channel Completion Pointer Registers (RX0_CP-RX7_CP) (0x260-0x27C)

The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

**Figure 9-86. CPDMA_STATERAM RX Channel Completion Pointer Registers**

| 31 | 0 |
|---|---|
| RX(0..7)_CP | |
| R/W-X | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; X = Unknown

**Table 9-98. CPDMA_STATERAM RX Channel Completion Pointer Registers Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | RX(0..7)_CP | 0-FFFF FFFFh | RX Completion Pointer Register. This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.72 RX (only) Statistics Descriptions (0x400-0x430)

The EMAC has a set of statistics that record events associated with frame traffic. The statistics values are cleared to zero 38 clocks after the rising edge of reset. When the GMII_EN bit in the MAC control register is set, all statistics registers (see Figure 9-87) are write-to-decrement. The value written is subtracted from the register value with the result stored in the register. If a value greater than the statistics value is written, then zero is written to the register (writing FFFF FFFFh clears a statistics location). When the GMII_EN bit is cleared, all statistics registers are read/write (normal write direct, so writing 0000 0000h clears a statistics location). All write accesses must be 32-bit accesses.

The statistics interrupt is issued, if enabled, when any statistics value is greater than or equal to 8000 0000h. The statistics interrupt is removed by writing to decrement any statistics value greater than 8000 0000h. The statistics are mapped into internal memory space and are 32-bits wide. All statistics rollover from FFFF FFFFh to 0000 0000h.

**Figure 9-87. Statistics Register**

| 31 | 0 |
|---|---|
| COUNT | |
| R/WD-0 | |

LEGEND: R/W = Read/Write; WD = Write to decrement; -*n* = value after reset

#### *9.4.1.72.1 Good RX Frames (0x400)*

The total number of good frames received on the port. A good frame is defined to be:

- any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was of length 64 to RX_MAXLEN bytes inclusive, and
- had no CRC error, alignment error or code error.

See the **RX Align/Code Errors** and **RX CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

#### *9.4.1.72.2 Broadcast RX Frames (0x404)*

The total number of good broadcast frames received on the port. A good broadcast frame is defined to be:

- any data or MAC control frame which was destined for address FFFF FFFF FFFFh only, and
- was of length 64 to RX_MAXLEN bytes inclusive, and
- had no CRC error, alignment error or code error.

See the **RX Align/Code Errors** and **RX CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

#### *9.4.1.72.3 Multicast RX Frames (0x408)*

The total number of good multicast frames received on the port. A good multicast frame is defined to be:

- any data or MAC control frame which was destined for any multicast address other than FFFF FFFF FFFFh, and
- was of length 64 to RX_MAXLEN bytes inclusive, and
- had no CRC error, alignment error or code error.

See the **RX Align/Code Errors** and **RX CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.4.1.72.4 Pause RX Frames (0x40C)

The total number of IEEE 802.3X pause frames received by the port (whether acted upon or not). Such a frame:

- contained any unicast, broadcast, or multicast address, and
- contained the length/type field value 88.08 (hex) and the opcode 0001h, and
- was of length 64 to RX_MAXLEN bytes inclusive, and
- had no CRC error, alignment error or code error, and
- pause-frames had been enabled on the port (tx_flow_en = 1).

The port could have been in either half or full-duplex mode.

See the **RX Align/Code Errors** and **RX CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic

#### 9.4.1.72.5 RX CRC Errors (0x410)

The total number of frames received on the port that experienced a CRC error. Such a frame:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was of length 64 to RX_MAXLEN bytes inclusive, and
- had no code/align error, and
- had a CRC error.

Overruns have no effect upon this statistic.

A CRC error is defined to be:

- a frame containing an even number of nibbles, and
- fails the Frame Check Sequence test.

#### 9.4.1.72.6 RX Align/Code Errors (0x414)

The total number of frames received on the port that experienced an alignment error or code error. Such a frame:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was of length 64 to RX_MAXLEN bytes inclusive, and
- had either an alignment error or a code error.

Over-runs have no effect upon this statistic.

An alignment error is defined to be:

- a frame containing an odd number of nibbles, and
- also fails the Frame Check Sequence test if the final nibble is ignored.

A code error is defined to be:

- a frame which has been discarded because the ports **MRXER** pin driven with a one for at least one bit-time's duration at any point during the frame's reception.

Note: RFC 1757 etherStatsCRCAlignErrors Ref. 1.5 can be calculated by summing **RX Align/Code Errors** and **RX CRC errors**.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.4.1.72.7 Oversize RX Frames (0x418)

The total number of oversized frames received on the port. An oversized frame is defined to be:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was greater than RX_MAXLEN in bytes, and
- had no CRC error, alignment error or code error.

See the **RX Align/Code Errors** and **RX CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

#### 9.4.1.72.8 RX Jabbers (0x41C)

The total number of jabber frames received on the port. A jabber frame is:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was greater than RX_MAXLEN bytes long, and
- had a CRC error, an alignment error, or a code error.

See the **RX Align/Code Errors** and **RX CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

#### 9.4.1.72.9 Undersize (Short) RX Frames (0x420)

The total number of undersized frames received on the port. An undersized frame is defined to be:

- any data frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was less than 64 bytes long, and
- had no CRC error, alignment error or code error.

See the **RX Align/Code Errors** and **RX CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

#### 9.4.1.72.10 RX Fragments (0x424)

The total number of frame fragments received on the port. A frame fragment is defined to be:

- any data frame (address matching does not matter), and
- was less than 64 bytes long, and
- had a CRC error, an alignment error, or a code error, and
- was not the result of a collision caused by half duplex, collision based flow control.

See the **RX Align/Code Errors** and **RX CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

#### 9.4.1.72.11 RX Octets (0x430)

The total number of bytes in all good frames received on the port. A good frame is defined to be:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was of length 64 to RX_MAXLEN bytes inclusive, and
- had no CRC error, alignment error or code error.

See the **RX Align/Code Errors** and **RX CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.4.1.72.12 RX Start of Frame Overruns (0x484)

The total number of frames received on the port that had a CPDMA start of frame (SOF) overrun or were dropped by due to FIFO resource limitations. SOF overrun frame is defined to be:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > RX_MAXLEN bytes), and
- The CPDMA had a start of frame overrun or the packet was dropped due to FIFO resource limitations.

#### 9.4.1.72.13 RX Middle of Frame Overruns (0x488)

The total number of frames received on the port that had a CPDMA middle of frame (MOF) overrun. MOF overrun frame is defined to be:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > RX_MAXLEN bytes), and
- the CPDMA had a middle of frame overrun.

#### 9.4.1.72.14 RX DMA Overruns (0x48C)

The total number of frames received on the port that had either a DMA start of frame (SOF) overrun or a DMA MOF overrun. An RX DMA overrun frame is defined to be:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > RX_MAXLEN bytes), and
- the CPGMAC_SL was unable to receive it because it did not have the DMA buffer resources to receive it (zero head descriptor pointer at the start or during the middle of the frame reception).

CRC errors, alignment errors and code errors have no effect upon this statistic.


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.73 TX (only) Statistics Descriptions (0x434 -0x464)

The EMAC has a set of statistics that record events associated with frame traffic. The statistics values are cleared to zero 38 clocks after the rising edge of reset. When the GMII_EN bit in the MAC control register is set, all statistics registers (see Figure 9-88) are write-to-decrement. The value written is subtracted from the register value with the result stored in the register. If a value greater than the statistics value is written, then zero is written to the register (writing FFFF FFFFh clears a statistics location). When the GMII_EN bit is cleared, all statistics registers are read/write (normal write direct, so writing 0000 0000h clears a statistics location). All write accesses must be 32-bit accesses.

The statistics interrupt is issued, if enabled, when any statistics value is greater than or equal to 8000 0000h. The statistics interrupt is removed by writing to decrement any statistics value greater than 8000 0000h. The statistics are mapped into internal memory space and are 32-bits wide. All statistics rollover from FFFF FFFFh to 0000 0000h.

The maximum and minimum transmit frame size is software controllable.

**Figure 9-88. Statistics Register**

| 31 | 0 |
|---|---|
| COUNT | |
| R/WD-0 | |

LEGEND: R/W = Read/Write; WD = Write to decrement; -*n* = value after reset

#### *9.4.1.73.1 Good TX Frames (0x434)*

The total number of good frames transmitted on the port. A good frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- was any length, and
- had no late or excessive collisions, no carrier loss and no underrun.

#### *9.4.1.73.2 Broadcast TX Frames (0x438)*

The total number of good broadcast frames transmitted on the port. A good broadcast frame is defined to be:

- any data or MAC control frame destined for address FFFF FFFF FFFFh only, and
- was of any length, and
- had no late or excessive collisions, no carrier loss and no underrun

#### *9.4.1.73.3 Multicast TX Frames (0x43C)*

The total number of good multicast frames transmitted on the port. A good multicast frame is defined to be:

- any data or MAC control frame destined for any multicast address other than FFFF FFFF FFFFh, and
- was of any length, and
- had no late or excessive collisions, no carrier loss and no underrun.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.4.1.73.4 Pause TX Frames (0x440)

This statistic indicates the number of IEEE 802.3X pause frames transmitted by the port.

Pause frames cannot underrun or contain a CRC error because they are created in the transmitting MAC, so these error conditions have no effect upon the statistic. Pause frames sent by software will not be included in this count.

Since pause frames are only transmitted in full duplex carrier loss and collisions have no effect upon this statistic.

Transmitted pause frames are always 64 byte multicast frames so will appear in the **TX Multicast Frames** and **64octect Frames** statistics.

#### 9.4.1.73.5 Deferred TX Frames (0x444)

The total number of frames transmitted on the port that first experienced deferment. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no carrier loss and no underrun, and
- experienced no collisions before being successfully transmitted, and
- found the medium busy when transmission was first attempted, so had to wait.

CRC errors have no effect upon this statistic

See RFC1623 Ref. 2.6 dot3StatsDefferredTransmissions.

#### 9.4.1.73.6 Collisions (0x448)

This statistic records the total number of times that the port experienced a collision. Collisions occur under two circumstances.

- When a transmit data or MAC control frame...
- was destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no carrier loss and no underrun, and
- experienced a collision. A jam sequence is sent for every non-late collision, so this statistic will increment on each occasion if a frame experiences multiple collisions (and increments on late collisions).

CRC errors have no effect upon this statistic.

- When the port is in half-duplex mode, flow control is active, and a frame reception begins.

#### 9.4.1.73.7 Single Collision TX Frames (0x44C)

The total number of frames transmitted on the port that experienced exactly one collision. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no carrier loss and no underrun, and
- experienced one collision before successful transmission. The collision was not late.

CRC errors have no effect upon this statistic.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.4.1.73.8 Multiple Collision TX Frames (0x450)

The total number of frames transmitted on the port that experienced multiple collisions. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no carrier loss and no underrun, and
- experienced 2 to 15 collisions before being successfully transmitted. None of the collisions were late.

CRC errors have no effect upon this statistic.

#### 9.4.1.73.9 Excessive Collisions (0x454)

The total number of frames for which transmission was abandoned due to excessive collisions. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no carrier loss and no underrun, and
- experienced 16 collisions before abandoning all attempts at transmitting the frame. None of the collisions were late.

CRC errors have no effect upon this statistic.

#### 9.4.1.73.10 Late Collisions (0x458)

The total number of frames on the port for which transmission was abandoned because they experienced a late collision. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- experienced a collision later than 512 bit-times into the transmission. There may have been up to 15 previous (non-late) collisions which had previously required the transmission to be re-attempted. The **Late Collisions** statistic dominates over the **single, multiple,** and **excessive Collisions** statistics - if a late collision occurs the frame will not be counted in any of these other three statistics.

CRC errors, carrier loss, and underrun have no effect upon this statistic.

#### 9.4.1.73.11 TX Underrun (0x45C)

There should be no transmitted frames that experience underrun.

#### 9.4.1.73.12 Carrier Sense Errors (0x460)

The total number of frames on the port that experienced carrier loss. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- the carrier sense condition was lost or never asserted when transmitting the frame (the frame is not retransmitted). This is a transmit only statistic. Carrier Sense is a don't care for received frames. Transmit frames with carrier sense errors are sent until completion and are not aborted.
- CRC errors and underrun have no effect upon this statistic.

#### 9.4.1.73.13 TX Octets (0x464)

The total number of bytes in all good frames transmitted on the port. A good frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no late or excessive collisions, no carrier loss and no underrun.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.4.1.74 RX and TX (shared) Statistics Descriptions (0x468-0x480)

The EMAC has a set of statistics that record events associated with frame traffic. The statistics values are cleared to zero, 38 clocks after the rising edge of reset. When the GMII_EN bit in the MAC control register is set, all statistics registers (see Figure 9-89) are write-to-decrement. The value written is subtracted from the register value with the result stored in the register. If a value greater than the statistics value is written, then zero is written to the register (writing FFFF FFFFh clears a statistics location). When the GMII_EN bit is cleared, all statistics registers are read/write (normal write direct, so writing 0000 0000h clears a statistics location). All write accesses must be 32-bit accesses.

The statistics interrupt is issued, if enabled, when any statistics value is greater than or equal to 8000 0000h. The statistics interrupt is removed by writing to decrement any statistics value greater than 8000 0000h. The statistics are mapped into internal memory space and are 32-bits wide. All statistics rollover from FFFF FFFFh to 0000 0000h.

**Figure 9-89. Statistics Register**

| 31 | 0 |
|---|---|
| COUNT | |
| R/WD-0 | |

LEGEND: R/W = Read/Write; WD = Write to decrement; -*n* = value after reset

##### *9.4.1.74.1 RX + TX 64 Octet Frames (0x468)*

The total number of 64-byte frames received and transmitted on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was exactly 64 bytes long. (If the frame was being transmitted and experienced carrier loss that resulted in a frame of this size being transmitted, then the frame will be recorded in this statistic).

CRC errors, code/align errors and overruns do not affect the recording of frames in this statistic.

##### *9.4.1.74.2 RX + TX 65-127 Octet Frames (0x46C)*

The total number of frames of size 65 to 127 bytes received and transmitted on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was 65 to 127 bytes long.

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

##### *9.4.1.74.3 RX + TX 128-255 Octet Frames (0x470)*

The total number of frames of size 128 to 255 bytes received and transmitted on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was 128 to 255 bytes long.

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

For RX reference only, see RFC1757 Ref. 1.13 etherStatsPkts128to255Octets.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 9.4.1.74.4 RX + TX 256-511 Octet Frames (0x474)

The total number of frames of size 256 to 511 bytes received and transmitted on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was 256 to 511 bytes long.

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

#### 9.4.1.74.5 RX + TX 512-1023 Octet Frames (0x478)

The total number of frames of size 512 to 1023 bytes received and transmitted on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was 512 to 1023 bytes long.

CRC errors, code/align errors and overruns do not affect the recording of frames in this statistic.

#### 9.4.1.74.6 RX + TX 1024_Up Octet Frames (0x47C)

The total number of frames of size 1024 to RX_MAXLEN bytes for receive or 1024 up for transmit on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was 1024 to RX_MAXLEN bytes long on receive, or any size on transmit.

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

#### 9.4.1.74.7 Network Octet Frames (0x480)

The total number of bytes of frame data received and transmitted on the EMAC. Each frame counted has all of the following:

- Was any data or MAC control frame destined for any unicast, broadcast, or multicast address (address match does not matter)
- Was of any size (including less than 64-byte and greater than RX_MAXLEN-byte frames)

Also counted in this statistic is:

- Every byte transmitted before a carrier-loss was experienced
- Every byte transmitted before each collision was experienced (multiple retries are counted each time)
- Every byte received if the EMAC is in half-duplex mode until a jam sequence was transmitted to initiate flow control. (The jam sequence is not counted to prevent double-counting).

Error conditions such as alignment errors, CRC errors, code errors, overruns, and underruns do not affect the recording of bytes in this statistic. The objective of this statistic is to give a reasonable indication of Ethernet utilization.


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.75 Identification and Version Register (CPTS_IDVER) (0x500)

**Figure 9-90. Identification and Version Register (CPTS_IDVER)**

| 31–16 | 15–11 | 10–8 | 7–0 |
|---|---|---|---|
| TX_IDENT | RTL_VER | MAJ_VER | MINOR_VER |
| R-4E8Ah | R-0 | R-1 | R-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-99. Identification and Version Register (CPTS_IDVER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | TX_IDENT | 0-FFFFh | TX Identification Value |
| 15-11 | RTL_VER | 0-1Fh | RTL Version Value |
| 10-8 | MAJ_VER | 0-7h | Major Version Value |
| 7-0 | MINOR_VER | 0-FFh | Minor Version Value |

### 9.4.1.76 Time Sync Control Register (CPTS_CONTROL) (0x504)

**Figure 9-91. Time Sync Control Register (CPTS_CONTROL)**

| 31–24 |
|---|
| Reserved |
| R-0 |

| 23–16 |
|---|
| Reserved |
| R-0 |

| 15–12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|
| Reserved | HW4_TS_PUSH_EN | HW3_TS_PUSH_EN | HW2_TS_PUSH_EN | HW1_TS_PUSH_EN |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7–2 | 1 | 0 |
|---|---|---|
| Reserved | INT_TEST | CPTS_EN |
| R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-100. Time Sync Control Register (CPTS_CONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | Reserved | 0 | Reserved |
| 11 | HW4_TS_PUSH_EN | 0-1 | Hardware push 4 enable |
| 10 | HW3_TS_PUSH_EN | 0-1 | Hardware push 3 enable |
| 9 | HW2_TS_PUSH_EN | 0-1 | Hardware push 2 enable |
| 8 | HW1_TS_PUSH_EN | 0-1 | Hardware push 1 enable |
| 7-2 | Reserved | 0 | Reserved |
| 1 | INT_TEST | 0-1 | Interrupt Test. When set, this bit allows the raw interrupt to be written to facilitate interrupt test. |
| 0 | CPTS_EN | | Time Sync Enable. When disabled (cleared to 0), the RCLK domain is held in reset. |
| | | 0 | Time Sync Disabled |
| | | 1 | Time Sync Enabled |

### 9.4.1.77 Reference Clock Select Register (CPTS_RFTCLK_SEL) (0x508)


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 9-92. Reference Clock Select Register (CPTS_RFTCLK_SEL)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | RFTCLK_SEL | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-101. Reference Clock Select Register (CPTS_RFTCLK_SEL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Reserved |
| 4-0 | RFTCLK_SEL | 0-1Fh | Reference Clock Select. This signal is used to control an external multiplexer that selects one of up to 32 clocks for time sync reference (RFTCLK). This RFTCLK_SEL value can be written only when the CPTS_EN bit is cleared to 0 in the TS_Control register. |

### 9.4.1.78 Time Stamp Event Push Register (CPTS_TS_PUSH) (0x50C)

**Figure 9-93. Time Stamp Event Push Register (CPTS_TS_PUSH)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TS_PUSH |
| R-0 | | W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-102. Time Stamp Event Push Register (CPTS_TS_PUSH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | TS_PUSH | 0-1 | Time stamp event push. When a logic high is written to this bit a time stamp event is pushed onto the event FIFO. The time stamp value is the time of the write of this register, not the time of the event read. The time stamp value can then be read on interrupt via the event registers. Software should not push a second time stamp event onto the event FIFO until the first time stamp value has been read from the event FIFO (there should be only one time stamp event in the event FIFO at any given time). This bit is write only and always reads zero. |

### 9.4.1.79 Time Stamp Load Value Register (CPTS_TS_LOAD_VAL) (0x510)

**Figure 9-94. Time Stamp Load Value Register (CPTS_TS_LOAD_VAL)**

| 31 | 0 |
|---|---|
| TS_LOAD_VAL | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-103. Time Stamp Load Value Register (CPTS_TS_LOAD_VAL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | TS_LOAD_VAL | 0-FFFF FFFFh | Time Stamp Load Value. Writing the TS_LOAD_EN bit in the CPTS_TS_LOAD_EN register causes the value contained in this register to be written into the time stamp. The time stamp value is read by initiating a time stamp push event, not by reading this register. When reading this register, the value read is not the time stamp, but is the value that was last written to this register. |

### 9.4.1.80 Time Stamp Load Enable Register (CPTS_TS_LOAD_EN) (0x514)


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 9-95. Time Stamp Load Enable Register (CPTS_TS_LOAD_EN)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TS_LOAD_EN |
| R-0 | | W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-104. Time Stamp Load Enable Register (CPTS_TS_LOAD_EN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | TS_LOAD_EN | 0-1 | Time Stamp Load. Writing a 1 to this bit enables the time stamp value to be written via the CPTS_TS_LOAD_VAL register. This feature is included for test purposes. This bit is write only. |

### 9.4.1.81 Time Sync Interrupt Status Register Raw Register (CPTS_INTSTAT_RAW) (0x520)

**Figure 9-96. Time Sync Interrupt Status Register Raw Register (CPTS_INTSTAT_RAW)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TS_PEND_RAW |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-105. Time Sync Interrupt Status Register Raw Register (CPTS_INTSTAT_RAW) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | TS_PEND_RAW | 0-1 | TS_PEND_RAW interrupt read (before enable). Writable when INT_TEST = 1. A 1 in this bit indicates that there is one or more events in the event FIFO. |

### 9.4.1.82 Time Sync Interrupt Status Register Masked Register (CPTS_INTSTAT_MASKED) (0x524)

**Figure 9-97. Time Sync Interrupt Status Register Masked Register (CPTS_INTSTAT_MASKED)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TS_PEND |
| R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-106. Time Sync Interrupt Status Register Masked Register (CPTS_INTSTAT_MASKED) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | TS_PEND | 0-1 | TS_PEND masked interrupt read (after enable). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.83 Time Sync Interrupt Enable Register (CPTS_INT_ENABLE) (0x528)

**Figure 9-98. Time Sync Interrupt Enable Register (CPTS_INT_ENABLE)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TS_PEND_EN |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-107. Time Sync Interrupt Enable Register (CPTS_INT_ENABLE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | TS_PEND_EN | 0-1 | TS_PEND masked interrupt enable. |

### 9.4.1.84 Event Interrupt Pop Register (CPTS_EVENT_POP) (0x530)

**Figure 9-99. Event Interrupt Pop Register (CPTS_EVENT_POP)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | EVENT_POP |
| R-0 | | W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-108. Event Interrupt Pop Register (CPTS_EVENT_POP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | EVENT_POP | 0-1 | Event Pop. When a logic high is written to this bit an event is popped off the event FIFO. The event FIFO pop occurs as part of the interrupt process after the event has been read in the EVENT_LOW and EVENT_HIGH registers. Popping an event discards the event and causes the next event, if any, to be moved to the top of the FIFO ready to be read by software on interrupt. |

### 9.4.1.85 Event Low Register (CPTS_EVENT_LOW) (0x534)

**Figure 9-100. Event Low Register (CPTS_EVENT_LOW)**

| 31 | 0 |
|---|---|
| TIME_STAMP | |
| R/W-X | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; X = Unknown

**Table 9-109. Event Low Register (CPTS_EVENT_LOW) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | TIME_STAMP | 0-FFFF FFFFh | Time Stamp. The timestamp is valid for transmit, receive, and time stamp push event types. The timestamp value is not valid for counter roll event types. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.86 Event High Register (CPTS_EVENT_HIGH) (0x538)

**Figure 9-101. Event High Register (CPTS_EVENT_HIGH)**

| 31–29 | 28–24 | 23–20 | 19–16 |
|---|---|---|---|
| Reserved | PORT_NUMBER | EVENT_TYPE | MESSAGE_TYPE |
| R-0 | R-X | R-X | R-X |

| 15–0 |
|---|
| SEQUENCE_ID |
| R-X |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; X = Unknown

**Table 9-110. Event High Register (CPTS_EVENT_HIGH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-29 | Reserved | 0 | |
| 28-24 | PORT_NUMBER | 0-1Fh | Port Number. Indicates the port number of an ethernet event. |
| 23-20 | EVENT_TYPE | | Time Sync Event Type |
| | | 0 | Time Stamp Push Event |
| | | 1h | Time Stamp Rollover Event |
| | | 2h | Time Stamp Half Rollover Event |
| | | 3h | Hardware Time Stamp Push Event |
| | | 4h | Ethernet Receive Event |
| | | 5h | Ethernet Transmit Event |
| | | 6h-Fh | Reserved |
| 19-16 | MESSAGE_TYPE | 0-Fh | Message type. The message type value that was contained in an ethernet transmit or receive time sync packet. This field is valid only for ethernet transmit or receive events. |
| 15-0 | SEQUENCE_ID | 0-FFFFh | Sequence ID. The 16-bit sequence id is the value that was contained in an ethernet transmit or receivetime sync packet. This field is valid only for ethernet transmit or receive events. |

### 9.4.1.87 Address Lookup Engine ID/Version Register (ALE_IDVER) (0x600)

**Figure 9-102. Address Lookup Engine ID/Version Register (ALE_IDVER)**

| 31–16 | 15–8 | 7–0 |
|---|---|---|
| ALE_IDENT | ALE_MAJ_VER | ALE_MINOR_VER |
| R-29h | R-1 | R-3h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-111. Address Lookup Engine ID/Version Register (ALE_IDVER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | ALE_IDENT | 0-FFFFh | Address Lookup Engine Identification Value |
| 15-8 | ALE_MAJ_VER | 0-FFh | Address Lookup Engine Major Version Value |
| 7-0 | ALE_MINOR_VER | 0-FFh | Address Lookup Engine Minor Version Value |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.88 Address Lookup Engine Control Register (ALE_CONTROL) (0x608)

**Figure 9-103. ALE Control Register (ALE_CONTROL)**

| 31 | 30 | 29 | 28 | 24 |
|---|---|---|---|---|
| ENABLE_ALE | CLEAR_TABLE | AGE_OUT_NOW | Reserved | |
| R/W-0 | R/W-0 | R/W-0 | R-0 | |

| 23 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| LEARN_NO_VID | EN_VID0_MODE | ENABLE_OUI_DENY | ALE_BYPASS | RATE_LIMIT_TX | ALE_VLAN_AWARE | ENABLE_AUTH_MODE | ENABLE_RATE_LIMIT |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-112. ALE Control Register (ALE_CONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | ENABLE_ALE | 0-1 | Enable ALE |
| 30 | CLEAR_TABLE | 0-1 | Clear ALE address table. Setting this bit causes the ALE hardware to write all table bit values to zero. Software must perform a clear table operation as part of the ALE setup/configuration process. Setting this bit causes all ALE accesses to be held up for 64 clocks while the clear is performed. Access to all ALE registers will be blocked (wait states) until the 64 clocks have completed. This bit cannot be read as one because the read is blocked until the clear table is completed at which time this bit is cleared to zero. |
| 29 | AGE_OUT_NOW | 0-1 | Age Out Address Table Now. Setting this bit causes the ALE hardware to remove (free up) any ageable table entry that does not have a set touch bit. This bit is cleared when the age out process has completed. This bit may be read. The age out process takes 4096 clocks best case (no ale packet processing during ageout) and 66550 clocks absolute worst case. |
| 28-8 | Reserved | 0 | Reserved |
| 7 | LEARN_NO_VID | | Learn No VID |
| | | 0 | VID is learned with the source address |
| | | 1 | VID is not learned with the source address (source address is not tied to VID). |
| 6 | EN_VID0_MODE | | Enable VLAN ID = 0 Mode |
| | | 0 | Process the packet with VID = PORT_VLAN[11:0]. |
| | | 1 | Process the packet with VID = 0. |
| 5 | ENABLE_OUI_DENY | 0-1 | Enable OUI Deny Mode. When set this bit indicates that a packet with a non OUI table entry matching source address will be dropped to the host unless the destination address matches a multicast table entry with the super bit set. |
| 4 | ALE_BYPASS | 0-1 | ALE Bypass. When set, all packets received on ports 1 and 2 are sent to the host (only to the host). |
| 3 | RATE_LIMIT_TX | | Rate Limit Transmit mode |
| | | 0 | Broadcast and multicast rate limit counters are received port based |
| | | 1 | Broadcast and multicast rate limit counters are transmit port based. |
| 2 | ALE_VLAN_AWARE | | ALE VLAN Aware. Determines what is done if VLAN not found. |
| | | 0 | Flood if VLAN not found |
| | | 1 | Drop packet if VLAN not found |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-112. ALE Control Register (ALE_CONTROL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | ENABLE_AUTH_MODE | | Enable MAC Authorization Mode. Mac authorization mode requires that all table entries be made by the host software. There are no learned address in authorization mode and the packet will be dropped if the source address is not found (and the destination address is not a multicast address with the super table entry bit set). |
| | | 0 | The ALE is not in MAC authorization mode |
| | | 1 | The ALE is in MAC authorization mode |
| 0 | ENABLE_RATE_LIMIT | | Enable Broadcast and Multicast Rate Limit |
| | | 0 | Broadcast/Multicast rates not limited |
| | | 1 | Broadcast/Multicast packet reception limited to the port control register rate limit fields. |

### 9.4.1.89 Address Lookup Engine Prescale Register (ALE_PRESCALE) (0x610)

**Figure 9-104. ALE Prescale Register (ALE_PRESCALE)**

| 31 | 19 | 0 |
|---|---|---|
| Reserved | ALE_PRESCALE | |
| R-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-113. ALE Prescale Register (ALE_PRESCALE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | Reserved | 0 | Reserved |
| 19-0 | ALE_PRESCALE | 0-F FFFFh | ALE Prescale Register. The input clock is divided by this value for use in the multicast/broadcast rate limiters. The minimum operating value is 10h. The prescaler is off when the value is zero. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.90 Address Lookup Engine Unknown VLAN Register (ALE_UNKNOWN_VLAN) (0x618)

**Figure 9-105. ALE Unknown VLAN Register (ALE_UNKNOWN_VLAN)**

| 31 30 | 29 24 | 23 22 | 21 16 |
|---|---|---|---|
| Reserved | UNKNOWN_FORCE_UNTAGGED_EGRESS | Reserved | UNKNOWN_REG_MCAST_FLOOD_MASK |
| R-0 | R/W-0 | R-0 | R/W-0 |

| 15 14 | 13 8 | 7 6 | 5 0 |
|---|---|---|---|
| Reserved | UNKNOWN_MCAST_FLOOD_MASK | Reserved | UNKNOWN_VLAN_MEMBER_LIST |
| R-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-114. ALE Unknown VLAN Register (ALE_UNKNOWN_VLAN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | |
| 29-24 | UNKNOWN_FORCE_ UNTAGGED_EGRESS | 0-3Fh | Unknown VLAN Force Untagged Egress. |
| 23-22 | Reserved | 0 | |
| 21-16 | UNKNOWN_REG_MCAST_ FLOOD_MASK | 0-3Fh | Unknown VLAN Registered Multicast Flood Mask |
| 15-14 | Reserved | 0 | |
| 13-8 | UNKNOWN_MCAST_ FLOOD_MASK | 0-3Fh | Unknown VLAN Multicast Flood Mask |
| 7-6 | Reserved | 0 | |
| 5-0 | UNKNOWN_VLAN_ MEMBER_LIST | 0-3Fh | Unknown VLAN Member List |

### 9.4.1.91 Address Lookup Engine Table Control Register (ALE_TBLCTL) (0x620)

**Figure 9-106. ALE Table Control Register (ALE_TBLCTL)**

| 31 | 30 16 |
|---|---|
| WRITE_RDZ | Reserved |
| R/W-0 | R-0 |

| 15 10 | 9 0 |
|---|---|
| Reserved | ENTRY_POINTER |
| R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

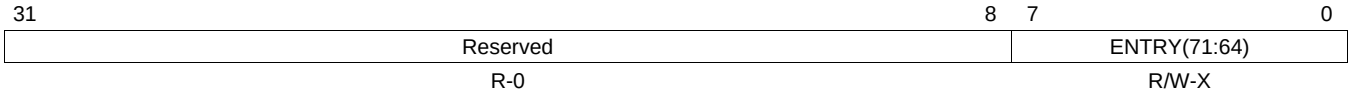
**Table 9-115. ALE Table Control Register (ALE_TBLCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | WRITE_RDZ | 0-1 | Write bit. This bit is always read as zero. Writing a 1 to this bit causes the three table word register values to be written to the ENTRY_POINTER location in the address table. Writing a 0 to this bit causes the three table word register values to be loaded from the ENTRY_POINTER location in the address table so that they may be subsequently read. A read of any ALE address location will be stalled until the read or write has completed. |
| 30-10 | Reserved | 0 | Reserved |
| 9-0 | ENTRY_POINTER | 0-3FFh | Table Entry Pointer. The ENTRY_POINTER contains the table entry value that will be read/written with accesses to the table word registers. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.92 Address Lookup Engine Table Word 2 Register (ALE_TBLW2) (0x634)

**Figure 9-107. ALE Table Word 2 Register (ALE_TBLW2)**



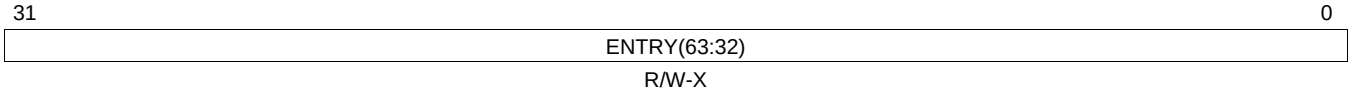

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; X = Unknown

**Table 9-116. ALE Table Word 2 Register (ALE_TBLW2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | |
| 7-0 | ENTRY(71:64) | 0-FFh | Table entry bits 71-64. |

### 9.4.1.93 Address Lookup Engine Table Word 1 Register (ALE_TBLW1) (0x638)

**Figure 9-108. ALE Table Word 1 Register (ALE_TBLW1)**



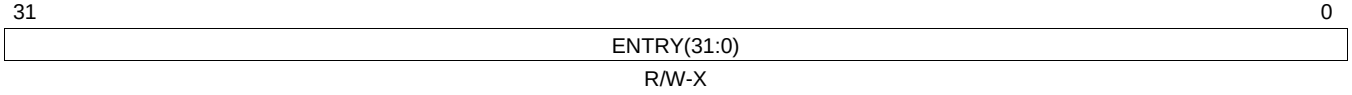

LEGEND: R/W = Read/Write; -*n* = value after reset; X = Unknown

**Table 9-117. ALE Table Word 1 Register (ALE_TBLW1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ENTRY(63:32) | 0-FFFF FFFFh | Table entry bits 63-32. |

### 9.4.1.94 Address Lookup Engine Table Word 0 Register (ALE_TBLW0) (0x63C)

**Figure 9-109. ALE Table Word 0 Register (ALE_TBLW0)**




LEGEND: R/W = Read/Write; -*n* = value after reset; X = Unknown

**Table 9-118. ALE Table Word 0 Register (ALE_TBLW0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ENTRY(31:0) | 0-FFFF FFFFh | Table entry bits 31-0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.95 Address Lookup Engine Port Control Registers (ALE_PORTCTL0-ALE_PORTCTL2) (0x640-0x648)

**Figure 9-110. ALE Port Control Registers**

| 31 | 24 |
|---|---|
| BCAST_LIMIT | |
| R/W-0 | |

| 23 | 16 |
|---|---|
| MCAST_LIMIT | |
| R/W-0 | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | NO_LEARN | VID_INGRESS_CHECK | DROP_UNTAGGED | PORT_STATE | |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 9-119. ALE Port Control Registers Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | BCAST_LIMIT | 0-FFh | Broadcast packet rate limit. Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field. |
| 23-16 | MCAST_LIMIT | 0-FFh | Multicast Packet Rate Limit. Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field. |
| 15-5 | Reserved | 0 | Reserved |
| 4 | MCAST_LIMIT | 0-1 | No Learn Mode. When set the port is disabled from learning or updating source addresses. |
| 3 | VID_INGRESS_CHECK | 0-1 | VLAN ID Ingress Check. If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped. |
| 2 | DROP_UNTAGGED | 0-1 | Drop Untagged Packets. Drop non-VLAN tagged ingress packets. |
| 1-0 | PORT_STATE | | Port State |
| | | 0 | Disabled |
| | | 1h | Blocked |
| | | 2h | Learn |
| | | 3h | Forward |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.96 CPGMAC_SL1 IDVER Register (SL1_IDVER) (0x700)

**Figure 9-111. CPGMAC_SL1 IDVER Register (SL1_IDVER)**

| 31 | 16 |
|---|---|
| IDENT | |
| R-17h | |

| 15 | 11 | 10 | 8 | 7 | 0 |
|---|---|---|---|---|---|
| Z | | X | | Y | |
| R-0 | | R-1 | | R-17h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 9-120. CPGMAC_SL1 IDVER Register (SL1_IDVER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | IDENT | 0-FFFFh | RX Identification Value |
| 15-11 | Z | 0-1Fh | RX Z value (X.Y.Z) |
| 10-8 | X | 0-7h | RX X value (major) |
| 7-0 | Y | 0-FFh | RX Y value (minor) |

### 9.4.1.97 CPGMAC_SL1 MAC Control Register (SL1_MACCONTROL) (0x704)

**Figure 9-112. CPGMAC_SL1 MAC Control Register (SL1_MACCONTROL)**

| 31 | 25 | 24 |
|---|---|---|
| Reserved | | RX_CMF_EN |
| R-0 | | R/W-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| RX_CSF_EN | RX_CEF_EN | TX_SHORT_GAP_LIM_EN | Reserved | | EXT_EN | GIG_FORCE | Reserved |
| R/W-0 | R/W-0 | R/W-0 | R-0 | | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| IFCTL_A | Reserved | | CMD_IDLE | TX_SHORT_GAP_EN | Reserved | |
| R/W-0 | R-0 | | R/W-0 | R/W-0 | R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| GIG | TX_PACE | GMII_EN | TX_FLOW_EN | RX_FLOW_EN | MTEST | LOOPBACK | FULLDUPLEX |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-121. CPGMAC_SL1 MAC Control Register (SL1_MACCONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-25 | Reserved | 0 | Reserved |
| 24 | RX_CMF_EN | | RX Copy MAC Control Frames Enable. Enables MAC control frames to be transferred to memory. MAC control frames are normally acted upon (if enabled), but not copied to memory. MAC control frames that are pause frames will be acted upon if enabled in the MACCONTROL register, regardless of the value of RX_CMF_EN. Frames transferred to memory due to RX_CMF_EN will have the **control** bit set in their EOP buffer descriptor. |
| | | 0 | MAC control frames are filtered (but acted upon if enabled). |
| | | 1 | MAC control frames are transferred to memory. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-121. CPGMAC_SL1 MAC Control Register (SL1_MACCONTROL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 23 | RX_CSF_EN | | RX Copy Short Frames Enable. Enables frames or fragments shorter than 64 bytes to be copied to memory. Frames transferred to memory due to RX_CSF_EN will have the **fragment** or **undersized** bit set in their receive footer. Fragments are short frames that contain CRC/align/code errors and undersized are short frames without errors. |
| | | 0 | Short frames are filtered. |
| | | 1 | Short frames are transferred to memory. |
| 22 | RX_CEF_EN | | RX Copy Error Frames Enable. Enables frames containing errors to be transferred to memory. The appropriate error bit will be set in the frame receive footer. Frames containing errors will be filtered when RX_CEF _EN is not set. |
| | | 0 | Frames containing errors are filtered. |
| | | 1 | Frames containing errors are transferred to memory. |
| 21 | TX_SHORT_GAP_LIM_EN | 0-1 | Transmit Short Gap Limit Enable. When set, this bit limits the number of short gap packets transmitted to 100ppm. Each time a short gap packet is sent, a counter is loaded with 10,000 and decremented on each wireside clock. Another short gap packet will not be sent out until the counter decrements to zero. This mode is included to preclude the host from filling up the FIFO and sending every packet out with short gap which would violate the maximum number of packets per second allowed. |
| 20-19 | Reserved | 0 | Reserved |
| 18 | EXT_EN | | Control Enable |
| | | 0 | Use this setting for RMII/GMII mode |
| | | 1 | Use this setting for RGMII mode |
| 17 | GIG_FORCE | 0-1 | Gigabit Mode Force. This bit is used to force the CPGMAC_SL into gigabit mode if the input GMII_MTCLK has been stopped by the PHY. |
| 16 | Reserved | 0 | Reserved |
| 15 | IFCTL_A | | Interface Control A |
| | | 0 | 10 Mbps mode |
| | | 1 | 100 Mbps mode |
| 14-12 | Reserved | 0 | Reserved |
| 11 | CMD_IDLE | | Command Idle. |
| | | 0 | Idle not commanded |
| | | 1 | Idle commanded (read IDLE in MACSTATUS register) |
| 10 | TX_SHORT_GAP_EN | | Transmit Short Gap Enable. |
| | | 0 | Transmit with a short IPG is disabled. |
| | | 1 | Transmit with a short IPG (when **TX_SHORT_GAP** input is asserted) is enabled. |
| 9-8 | Reserved | 0 | Reserved. Read as zero |
| 7 | GIG | | Gigabit Mode |
| | | 0 | 10/100 mode |
| | | 1 | Gigabit mode (full duplex only). The **GIG_OUT** output is the value of this bit. |
| 6 | TX_PACE | | Transmit Pacing Enable |
| | | 0 | Transmit pacing is disabled. |
| | | 1 | Transmit pacing is enabled. |
| 5 | GMII_EN | | G/MII Enable. |
| | | 0 | G/MII RX and TX held in reset. |
| | | 1 | G/MII RX and TX are released from reset. |
| 4 | TX_FLOW_EN | | Transmit Flow Control Enable. Determines if incoming pause frames are acted upon in full-duplex mode. Incoming pause frames are not acted upon in half-duplex mode regardless of this bit setting. The RX_CMF_EN bits determine whether or not received pause frames are transferred to memory. |
| | | 0 | Transmit flow control is disabled. Full-duplex mode: incoming pause frames are not acted upon. |
| | | 1 | Transmit flow control is enabled . Full-duplex mode: incoming pause frames are acted upon. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-121. CPGMAC_SL1 MAC Control Register (SL1_MACCONTROL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 3 | RX_FLOW_EN | | Receive Flow Control Enable. |
| | | 0 | Receive flow control is disabled. Half-duplex mode: no flow control generated collisions are sent. Full-duplex mode: no outgoing pause frames are sent. |
| | | 1 | Receive flow control is enabled. Half-duplex mode: collisions are initiated when receive flow control is triggered. Full-duplex mode: outgoing pause frames are sent when receive flow control is triggered. |
| 2 | MTEST | 0-1 | Manufacturing Test mode. This bit must be set to allow writes to the BOFFTEST and PAUSE registers. |
| 1 | LOOPBACK | | Loop Back Mode. Loopback mode forces internal fullduplex mode regardless of whether the FULLDUPLEX bit is set or not. The LOOPBACK bit should be changed only when GMII_EN is deasserted. |
| | | 0 | Loop back mode is disabled. |
| | | 1 | Loop back mode is enabled. |
| 0 | FULLDUPLEX | | Full Duplex mode. Gigabit mode forces fullduplex mode regardless of whether the FULLDUPLEX bit is set or not. The **FULLDUPLEX_OUT** output is the value of this register bit. |
| | | 0 | Half-duplex mode is enabled. |
| | | 1 | Full-duplex mode is enabled. |

### 9.4.1.98 CPGMAC_SL1 MAC Status Register (SL1_MACSTATUS) (0x708)

**Figure 9-113. CPGMAC_SL1 MAC Status Register (SL1_MACSTATUS)**

| 31 | 30 | | | | | 16 |
|---|---|---|---|---|---|---|
| IDLE | Reserved | | | | | |
| R-1 | R-0 | | | | | |
| 15 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | EXT_GIG | EXT_FULLDUPLEX | Rsvd | RX_FLOW_ACT | TX_FLOW_ACT |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

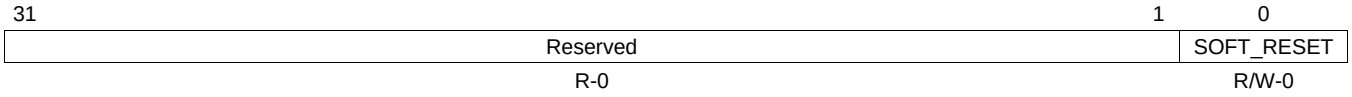
**Table 9-122. CPGMAC_SL1 MAC Status Register (SL1_MACSTATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | IDLE | | CPGMAC_SL1 IDLE. The CPGMAC_SL1 is in the idle state (valid after an idle command). |
| | | 0 | The CPGMAC_SL1 is not in the idle state. |
| | | 1 | The CPGMAC_SL 1 is in the idle state. |
| 30-5 | Reserved | 0 | Reserved. Read as zero |
| 4 | EXT_GIG | 0-1 | External GIG. This is the value of the EXT_GIG input bit. |
| 3 | EXT_FULLDUPLEX | 0-1 | External Fullduplex. This is the value of the EXT_FULLDUPLEX input bit. |
| 2 | Reserved | 0 | Reserved Read as zero |
| 1 | RX_FLOW_ACT | | Receive Flow Control Active. When asserted, indicates that receive flow control is enabled and triggered. |
| | | 0 | Receive flow control is inactive. |
| | | 1 | Receive flow control is active. |
| 0 | TX_FLOW_ACT | | Transmit Flow Control Active. When asserted, this bit indicates that the pause time period is being observed for a received pause frame. No new transmissions will begin while this bit is asserted except for the transmission of pause frames. Any transmission in progress when this bit is asserted will complete. |
| | | 0 | Transmit flow control is inactive. |
| | | 1 | Transmit flow control is active. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.99 CPGMAC_SL1 Software Reset Register (SL1_SOFT_RESET) (0x70C)

**Figure 9-114. CPGMAC_SL1 Software Reset Register (SL1_SOFT_RESET)**



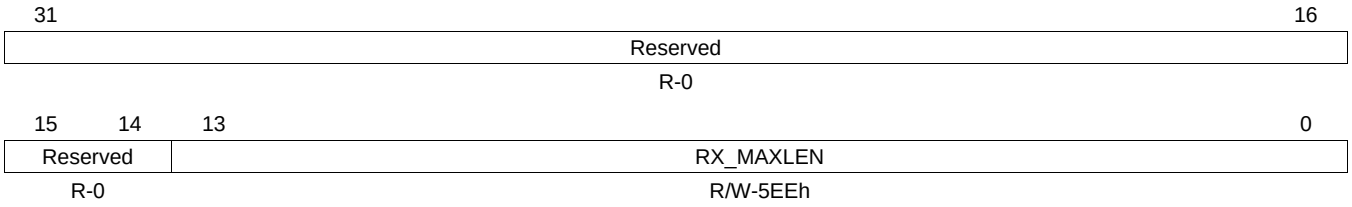

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-123. CPGMAC_SL1 Software Reset Register (SL1_SOFT_RESET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | SOFT_RESET | 0-1 | Software reset. Writing a 1 to this bit causes the CPGMAC_SL1 logic to be reset. After writing a 1 to this bit, it may be polled to determine if the reset has occurred. If a 1 is read, the reset has not yet occurred; if a 0 is read, then reset has occurred. |

### 9.4.1.100 CPGMAC_SL1 RX Maximum Length Register (SL1_RX_MAXLEN) (0x710)

**Figure 9-115. CPGMAC_SL1 RX Maximum Length Register (SL1_RX_MAXLEN)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-124. CPGMAC_SL1 RX Maximum Length Register (SL1_RX_MAXLEN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-14 | Reserved | 0 | Reserved. Read as zero |
| 13-0 | RX_MAXLEN | 0-3FFFh | RX Maximum Frame Length. This field determines the maximum length of a received frame. The reset value is 5EEh (1518 decimal). Frames with byte counts greater than the RX_MAXLEN value are long frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment error are jabber frames. The maximum value is 16,383. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.101 CPGMAC_SL1 Backoff Random Number Generator Test Register (SL1_BOFFTEST) (0x714)

**Figure 9-116. CPGMAC_SL1 Backoff Random Number Generator Test Register (SL1_BOFFTEST)**

| 31 | 30 | 26 | 25 | 16 |
|---|---|---|---|---|
| Rsvd | PACEVAL | | RNDNUM | |
| R-0 | R/W-0 | | R/W-0 | |

| 15 | 12 | 11 | 10 | 9 | 0 |
|---|---|---|---|---|---|
| COLL_COUNT | | Reserved | | TX_BACKOFF | |
| R-0 | | R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-125. CPGMAC_SL1 Backoff Random Number Generator Test Register (SL1_BOFFTEST) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved |
| 30-26 | PACEVAL | 0-1Fh | Pacing Register Current Value. A non-zero value in this field indicates that transmit pacing is active. A transmit frame collision or deferral causes PACEVAL to be loaded with decimal 31, good frame transmissions (with no collisions or deferrals) cause PACEVAL to be decremented down to zero. When PACEVAL is nonzero, the transmitter delays 4 IPGs between new frame transmissions after each successfully transmitted frame that had no deferrals or collisions. Transmit pacing helps reduce capture effects improving overall network bandwidth. |
| 25-16 | RNDNUM | 0-3FFh | Backoff Random Number Generator. This field allows the Backoff Random Number Generator to be read (or written in test mode only). This field can be written only when MTEST in MACCONTROL register has previously been set. Reading this field returns the generator's current value. The value is reset to zero and begins counting on the clock after the deassertion of reset. |
| 15-12 | COLL_COUNT | 0-Fh | Collision Count. The number of collisions the current frame has experienced. |
| 11-10 | Reserved | 0 | Reserved |
| 9-0 | TX_BACKOFF | 0-3FFh | Backoff Count. This field allows the current value of the backoff counter to be observed for test purposes. This field is loaded automatically according to the backoff algorithm, and is decremented by one for each slot time after the collision. |

### 9.4.1.102 CPGMAC_SL1 RX Pause Timer Register (SL1_RX_PAUSE) (0x718)

**Figure 9-117. CPGMAC_SL1 RX Pause Timer Register (SL1_RX_PAUSE)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 0 |
|---|---|
| RX_PAUSETIMER | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

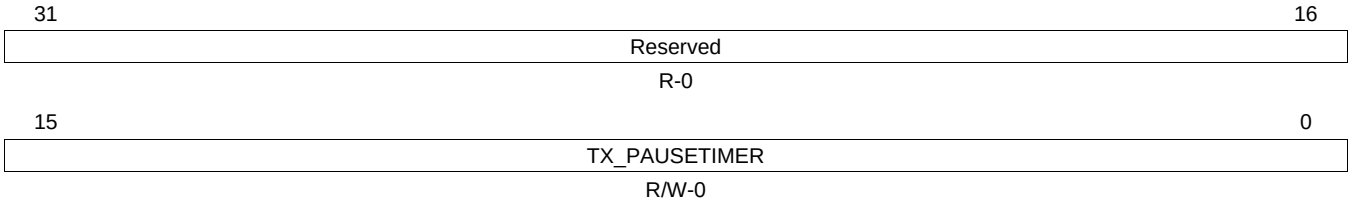
**Table 9-126. CPGMAC_SL1 RX Pause Timer Register (SL1_RX_PAUSE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. Read as zero |
| 15-0 | RX_PAUSETIMER | 0-FFFFh | RX Pause Timer Value. This field allows the contents of the receive pause timer to be observed (and written in test mode). The receive pause timer is loaded with FF00h when the CPGMAC_SL sends an outgoing pause frame (with pause time of FFFFh). The receive pause timer is decremented at slot time intervals. If the receive pause timer decrements to zero, then another outgoing pause frame will be sent and the load/decrement process will be repeated. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.103 CPGMAC_SL1 TX Pause Timer Register (SL1_TX_PAUSE) (0x71C)

**Figure 9-118. CPGMAC_SL1 TX Pause Timer Register (SL1_TX_PAUSE)**



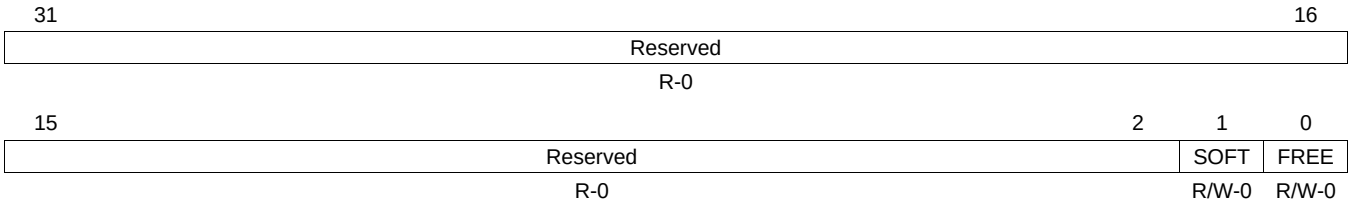

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-127. CPGMAC_SL1 TX Pause Timer Register (SL1_TX_PAUSE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. Read as zero |
| 15-0 | TX_PAUSETIMER | 0-FFFFh | TX Pause Timer Value. This field allows the contents of the transmit pause timer to be observed (and written in test mode). The transmit pause timer is loaded by a received (incoming) pause frame, and then decremented, at slottime intervals, down to zero at which time CPGMAC_SL transmit frames are again enabled. |

### 9.4.1.104 CPGMAC_SL1 Emulation Control Register (SL1_EMCONTROL) (0x720)

**Figure 9-119. CPGMAC_SL1 Emulation Control Register (SL1_EMCONTROL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-128. CPGMAC_SL1 Emulation Control Register (SL1_EMCONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved. Read as zero |
| 1 | SOFT | | Emulation soft bit. This bit is used in conjunction with FREE bit to determine the emulation suspend mode. This bit has no effect if FREE = 1. |
| | | 0 | Soft mode is disabled. EMAC stops immediately during emulation halt. |
| | | 1 | Soft mode is enabled. During emulation halt, EMAC stops after completion of current operation. |
| 0 | FREE | | Emulation free bit. This bit is used in conjunction with SOFT bit to determine the emulation suspend mode. |
| | | 0 | Free-running mode is disabled. During emulation halt, SOFT bit determines operation of EMAC. |
| | | 1 | Free-running mode is enabled. During emulation halt, EMAC continues to operate. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.105 CPGMAC_SL1 RX Packet Priority to Header Priority Mapping Register (SL1_RX_PRI_MAP) (0x724)

**Figure 9-120. CPGMAC_SL1 RX Packet Priority to Header Priority Mapping Register (SL1_RX_Pri_Map)**

| 31 | 30 28 | 27 | 26 24 | 23 | 22 20 | 19 | 18 16 |
|---|---|---|---|---|---|---|---|
| Rsvd | PRI7 | Rsvd | PRI6 | Rsvd | PRI5 | Rsvd | PRI4 |
| R-0 | R/W-7h | R-0 | R/W-6h | R-0 | R/W-5h | R-0 | R/W-4h |

| 15 | 14 12 | 11 | 10 8 | 7 | 6 4 | 3 | 2 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | PRI3 | Rsvd | PRI2 | Rsvd | PRI1 | Rsvd | PRI0 |
| R-0 | R/W-3h | R-0 | R/W-2h | R-0 | R/W-1h | R-0 | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-129. CPGMAC_SL1 RX Packet Priority to Header Priority Mapping Register (SL1_Pri_Map) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved. Read as zero |
| 30-28 | PRI7 | 0-7h | Priority 7. A packet priority of 7 is mapped (changed) to this value. |
| 27 | Reserved | 0 | Reserved. Read as zero |
| 26-24 | PRI6 | 0-7h | Priority 6. A packet priority of 6 is mapped (changed) to this value. |
| 23 | Reserved | 0 | Reserved. Read as zero |
| 22-20 | PRI5 | 0-7h | Priority 5. A packet priority of 5 is mapped (changed) to this value. |
| 19 | Reserved | 0 | Reserved. Read as zero |
| 18-16 | PRI4 | 0-7h | Priority 4. A packet priority of 4 is mapped (changed) to this value. |
| 15 | Reserved | 0 | Reserved. Read as zero |
| 14-12 | PRI3 | 0-7h | Priority 3. A packet priority of 3 is mapped (changed) to this value. |
| 11 | Reserved | 0 | Reserved. Read as zero |
| 10-8 | PRI2 | 0-7h | Priority 2. A packet priority of 2 is mapped (changed) to this value. |
| 7 | Reserved | 0 | Reserved. Read as zero |
| 6-4 | PRI1 | 0-7h | Priority 1. A packet priority of 1 is mapped (changed) to this value. |
| 3 | Reserved | 0 | Reserved. Read as zero |
| 2-0 | PRI0 | 0-7h | Priority 0. A packet priority of 0 is mapped (changed) to this value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.106 CPGMAC_SL2 IDVER Register (SL2_IDVER) (0x740)

**Figure 9-121. CPGMAC_SL2 IDVER Register (SL2_IDVER)**

| 31 | 16 |
|---|---|
| IDENT | |
| R-17h | |

| 15 | 11 | 10 | 8 | 7 | 0 |
|---|---|---|---|---|---|
| Z | | X | | Y | |
| R-0 | | R-1 | | R-17h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 9-130. CPGMAC_SL2 IDVER Register (SL2_IDVER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | IDENT | 0-FFFFh | RX Identification Value |
| 15-11 | Z | 0-1Fh | RX Z value (X.Y.Z) |
| 10-8 | X | 0-7h | RX X value (major) |
| 7-0 | Y | 0-FFh | RX Y value (minor) |

### 9.4.1.107 CPGMAC_SL2 MAC Control Register (SL2_MACCONTROL) (0x744)

**Figure 9-122. CPGMAC_SL2 MAC Control Register (SL2_MACCONTROL)**

| 31 | 25 | 24 |
|---|---|---|
| Reserved | | RX_CMF_EN |
| R-0 | | R/W-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| RX_CSF_EN | RX_CEF_EN | TX_SHORT_GAP_LIM_EN | Reserved | | EXT_EN | GIG_FORCE | Reserved |
| R/W-0 | R/W-0 | R/W-0 | R-0 | | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| IFCTL_A | Reserved | | CMD_IDLE | TX_SHORT_GAP_EN | Reserved | |
| R/W-0 | R-0 | | R/W-0 | R/W-0 | R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| GIG | TX_PACE | GMII_EN | TX_FLOW_EN | RX_FLOW_EN | MTEST | LOOPBACK | FULLDUPLEX |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-131. CPGMAC_SL2 MAC Control Register (SL2_MACCONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-25 | Reserved | 0 | Reserved |
| 24 | RX_CMF_EN | | RX Copy MAC Control Frames Enable. Enables MAC control frames to be transferred to memory. MAC control frames are normally acted upon (if enabled), but not copied to memory. MAC control frames that are pause frames will be acted upon if enabled in the MACCONTROL register, regardless of the value of RX_CMF_EN. Frames transferred to memory due to RX_CMF_EN will have the **control** bit set in their EOP buffer descriptor. |
| | | 0 | MAC control frames are filtered (but acted upon if enabled). |
| | | 1 | MAC control frames are transferred to memory. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-131. CPGMAC_SL2 MAC Control Register (SL2_MACCONTROL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 23 | RX_CSF_EN | | RX Copy Short Frames Enable. Enables frames or fragments shorter than 64 bytes to be copied to memory. Frames transferred to memory due to RX_CSF_EN will have the **fragment** or **undersized** bit set in their receive footer. Fragments are short frames that contain CRC/align/code errors and undersized are short frames without errors. |
| | | 0 | Short frames are filtered. |
| | | 1 | Short frames are transferred to memory. |
| 22 | RX_CEF_EN | | RX Copy Error Frames Enable. Enables frames containing errors to be transferred to memory. The appropriate error bit will be set in the frame receive footer. Frames containing errors will be filtered when RX_CEF _EN is not set. |
| | | 0 | Frames containing errors are filtered. |
| | | 1 | Frames containing errors are transferred to memory. |
| 21 | TX_SHORT_GAP_LIM_EN | 0-1 | Transmit Short Gap Limit Enable. When set, this bit limits the number of short gap packets transmitted to 100ppm. Each time a short gap packet is sent, a counter is loaded with 10,000 and decremented on each wireside clock. Another short gap packet will not be sent out until the counter decrements to zero. This mode is included to preclude the host from filling up the FIFO and sending every packet out with short gap which would violate the maximum number of packets per second allowed. |
| 20-19 | Reserved | 0 | Reserved |
| 18 | EXT_EN | | Control Enable |
| | | 0 | Use this setting for RMII/GMII mode |
| | | 1 | Use this setting for RGMII mode |
| 17 | GIG_FORCE | 0-1 | Gigabit Mode Force. This bit is used to force the CPGMAC_SL into gigabit mode if the input GMII_MTCLK has been stopped by the PHY. |
| 16 | Reserved | 0 | Reserved |
| 15 | IFCTL_A | | Interface Control A |
| | | 0 | 10 Mbps mode |
| | | 1 | 100 Mbps mode |
| 14-12 | Reserved | 0 | Reserved |
| 11 | CMD_IDLE | | Command Idle. |
| | | 0 | Idle not commanded |
| | | 1 | Idle commanded (read IDLE in MACSTATUS register) |
| 10 | TX_SHORT_GAP_EN | | Transmit Short Gap Enable. |
| | | 0 | Transmit with a short IPG is disabled. |
| | | 1 | Transmit with a short IPG (when **TX_SHORT_GAP** input is asserted) is enabled. |
| 9-8 | Reserved | 0 | Reserved. Read as zero |
| 7 | GIG | | Gigabit Mode |
| | | 0 | 10/100 mode |
| | | 1 | Gigabit mode (full duplex only). The **GIG_OUT** output is the value of this bit. |
| 6 | TX_PACE | | Transmit Pacing Enable |
| | | 0 | Transmit pacing is disabled. |
| | | 1 | Transmit pacing is enabled. |
| 5 | GMII_EN | | G/MII Enable. |
| | | 0 | G/MII RX and TX held in reset. |
| | | 1 | G/MII RX and TX are released from reset. |
| 4 | TX_FLOW_EN | | Transmit Flow Control Enable. Determines if incoming pause frames are acted upon in full-duplex mode. Incoming pause frames are not acted upon in half-duplex mode regardless of this bit setting. The RX_CMF_EN bits determine whether or not received pause frames are transferred to memory. |
| | | 0 | Transmit flow control is disabled. Full-duplex mode: incoming pause frames are not acted upon. |
| | | 1 | Transmit flow control is enabled . Full-duplex mode: incoming pause frames are acted upon. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 9-131. CPGMAC_SL2 MAC Control Register (SL2_MACCONTROL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 3 | RX_FLOW_EN | | Receive Flow Control Enable. |
| | | 0 | Receive flow control is disabled. Half-duplex mode: no flow control generated collisions are sent. Full-duplex mode: no outgoing pause frames are sent. |
| | | 1 | Receive flow control is enabled. Half-duplex mode: collisions are initiated when receive flow control is triggered. Full-duplex mode: outgoing pause frames are sent when receive flow control is triggered. |
| 2 | MTEST | 0-1 | Manufacturing Test mode. This bit must be set to allow writes to the BOFFTEST and PAUSE registers. |
| 1 | LOOPBACK | | Loop Back Mode. Loopback mode forces internal fullduplex mode regardless of whether the FULLDUPLEX bit is set or not. The LOOPBACK bit should be changed only when GMII_EN is deasserted. |
| | | 0 | Loop back mode is disabled. |
| | | 1 | Loop back mode is enabled. |
| 0 | FULLDUPLEX | | Full Duplex mode. Gigabit mode forces fullduplex mode regardless of whether the FULLDUPLEX bit is set or not. The **FULLDUPLEX_OUT** output is the value of this register bit. |
| | | 0 | Half-duplex mode is enabled. |
| | | 1 | Full-duplex mode is enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.108 CPGMAC_SL2 MAC Status Register (SL2_MACSTATUS) (0x748)

**Figure 9-123. CPGMAC_SL2 MAC Status Register (SL2_MACSTATUS)**

| 31 | 30 | 16 |
|---|---|---|
| IDLE | Reserved | |
| R-1 | R-0 | |

| 15 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | EXT_GIG | EXT_FULLDUPLEX | Rsvd | RX_FLOW_ACT | TX_FLOW_ACT |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 9-132. CPGMAC_SL2 MAC Status Register (SL2_MACSTATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | IDLE | | CPGMAC_SL2 IDLE. The CPGMAC_SL2 is in the idle state (valid after an idle command). |
| | | 0 | The CPGMAC_SL2 is not in the idle state. |
| | | 1 | The CPGMAC_SL 2 is in the idle state. |
| 30-5 | Reserved | 0 | Reserved. Read as zero |
| 4 | EXT_GIG | 0-1 | External GIG. This is the value of the EXT_GIG input bit. |
| 3 | EXT_FULLDUPLEX | 0-1 | External Fullduplex. This is the value of the EXT_FULLDUPLEX input bit. |
| 2 | Reserved | 0 | Reserved Read as zero |
| 1 | RX_FLOW_ACT | | Receive Flow Control Active. When asserted, indicates that receive flow control is enabled and triggered. |
| | | 0 | Receive flow control is inactive. |
| | | 1 | Receive flow control is active. |
| 0 | TX_FLOW_ACT | | Transmit Flow Control Active. When asserted, this bit indicates that the pause time period is being observed for a received pause frame. No new transmissions will begin while this bit is asserted except for the transmission of pause frames. Any transmission in progress when this bit is asserted will complete. |
| | | 0 | Transmit flow control is inactive. |
| | | 1 | Transmit flow control is active. |

### 9.4.1.109 CPGMAC_SL2 Software Reset Register (SL2_SOFT_RESET) (0x74C)

**Figure 9-124. CPGMAC_SL2 Software Reset Register (SL2_SOFT_RESET)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | SOFT_RESET |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-133. CPGMAC_SL2 Software Reset Register (SL2_SOFT_RESET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | SOFT_RESET | 0-1 | Software reset. Writing a 1 to this bit causes the CPGMAC_SL2 logic to be reset. After writing a 1 to this bit, it may be polled to determine if the reset has occurred. If a 1 is read, the reset has not yet occurred; if a 0 is read, then reset has occurred. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.110 CPGMAC_SL2 RX Maximum Length Register (SL2_RX_MAXLEN) (0x750)

**Figure 9-125. CPGMAC_SL2 RX Maximum Length Register (SL2_RX_MAXLEN)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 0 |
|---|---|---|---|
| Reserved | | RX_MAXLEN | |
| R-0 | | R/W-5EEh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-134. CPGMAC_SL2 RX Maximum Length Register (SL2_RX_MAXLEN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-14 | Reserved | 0 | Reserved. Read as zero |
| 13-0 | RX_MAXLEN | 0-3FFFh | RX Maximum Frame Length. This field determines the maximum length of a received frame. The reset value is 5EEh (1518 decimal). Frames with byte counts greater than the RX_MAXLEN value are long frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment error are jabber frames. The maximum value is 16,383. |

### 9.4.1.111 CPGMAC_SL2 Backoff Random Number Generator Test Register (SL2_BOFFTEST) (0x754)

**Figure 9-126. CPGMAC_SL2 Backoff Random Number Generator Test Register (SL2_BOFFTEST)**

| 31 | 30 | 26 | 25 | 16 |
|---|---|---|---|---|
| Rsvd | PACEVAL | | RNDNUM | |
| R-0 | R/W-0 | | R/W-0 | |

| 15 | 12 | 11 | 10 | 9 | 0 |
|---|---|---|---|---|---|
| COLL_COUNT | | Reserved | | TX_BACKOFF | |
| R-0 | | R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-135. CPGMAC_SL2 Backoff Random Number Generator Test Register (SL2_BOFFTEST) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved. Read as zero |
| 30-26 | PACEVAL | 0-1Fh | Pacing Register Current Value. A non-zero value in this field indicates that transmit pacing is active. A transmit frame collision or deferral causes PACEVAL to be loaded with decimal 31, good frame transmissions (with no collisions or deferrals) cause PACEVAL to be decremented down to zero. When PACEVAL is nonzero, the transmitter delays 4 IPGs between new frame transmissions after each successfully transmitted frame that had no deferrals or collisions. Transmit pacing helps reduce capture effects improving overall network bandwidth. |
| 25-16 | RNDNUM | 0-3FFh | Backoff Random Number Generator. This field allows the Backoff Random Number Generator to be read (or written in test mode only). This field can be written only when MTEST in MACCONTROL register has previously been set. Reading this field returns the generator's current value. The value is reset to zero and begins counting on the clock after the deassertion of reset. |
| 15-12 | COLL_COUNT | 0-Fh | Collision Count. The number of collisions the current frame has experienced. |
| 11-10 | Reserved | 0 | Reserved. Read as zero |
| 9-0 | TX_BACKOFF | 0-3FFh | Backoff Count. This field allows the current value of the backoff counter to be observed for test purposes. This field is loaded automatically according to the backoff algorithm, and is decremented by one for each slot time after the collision. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.112 CPGMAC_SL2 RX Pause Timer Register (SL2_RX_PAUSE) (0x758)

**Figure 9-127. CPGMAC_SL2 RX Pause Timer Register (SL2_RX_PAUSE)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |
| 15 | 0 |
| RX_PAUSETIMER | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-136. CPGMAC_SL2 RX Pause Timer Register (SL2_RX_PAUSE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. Read as zero |
| 15-0 | RX_PAUSETIMER | 0-FFFFh | RX Pause Timer Value. This field allows the contents of the receive pause timer to be observed (and written in test mode). The receive pause timer is loaded with FF00h when the CPGMAC_SL sends an outgoing pause frame (with pause time of FFFFh). The receive pause timer is decremented at slot time intervals. If the receive pause timer decrements to zero, then another outgoing pause frame will be sent and the load/decrement process will be repeated. |

### 9.4.1.113 CPGMAC_SL2 TX Pause Timer Register (SL2_TX_PAUSE) (0x75C)

**Figure 9-128. CPGMAC_SL2 TX Pause Timer Register (SL2_TX_PAUSE)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |
| 15 | 0 |
| TX_PAUSETIMER | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-137. CPGMAC_SL2 TX Pause Timer Register (SL2_TX_PAUSE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. Read as zero |
| 15-0 | TX_PAUSETIMER | 0-FFFFh | TX Pause Timer Value. This field allows the contents of the transmit pause timer to be observed (and written in test mode). The transmit pause timer is loaded by a received (incoming) pause frame, and then decremented, at slottime intervals, down to zero at which time CPGMAC_SL transmit frames are again enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.114 CPGMAC_SL2 Emulation Control Register (SL2_EMCONTROL) (0x760)

**Figure 9-129. CPGMAC_SL2 Emulation Control Register (SL2_EMCONTROL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | SOFT | FREE |
| R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-138. CPGMAC_SL2 Emulation Control Register (SL2_EMCONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved. Read as zero |
| 1 | SOFT | | Emulation soft bit. This bit is used in conjunction with FREE bit to determine the emulation suspend mode. This bit has no effect if FREE = 1. |
| | | 0 | Soft mode is disabled. EMAC stops immediately during emulation halt. |
| | | 1 | Soft mode is enabled. During emulation halt, EMAC stops after completion of current operation. |
| 0 | FREE | | Emulation free bit. This bit is used in conjunction with SOFT bit to determine the emulation suspend mode. |
| | | 0 | Free-running mode is disabled. During emulation halt, SOFT bit determines operation of EMAC. |
| | | 1 | Free-running mode is enabled. During emulation halt, EMAC continues to operate. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.1.115 CPGMAC_SL2 RX Packet Priority to Header Priority Mapping Register (SL2_RX_PRI_MAP) (0x764)

**Figure 9-130. CPGMAC_SL2 RX Packet Priority to Header Priority Mapping Register (SL2_RX_PRI_MAP)**

| 31 | 30 28 | 27 | 26 24 | 23 | 22 20 | 19 | 18 16 |
|---|---|---|---|---|---|---|---|
| Rsvd | PRI7 | Rsvd | PRI6 | Rsvd | PRI5 | Rsvd | PRI4 |
| R-0 | R/W-7h | R-0 | R/W-6h | R-0 | R/W-5h | R-0 | R/W-4h |

| 15 | 14 12 | 11 | 10 8 | 7 | 6 4 | 3 | 2 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | PRI3 | Rsvd | PRI2 | Rsvd | PRI1 | Rsvd | PRI0 |
| R-0 | R/W-3h | R-0 | R/W-2h | R-0 | R/W-1h | R-0 | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-139. CPGMAC_SL2 RX Packet Priority to Header Priority Mapping Register (SL2_RX_PRI_MAP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved. Read as zero |
| 30-28 | PRI7 | 0-7h | Priority 7. A packet priority of 7 is mapped (changed) to this value. |
| 27 | Reserved | 0 | Reserved. Read as zero |
| 26-24 | PRI6 | 0-7h | Priority 6. A packet priority of 6 is mapped (changed) to this value. |
| 23 | Reserved | 0 | Reserved. Read as zero |
| 22-20 | PRI5 | 0-7h | Priority 5. A packet priority of 5 is mapped (changed) to this value. |
| 19 | Reserved | 0 | Reserved. Read as zero |
| 18-16 | PRI4 | 0-7h | Priority 4. A packet priority of 4 is mapped (changed) to this value. |
| 15 | Reserved | 0 | Reserved. Read as zero |
| 14-12 | PRI3 | 0-7h | Priority 3. A packet priority of 3 is mapped (changed) to this value. |
| 11 | Reserved | 0 | Reserved. Read as zero |
| 10-8 | PRI2 | 0-7h | Priority 2. A packet priority of 2 is mapped (changed) to this value. |
| 7 | Reserved | 0 | Reserved. Read as zero |
| 6-4 | PRI1 | 0-7h | Priority 1. A packet priority of 1 is mapped (changed) to this value. |
| 3 | Reserved | 0 | Reserved. Read as zero |
| 2-0 | PRI0 | 0-7h | Priority 0. A packet priority of 0 is mapped (changed) to this value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.2 MDIO Module

This section describes the memory-mapped registers for the MDIO module. The base address of the MDIO module registers is 4A10 0800h. Table 9-140 lists the memory-mapped registers for the MDIO module.

**Table 9-140. MDIO Module Registers**

| Offset | Acronym | Register Name |
|---|---|---|
| 800h | VERSION | MDIO version register |
| 804h | CONTROL | MDIO control register |
| 808h | ALIVE | PHY alive status register |
| 80Ch | LINK | PHY link status register |
| 810h | LINKINTRAW | MDIO link status change interrupt register (raw value) |
| 814h | LINKINTMASKED | MDIO link status change interrupt register (masked value) |
| 820h | USERINTRAW | MDIO user command complete interrupt register (raw value) |
| 824h | USERINTMASKED | MDIO user command complete interrupt register (masked value) |
| 828h | USERINTMASKSET | MDIO user interrupt mask set register |
| 82Ch | USERINTMASKCLEAR | MDIO user interrupt mask clear register |
| 880h | USERACCESS0 | MDIO user access register 0 |
| 884h | USERPHYSEL0 | MDIO user PHY select register 0 |
| 888h | USERACCESS1 | MDIO user access register 1 |
| 88Ch | USERPHYSEL1 | MDIO user PHY select register 1 |

#### 9.4.2.1 MDIO Version Register (VERSION) (800h)

**Figure 9-131. MDIO Version Register (VERSION)**

| 31 16 | 15 8 | 7 0 |
|---|---|---|
| MODID | REVMAJ | REVMIN |
| R-7h | R-1 | R-5h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-141. MDIO Version Register (VERSION) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | MODID | 0-FFFFh | Module ID |
| 15-8 | REVMAJ | 0-FFh | VBUS Management Interface module major revision value |
| 7-0 | REVMIN | 0-FFh | VBUS Management Interface module minor revision value |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.2.2 MDIO Control Register (CONTROL) (804h)

**Figure 9-132. MDIO Control (CONTROL) Register**

| 31 | 30 | 29 | 28 24 | 23 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|
| IDLE | ENABLE | Rsvd | HIGHEST_USER_CHANNEL | Reserved | PREAMBLE | FAULT | FAULT DETECT ENABLE | INT TEST ENABLE | Rsvd |
| R-1 | R/W-0 | R-0 | R-1 | R-0 | R/W-0 | R/WC-0 | R/W-0 | R/W-0 | R-0 |

| 15 0 |
|---|
| CLKDIV |
| R/W |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-142. MDIO Control Register (CONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | IDLE | | MDIO state machine IDLE. |
| | | 0 | State machine is not in idle state. |
| | | 1 | State machine is in idle state. |
| 30 | ENABLE | | Enable control. If the MDIO state machine is active at the time it is disabled, it will complete the current operation before halting and setting the idle bit. If using byte access, the enable bit has to be the last bit written in this register. |
| | | 0 | Disables the MDIO state machine. |
| | | 1 | Enable the MDIO state machine. |
| 29 | Reserved | 0 | Reserved |
| 28-24 | HIGHEST_USER_ CHANNEL | 0-1Fh | Highest user channel. This field specifies the highest useraccess channel that is available in the module and is currently set to 1. This implies that USERACCESS1 is the highest available user access channel. |
| 23-21 | Reserved | 0 | Reserved |
| 20 | PREAMBLE | | Preamble disable. |
| | | 0 | Standard MDIO preamble is used. |
| | | 1 | Disables this device from sending MDIO frame preambles. |
| 19 | FAULT | | Fault indicator. This bit is set to 1 if the MDIO pins fail to read back what the device is driving onto them. This indicates a physical layer fault and the module state machine is reset. Writing a 1 to it clears this bit. |
| | | 0 | No failure |
| | | 1 | Physical layer fault; the MDIO state machine is reset. |
| 18 | FAULT_DETECT_ ENABLE | | Fault detect enable. This bit has to be set to 1 to enable the physical layer fault detection. |
| | | 0 | Disables the physical layer fault detection. |
| | | 1 | Enables the physical layer fault detection. |
| 17 | INT_TEST_ENABLE | | Interrupt test enable. This bit can be set to 1 to enable the host to set the USERINT and LINKINT bits for test purposes. |
| 16 | Reserved | 0 | Reserved |
| 15-0 | CLKDIV | 0-FFFFh | Clock Divider. This field specifies the division ratio between CLK and the frequency of MDCLK. MDCLK is disabled when CLKDIV is cleared to 0. MDCLK frequency = clk frequency/(CLKDIV+1). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.2.3 PHY Alive Status Register (ALIVE) (0x808)

**Figure 9-133. PHY Alive Status Register (ALIVE)**

| 31 | 0 |
|---|---|
| ALIVE | |
| R/WC-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-143. PHY Alive Status Register (ALIVE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ALIVE | | MDIO Alive. Each of the 32 bits of this register is set if the most recent access to the PHY with address corresponding to the register bit number was acknowledged by the PHY, the bit is reset if the PHY fails to acknowledge the access. Both the user and polling accesses to a PHY will cause the corresponding alive bit to be updated. The alive bits are only meant to be used to give an indication of the presence or not of a PHY with the corresponding address. |
| | | 0 | No effect |
| | | 1 | Clears the bit |

### 9.4.2.4 PHY Link Status Register (LINK) (0x80C)

**Figure 9-134. PHY Link Status Register (LINK)**

| 31 | 0 |
|---|---|
| LINK | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-144. PHY Link Status Register (LINK) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | LINK | 0-FFFF FFFFh | MDIO Link state. This register is updated after a read of the Generic Status Register of a PHY. The bit is set if the PHY with the corresponding address has link and the PHY acknowledges the read transaction. The bit is reset if the PHY indicates it does not have link or fails to acknowledge the read transaction. Writes to the register have no effect. In addition, the status of the two PHYs specified in the USERPHYSEL registers can be determined using the MLINK input pins. This is determined by the LINKSEL bit in the USERPHYSEL register. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.2.5 MDIO Link Status Change Interrupt (raw value) Register (LINKINTRAW) (0x810)

**Figure 9-135. MDIO Link Status Change Interrupt (raw value) Register (LINKINTRAW)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | LINKINTRAW | |
| R-0 | | R/WC-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-145. MDIO Link Status Change Interrupt (raw value) Register (LINKINTRAW) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1-0 | LINKINTRAW | | MDIO link change event, raw value. When asserted '1', a bit indicates that there was an MDIO link change event (that is, change in the MDIO link register) corresponding to the PHY address in the USERPHYSEL register. LINKINTRAW[0] and LINKINTRAW[1] correspond to USERPHYSEL0 and USERPHYSEL1, respectively. If the INT_TEST_ENABLE bit in the MDIO control register is set, the host may set the LINKINTRAW bits to a '1'. This mode may be used for test purposes. |
| | | 0 | No effect |
| | | 1 | Clears the event |

### 9.4.2.6 MDIO Link Status Change Interrupt (masked value) Register (LINKINTMASKED) (0x814)

**Figure 9-136. MDIO Link Status Change Interrupt (masked value) Register (LINKINTMASKED)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | LINKINTMASKED | |
| R-0 | | R/WC-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-146. MDIO Link Status Change Interrupt (masked value) Register (LINKINTMASKED) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | | Reserved |
| 1-0 | LINKINTMASKED | | MDIO link change interrupt, masked value. When asserted '1', a bit indicates that there was an MDIO link change event (that is, change in the MDIO link register) corresponding to the PHY address in the USERPHYSEL register and the corresponding LINKINT_ENABLE bit was set. LINKINTMASKED[0] and LINKINTMASKED[1] correspond to USERPHYSEL0 and USERPHYSEL1, respectively. If the INT_TEST_ENABLE bit in the MDIO control register is set, the host may set the LINKINT bits to a '1'. This mode may be used for test purposes. |
| | | 0 | No effect |
| | | 1 | Clears the interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.2.7 MDIO User Command Complete Interrupt (raw value) Register (USERINTRAW) (0x820)

**Figure 9-137. MDIO User Command Complete Interrupt (raw value) Register (USERINTRAW)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | USERINTRAW | |
| R-0 | | R/WC-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-147. MDIO User Command Complete Interrupt (raw value) Register (USERINTRAW) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1-0 | USERINTRAW | | Raw value of MDIO user command complete event for USERACCESS1 and MDIOUSERACCESS0, respectively. When asserted '1', a bit indicates that the previously scheduled PHY read or write command using that particular USERACCESS register has completed. Writing a '1' will clear the event and writing '0' has no effect. If the INT_TEST_ENABLE bit in the MDIO control register is set, the host may set the USERINTRAW bits to a '1'. This mode may be used for test purposes. |

### 9.4.2.8 MDIO User Command Complete Interrupt (masked value) Register (USERINTMASKED) (0x824)

**Figure 9-138. MDIO User Command Complete Interrupt (masked value) Register (USERINTMASKED)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | USERINTMASKED | |
| R-0 | | R/WC-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-148. MDIO User Command Complete Interrupt (masked value) Register (USERINTMASKED) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1-0 | USERINTMASKED | | Masked value of MDIO user command complete interrupt for USERACCESS1 and USERACCESS0, respectively. When asserted '1', a bit indicates that the previously scheduled PHY read or write command using that particular USERACCESS register has completed and the corresponding userintmaskset bit is set to '1'. If the INT_TEST_ENABLE bit in the MDIO control register is set, the host may set the USERINTMASKED bits to a '1'. This mode may be used for test purposes. |
| | | 0 | No effect |
| | | 1 | Clears the interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.2.9 MDIO User Command Complete Interrupt Mask Set Register (USERINTMASKSET) (0x828)

**Figure 9-139. MDIO User Command Complete Interrupt Mask Set Register (USERINTMASKSET)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | USERINTMASKSET | |
| R-0 | | R/WS-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-149. MDIO User Command Complete Interrupt Mask Set Register (USERINTMASKSET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1-0 | USERINTMASKSET | | MDIO user interrupt mask set for USERINTMASKED[1:0], respectively. MDIO user interrupt for a particular USERACCESS register is disabled if the corresponding bit is cleared to 0. |
| | | 0 | No effect |
| | | 1 | Enable MDIO user command complete interrupts for that particular USERACCESS register. |

### 9.4.2.10 MDIO User Command Complete Interrupt Mask Clear Register (USERINTMASKCLEAR) (0x82C)

**Figure 9-140. MDIO User Command Complete Interrupt Mask Clear Register (USERINTMASKCLEAR)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | USERINTMASKCLR | |
| R-0 | | R/WC-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-150. MDIO User Command Complete Interrupt Mask Clear Register (USERINTMASKCLEAR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1-0 | USERINTMASKCLR | | MDIO user command complete interrupt mask clear for USERINTMASKED[1:0], respectively. . |
| | | 0 | No effect |
| | | 1 | Disables further user command complete interrupts for that particular MDIOUserAccess register |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.2.11 MDIO User Access Register 0 (USERACCESS0) (0x880)

**Figure 9-141. MDIO User Access Register 0 (USERACCESS0)**

| 31 | 30 | 29 | 28 26 | 25 21 | 20 16 |
|---|---|---|---|---|---|
| GO | WRITE | ACK | Reserved | REGADR | PHYADR |
| R/WS-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 |

| 15 0 |
|---|
| DATA |
| R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-151. MDIO User Access Register 0 (USERACCESS0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | GO | 0-1 | Go. Writing a 1 to this bit causes the MDIO state machine to perform an MDIO access when it is convenient for it to do so, this is not an instantaneous process. Writing a 0 to this bit has no effect. This bit is write able only if the MDIO state machine is enabled. This bit will self clear when the requested access has been completed. Any writes to the MDIOUserAccess0 register are blocked when the go bit is '1'. If byte access is being used, the go bit should be written last. |
| 30 | WRITE | | Write enable. |
| | | 0 | The MDIO transaction is to be a register read. |
| | | 1 | The MDIO transaction is to be a register write. |
| 29 | ACK | 0-1 | Acknowledge. This bit is set if the PHY acknowledged the read transaction. |
| 28-26 | Reserved | 0 | Reserved. Writes to these bits have no effect. They always read as 0. |
| 25-21 | REGADR | 0-1Fh | Register address. This field specifies the PHY register to be accessed for this transaction. |
| 20-16 | PHYADR | 0-1Fh | PHY address. This field specifies the PHY to be accessed for this transaction. |
| 15-0 | DATA | 0-FFFFh | User data. The data value read from or to be written to the specified PHY register. |

### 9.4.2.12 MDIO User PHY Select Register 0 (USERPHYSEL0) (0x884)

**Figure 9-142. MDIO User PHY Select Register 0 (USERPHYSEL0)**

| 31 16 |
|---|
| Reserved |
| R-0 |

| 15 8 | 7 | 6 | 5 | 4 0 |
|---|---|---|---|---|
| Reserved | LINKSEL | LINKINT_ENABLE | Reserved | PHYADR_MON |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-152. MDIO User PHY Select Register 0 (USERPHYSEL0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | LINKSEL | | Link status determination select. |
| | | 0 | The link status is determined by the MDIO state machine. |
| | | 1 | The link status is determined using the MLINK pin. |
| 6 | LINKINT_ENABLE | | Link change interrupt enable. |
| | | 0 | Link change interrupts are disabled. |
| | | 1 | Enable link change status interrupts for PHY address specified in PHYADR_MON bit. |
| 5 | Reserved | 0 | Reserved |
| 4-0 | PHYADR_MON | 0-1Fh | PHY address whose link status is to be monitored. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.2.13 MDIO User Access Register 1 (USERACCESS1) (0x888)

**Figure 9-143. MDIO User Access Register 1 (USERACCESS1)**

| 31 | 30 | 29 | 28 26 | 25 21 | 20 16 |
|---|---|---|---|---|---|
| GO | WRITE | ACK | Reserved | REGADR | PHYADR |
| R/WS-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 |

| 15 0 |
|---|
| DATA |
| R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-153. MDIO User Access Register 1 (USERACCESS1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | GO | 0-1 | Go. Writing a 1 to this bit causes the MDIO state machine to perform an MDIO access when it is convenient for it to do so, this is not an instantaneous process. Writing a 0 to this bit has no effect. This bit is write able only if the MDIO state machine is enabled. This bit will self clear when the requested access has been completed. Any writes to the MDIOUserAccess0 register are blocked when the go bit is '1'. If byte access is being used, the go bit should be written last. |
| 30 | WRITE | | Write enable. |
| | | 0 | The MDIO transaction is to be a register read. |
| | | 1 | The MDIO transaction is to be a register write. |
| 29 | ACK | 0-1 | Acknowledge. This bit is set if the PHY acknowledged the read transaction. |
| 28-26 | Reserved | 0 | Reserved. Writes to these bits have no effect. They always read as 0. |
| 25-21 | REGADR | 0-1Fh | Register address. This field specifies the PHY register to be accessed for this transaction. |
| 20-16 | PHYADR | 0-1Fh | PHY address. This field specifies the PHY to be accessed for this transaction. |
| 15-0 | DATA | 0-FFFFh | User data. The data value read from or to be written to the specified PHY register. |

### 9.4.2.14 MDIO User PHY Select Register 1 (USERPHYSEL1) (0x88C)

**Figure 9-144. MDIO User PHY Select Register 1 (USERPHYSEL1)**

| 31 16 |
|---|
| Reserved |
| R-0 |

| 15 8 | 7 | 6 | 5 | 4 0 |
|---|---|---|---|---|
| Reserved | LINKSEL | LINKINT_ENABLE | Reserved | PHYADR_MON |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-154. MDIO User PHY Select Register 1 (USERPHYSEL1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | LINKSEL | | Link status determination select. |
| | | 0 | The link status is determined by the MDIO state machine. |
| | | 1 | The link status is determined using the MLINK pin. |
| 6 | LINKINT_ENABLE | | Link change interrupt enable. |
| | | 0 | Link change interrupts are disabled. |
| | | 1 | Enable link change status interrupts for PHY address specified in PHYADR_MON bit. |
| 5 | Reserved | 0 | Reserved |
| 4-0 | PHYADR_MON | 0-1Fh | PHY address whose link status is to be monitored. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.3 CPSW Subsytem Module

This section describes the memory-mapped registers for the CPSW subsytem module (CPSW_3GSS). The base address of the CPSW subsytem module registers is 4A10 0900h. Table 9-155 lists the memory-mapped registers for the CPSW subsytem module.

**Table 9-155. CPSW Subsytem Module Registers**

| Offset | Acronym | Register Name |
|---|---|---|
| 900h | IDVER | Subsystem ID Version Register |
| 904h | SOFT_RESET | Subsystem Software Reset Register |
| 908h | CONTROL | Subsystem Control Register |
| 90Ch | INT_CONTROL | Subsystem Interrupt Control |
| 910h | RX_THRESH_EN | Subsystem Receive Threshold Interrupt Enable Register |
| 914h | RX_EN | Subsystem Receive Interrupt Enable Register |
| 918h | TX_EN | Subsystem Transmit Interrupt Enable Register |
| 91Ch | MISC_EN | Subsystem Miscellaneous Interrupt Enable Register |
| 940h | RX_THRESH_STAT | Subsystem RX Threshold Masked Interrupt Status Register |
| 944h | RX_STAT | Subsystem RX Masked Interrupt Status Register |
| 948h | TX_STAT | Subsystem TX Masked Interrupt Status Register |
| 94Ch | MISC_STAT | Subsystem Miscellaneous Masked Interrupt Status Register |
| 970h | RX_IMAX | Subsystem Receive Interrupts Per Millisecond |
| 974h | TX_IMAX | Subsystem Transmit Interrupts Per Millisecond |
| 988h | RGMII_CTL | RGMII Control Signal Register |

#### 9.4.3.1 Subsystem ID Version Register (IDVER) (0x900)

**Figure 9-145. Subsystem ID Version Register (IDVER)**

| 31–30 | 29–28 | 27–16 |
|---|---|---|
| SCHEME | Reserved | FUNCTION |
| R-1h | R-0 | R-EDAh |

| 15–11 | 10–8 | 7–6 | 5–0 |
|---|---|---|---|
| RTL | MAJOR | CUSTOM | MINOR |
| R-0 | R-1h | R-0 | R-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-156. Subsystem ID Version Register (IDVER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | SCHEME | 0-3h | |
| 29-28 | Reserved | 0 | Read as zero. |
| 27-16 | FUNCTION | 0-FFFh | Function value |
| 15-11 | RTL | 0-1Fh | RTL version |
| 10-8 | MAJOR | 0-7h | Major version |
| 7-6 | CUSTOM | 0-3h | Custom version |
| 5-0 | MINOR | 0-3Fh | Minor version |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.3.2 Subsystem Software Reset Register (SOFT_RESET) (0x904)

**Figure 9-146. Subsystem Software Reset Register (SOFT_RESET)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | SOFT_RESET |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-157. Subsystem Software Reset Register (SOFT_RESET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Read as zero. |
| 0 | SOFT_RESET | 0-1 | Software reset - Writing a one to this bit causes the CPGMACSS_R logic to be reset (INT, REGS, CPPI). Software reset occurs on the clock following the register bit write. |

### 9.4.3.3 Subsystem Control Register (CONTROL) (0x908)

**Figure 9-147. Subsystem Control Register (CONTROL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | MMR_STDBYMODE | | MMR_IDLEMODE | |
| R-0 | | R/W-2h | | R/W-2h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-158. Subsystem Control Register (CONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Read as zero. |
| 3-2 | MMR_STDBYMODE | 0-3h | Standbymode MMR bits. This field is used to cause the CPSW_3GSS_R to go to standby mode (software standby). Smart standby is not supported. |
| 1-0 | MMR_IDLEMODE | 0-3h | Idlemode MMR bits. This field is used to cause the CPSW_3GSS_R to go to idle mode (software idle). Smart idle is not supported. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.3.4 Subsystem Interrupt Control Register (INT_CONTROL) (0x90C)

**Figure 9-148. Subsystem Interrupt Control Register (INT_CONTROL)**

| 31 | 30 | 22 | 21 | 16 |
|---|---|---|---|---|
| INT_TEST | Reserved | | INT_PACE_EN | |
| R/W-0 | R-0 | | R/W-0 | |

| 15 | 12 | 11 | 0 |
|---|---|---|---|
| Reserved | | INT_PRESCALE | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-159. Subsystem Interrupt Control Register (INT_CONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | INT_TEST | 0-1 | Interrupt Test. Test bit to the interrupt pacing blocks |
| 30-22 | Reserved | 0 | Read as zero. |
| 21-16 | INT_PACE_EN | | Interrupt Pacing Enable Bus |
| | | 0 | Enables RX_Pulse Pacing (0 is pacing bypass) |
| | | 1 | Enables TX_Pulse Pacing (0 is pacing bypass) |
| 15-10 | Reserved | 0 | Read as zero. |
| 11-0 | INT_PRESCALE | 0-FFFh | Interrupt Counter Prescaler. The number of **MAIN_CLK** periods in 4 us. |

### 9.4.3.5 Subsystem Receive Threshold Interrupt Enable Register (RX_THRESH_EN) (0x910)

**Figure 9-149. Subsystem Receive Threshold Interrupt Enable Register (RX_THRESH_EN)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RX_THRESH_EN | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

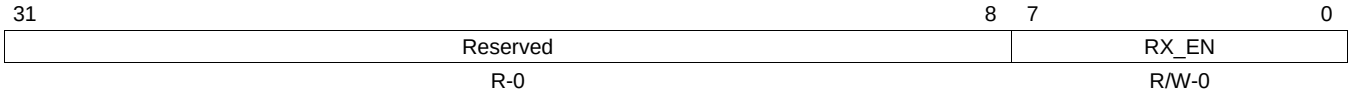
**Table 9-160. Subsystem Receive Threshold Interrupt Enable Register (RX_THRESH_EN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero. |
| 7-0 | RX_THRESH_EN | 0-FFh | Receive Threshold Enable - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled to generate an interrupt on **RX_THRESH_PULSE**. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.3.6 Subsystem Receive Interrupt Enable Register (RX_EN) (0x914)

**Figure 9-150. Subsystem Receive Interrupt Enable Register (RX_EN)**



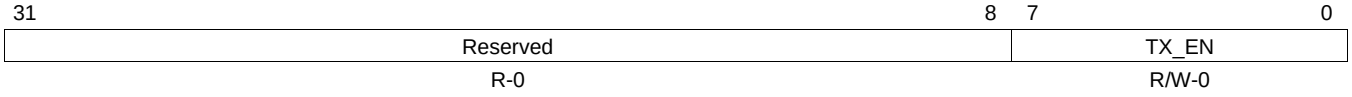

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-161. Subsystem Receive Interrupt Enable Register (RX_EN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero. |
| 7-0 | RX_EN | 0-FFh | Receive Enable. Each bit in this register corresponds to the bit in the RX interrupt that is enabled to generate an interrupt on **RX_PULSE**. |

### 9.4.3.7 Subsystem Transmit Interrupt Enable Register (TX_EN) (0x918)

**Figure 9-151. Subsystem Transmit Interrupt Enable Register (TX_EN)**



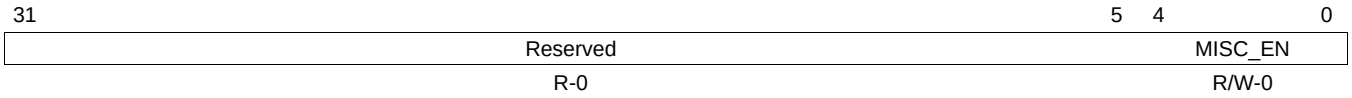

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-162. Subsystem Transmit Interrupt Enable Register (TX_EN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero. |
| 7-0 | TX_EN | 0-FFh | Transmit Enable. Each bit in this register corresponds to the bit in the TX interrupt that is enabled to generate an interrupt on **TX_PULSE**. |

### 9.4.3.8 Subsystem Miscellaneous Interrupt Enable Register (MISC_EN) (0x91C)

**Figure 9-152. Subsystem Miscellaneous Interrupt Enable Register (MISC_EN)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

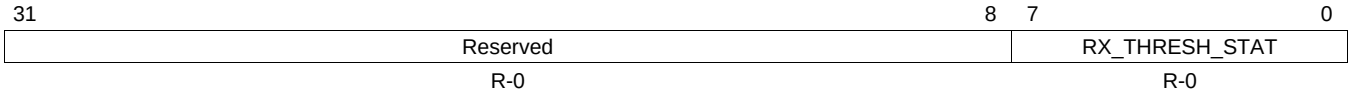
**Table 9-163. Subsystem Miscellaneous Interrupt Enable Register (MISC_EN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Read as zero. |
| 4-0 | MISC_EN | 0-1Fh | Misc Enable. Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled to generate an interrupt on **Misc_PULSE**. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.3.9 Subsystem Receive Threshold Masked Interrupt Status Register (RX_THRESH_STAT) (0x940)

**Figure 9-153. Subsystem Receive Threshold Masked Interrupt Status Register (RX_THRESH_STAT)**



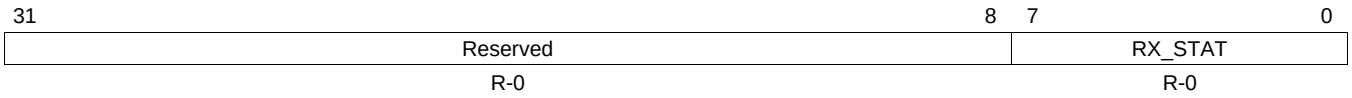

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-164. Subsystem Receive Threshold Masked Interrupt Status Register (RX_THRESH_STAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero. |
| 7-0 | RX_THRESH_STAT | 0-FFh | Receive Threshold Masked Interrupt Status. Each bit in this read only register corresponds to the bit in the receive threshold interrupt that is enabled and generating an interrupt on **RX_THRESH_PULSE**. |

### 9.4.3.10 Subsystem Receive Masked Interrupt Status Register (RX_STAT) (0x944)

**Figure 9-154. Subsystem Receive Masked Interrupt Status Register (RX_STAT)**



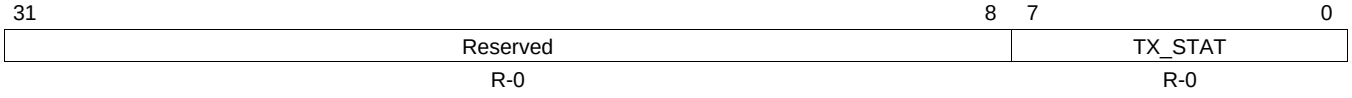

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-165. Subsystem Receive Masked Interrupt Status Register (RX_STAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero. |
| 7-0 | RX_STAT | 0-FFh | Receive Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the RX interrupt that is enabled and generating an interrupt on **RX_PULSE**. |

### 9.4.3.11 Subsystem Transmit Masked Interrupt Status Register (TX_STAT) (0x948)

**Figure 9-155. Subsystem Transmit Masked Interrupt Status Register (TX_STAT)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-166. Subsystem Transmit Masked Interrupt Status Register (TX_STAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero. |
| 7-0 | TX_STAT | 0-FFh | Transmit Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the TX interrupt that is enabled and generating an interrupt on **TX_PULSE**. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.3.12 Subsystem Miscellaneous Masked Interrupt Status Register (MISC_STAT) (0x94C)

**Figure 9-156. Subsystem Miscellaneous Masked Interrupt Status Register (MISC_STAT)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | MISC_STAT | |
| R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-167. Subsystem Miscellaneous Masked Interrupt Status Register (MISC_STAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Read as zero. |
| 4-0 | MISC_STAT | 0-1Fh | Misc Masked Interrupt Status. Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled and generating an interrupt on **MISC_PULSE**. |

### 9.4.3.13 Subsystem Receive Interrupts per Millisecond Register (RX_IMAX) (0x970)

**Figure 9-157. Subsystem Receive Interrupts per Millisecond Register (RX_IMAX)**

| 31 | 6 | 5 | 0 |
|---|---|---|---|
| Reserved | | RX_IMAX | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-168. Subsystem Receive Interrupts per Millisecond Register (RX_IMAX) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Read as zero. |
| 6-0 | RX_IMAX | 0-3Fh | Receive Interrupts per Millisecond. The maximum number of interrupts per millisecond generated on **RX_PULSE** if pacing is enabled for this interrupt. |

### 9.4.3.14 Subsystem Transmit Interrupts per Millisecond Register (TX_IMAX) (0x974)

**Figure 9-158. Subsystem Transmit Interrupts per Millisecond Register (TX_IMAX)**

| 31 | 6 | 5 | 0 |
|---|---|---|---|
| Reserved | | TX_IMAX | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-169. Subsystem Transmit Interrupts per Millisecond Register (TX_IMAX) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Read as zero. |
| 5-0 | TX_IMAX | 0-3Fh | Transmit Interrupts per Millisecond. The maximum number of interrupts per millisecond generated on **TX_PULSE** if pacing is enabled for this interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 9.4.3.15 RGMII Control Read Register (RGMII_CTL) (0x988)

**Figure 9-159. RGMII Control Read Register (RGMII_CTL)**

| 31 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RGMII1_FULLDUPLEX | RGMII1_SPEED | | RGMII1_LINK | RGMII0_FULLDUPLEX | RGMII0_SPEED | | RGMII0_LINK |
| R-0 | R-0 | | R-0 | R-0 | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 9-170. RGMII Control Read Register (RGMII_CTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Read as zero. |
| 7 | RGMII1_FULLDUPLEX | | RGMII 2 Fullduplex. The CPRGMII fullduplex output signal. |
| | | 0 | Half-duplex mode |
| | | 1 | Fullduplex mode |
| 6-5 | RGMII1_SPEED | | RGMII2 Speed. The CPRGMI speed I output signal. |
| | | 0 | 10Mbps mode |
| | | 1h | 100Mbps mode |
| | | 2h | 1000Mbps (gig) mode |
| | | 3h | Reserved |
| 4 | RGMII1_LINK | | RGMII2 Link Indicator. The CPRGMII link output signal. |
| | | 0 | RGMII2 link is down |
| | | 1 | RGMII2 link is up |
| 3 | RGMII0_FULLDUPLEX | | RGMII 1 Fullduplex. The CPRGMII fullduplex output signal. |
| | | 0 | Half-duplex mode |
| | | 1 | Fullduplex mode |
| 2-1 | RGMII0_SPEED | | RGMII2 Speed. The CPRGMII speed output signal. |
| | | 0 | 10Mbps mode |
| | | 1h | 100Mbps mode |
| | | 2h | 1000Mbps (gig) mode |
| | | 3h | Reserved |
| 0 | RGMII0_LINK | | RGMII2 Link Indicator. The CPRGMII link output signal. |
| | | 0 | RGMII2 link is down |
| | | 1 | RGMII2 link is up |


Copyright © 2011–2016, Texas Instruments Incorporated

# General-Purpose I/O (GPIO) Interface

This chapter describes the general-purpose I/O (GPIO) interface.

**Topic** **Page**

**10.1 Introduction** ..... **1606**
**10.2 Architecture** ..... **1608**
**10.3 GPIO Registers** ..... **1615**


Copyright © 2011–2016, Texas Instruments Incorporated

## 10.1 Introduction

### 10.1.1 Purpose of the Peripheral

The general-purpose interface combines two general-purpose input/output (GPIO) banks. Each GPIO module provides 32 dedicated general-purpose pins with input and output capabilities; thus, the general-purpose interface supports up to 64 (2 × 32) pins. These pins can be configured for the following applications:

- Data input (capture)/output (drive)
- Keyboard interface with a debounce cell
- Interrupt generation in active mode upon the detection of external events. Detected events are processed by two parallel independent interrupt-generation submodules to support biprocessor operations.

### 10.1.2 Features

Each channel in the GPIO module has the following features:

- The output enable register (GPIO_OE) controls the output capability for each pin.
- The output line level reflects the value written in the data output register (GPIO_DATAOUT) through the peripheral bus.
- The input line can be fed in to the GPIO module through an optional and configurable debouncing cell.
- The input line value is sampled into the data input register (GPIO_DATAIN) and can be read from the peripheral bus.
- In Active mode, the input line can be used through level and edge detectors to trigger synchronous interrupts. The edge (rising, falling, or both) or the level (logical 0, logical 1, or both) to be used can be configured.

The global features of the GPIO module are:

- Two identical interrupt generation sub-modules process synchronous interrupt requests from each channel in order to be used independently in a bi-processor environment. Each sub-module controls its own synchronous interrupt request line and has its own interrupt enable and interrupt status registers. The interrupt enable register (GPIO_IRQENABLE_SET_x) selects the channel(s) considered for the interrupt request generation, and the interrupt status register (GPIO_IRQSTATUS_RAW_x) determines which channel(s) has/have activated the interrupt request. Event detection on GPIO channels is reflected into the interrupt status registers independently from the interrupt enable registers content.

The module provides an alternative to the atomic ‘Test and Set’ operations for the data output and interrupt enable registers. For these registers, the module implements the “Set and Clear protocol register update”.


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.1.3 Block Diagram

Figure 10-1 details the GPIO block diagram with its configuration registers and its main functional paths:

- The synchronous path (for Active mode operations), Figure 10-2, used to generate a synchronous interrupt request upon expected event detection on any input GPIO; the synchronous interrupt request lines 0 and 1 are active according to their respective interrupt enable 0 and 1 registers.
- The blocks handling the internal clock (clock gating) and managing the Sleep mode request/acknowledge protocol (enabling the Synchronous path in Active mode).

**Figure 10-1. GPIO Block Diagram**

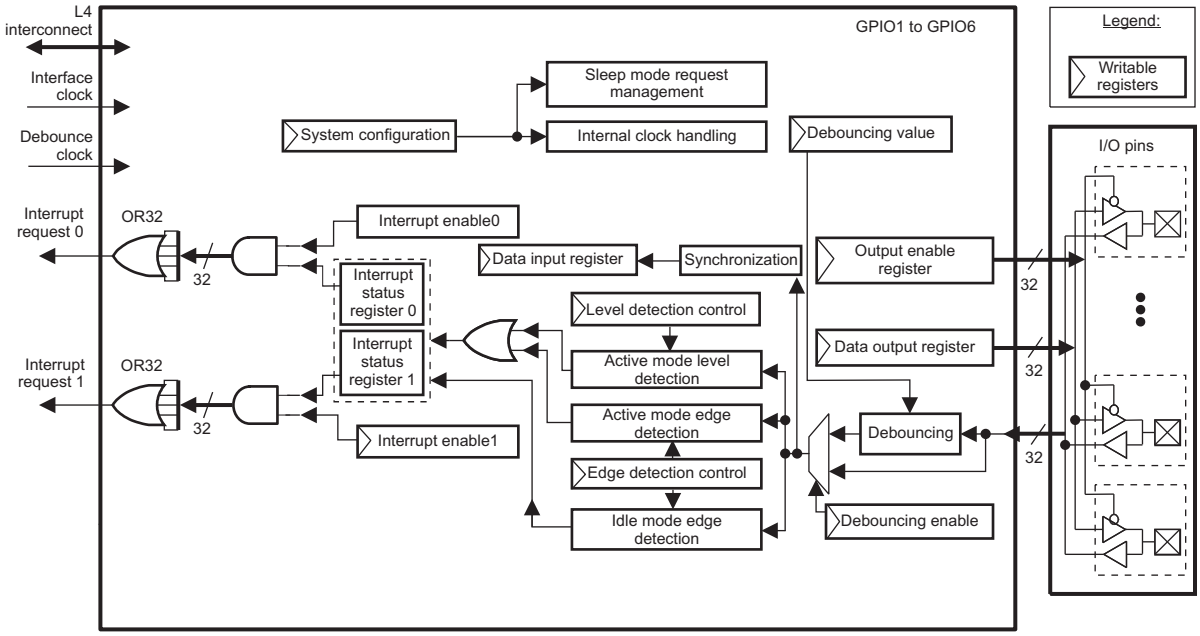
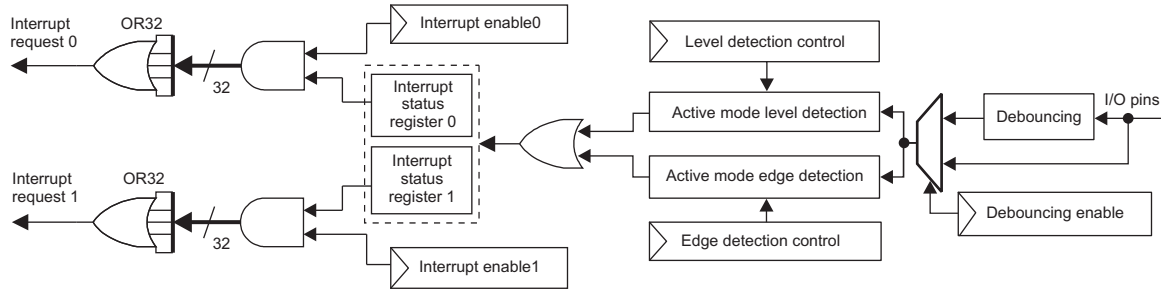



**Figure 10-2. Synchronous Path Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

## 10.2 Architecture

This section discusses the operational details and basic functions of the GPIO peripheral.

### *10.2.1 Operating Modes*

Four operating modes are defined for the module:

- **Active mode:** the module is running synchronously on the interface clock, interrupt can be generated according to the configuration and the external signals.
- **Idle mode:** the module is in a waiting state, interface clock can be stopped , no interrupt can be generated. Check the chip top-level functional specification for the availability of the debouncing clock while in Idle mode.
- **Inactive mode:** the module has no activity, interface clock can be stopped, no interrupt can be generated.
- **Disabled mode:** the module is not used, internal clock paths are gated, no interrupt request can be generated.

The Idle and Inactive modes are configured within the module and activated on request by the host processor through system interface sideband signals. The Disabled mode is set by software through a dedicated configuration bit. It unconditionally gates the internal clock paths not use for the system interface. All module registers are 8, 16 or 32-bit accessible through the OCP compatible interface (little endian encoding). In active mode, the event detection (level or transition) is performed in the GPIO module using the interface clock. The detection's precision is set by the frequency of this clock and the selected internal gating scheme.

### *10.2.2 Clocking and Reset Strategy*

#### 10.2.2.1 Clocks

GPIO module runs using two clocks:

- The debouncing clock is used for the debouncing sub-module logic (without the corresponding configuration registers). This module can sample the input line and filters the input level using a programmed delay.
- The interface clock provided by the peripheral bus (OCP compatible system interface). It is used through the entire GPIO module (except within the debouncing sub-module logic). It clocks the OCP interface and the internal logic. Clock gating features allow adapting the module power consumption to the activity.

#### 10.2.2.2 Clocks, Gating and Active Edge Definitions

The interface clock provided by the peripheral bus (OCP compatible system interface) is used through the entire GPIO module. Two clock domains are defined: the OCP interface and the internal logic. Each clock domain can be controlled independently. Sampling operations for the data capture and for the events detection are done using the rising edge. The data loaded in the data output register (GPIO_DATAOUT) is set at the output GPIO pins synchronously with the rising edge of the interface clock.

Five clock gating features are available:

- Clock for the system interface logic can be gated when the module is not accessed, if the AUTOIDLE configuration bit in the system configuration register (GPIO_SYSCONFIG) is set. Otherwise, this logic is free running on the interface clock.
- Clock for the input data sample logic can be gated when the data in register is not accessed.
- Four clock groups are used for the logic in the synchronous events detection. Each 8 input GPIO_V2 pins group will have a separate enable signal depending on the edge/level detection register setting. If a group requires no detection, then the corresponding clock will be gated. All channels are also gated using a 'one out of N' scheme. N can take the values 1, 2, 4 or 8. The interface clock is enabled for this logic one cycle every N cycles. When N is equal to 1, there is no gating and this logic is free running on the interface clock. When N is between 2 to 8, this logic is running at the equivalent frequency of interface clock frequency divided by N.


Copyright © 2011–2016, Texas Instruments Incorporated

- In Inactive mode, all internal clock paths are gated.
- In Disabled mode, all internal clock paths not used for the system interface are gated. All GPIO registers are accessible synchronously with the interface clock.

#### 10.2.2.3 Sleep Mode Request and Acknowledge

Upon a Sleep mode request issued by the host processor, the GPIO module goes to the Idle mode according to the IDLEMODE field in the system configuration register (GPIO_SYSCONFIG).

- IDLEMODE = 0 (Force-Idle mode): the GPIO goes in Inactive mode independently of the internal module state and the Idle acknowledge is unconditionally sent. In Force-Idle mode, the module is in Inactive mode.
- IDLEMODE = 1h (No-Idle mode): the GPIO does not go to the Idle mode and the Idle acknowledge is never sent.
- IDLEMODE = 2h (Smart-Idle mode) or IDLEMODE = 3h (Smart-Idle mode): the GPIO module evaluates its internal capability to have the interface clock switched off. Once there is no more internal activity (the data input register completed to capture the input GPIO pins, there is no pending interrupt, all interrupt status bits are cleared, and there is no write access to GPIO_DEBOUNCINGTIME register pending to be synchronized), the Idle acknowledge is asserted and the GPIO enters Idle mode. When the system is awake, the Idle Request goes inactive, the Idle acknowledge signals are immediately de-asserted.

**NOTE:** Idle mode request and Idle acknowledge are system interface sideband signals. Once the GPIO acknowledges the Sleep mode request (Idle acknowledge has been sent), the interface clock can be stopped anytime.

Upon a Sleep mode request issued by the host processor, the GPIO module goes to the Idle mode only if there is no active bit in GPIO_IRQSTATUS_RAW_n registers.

#### 10.2.2.4 Reset

The OCP hardware Reset signal has a global reset action on the GPIO. All configuration registers, all DFFs clocked with the Interface clock or Debouncing clock and all internal state machines are reset when the OCP hardware Reset is active (low level). The RESETDONE bit in the system status register (GPIO_SYSSTATUS) monitors the internal reset status: it is set when the Reset is completed on both OCP and Debouncing clock domains. The software Reset (SOFTRESET bit in the system configuration register) has the same effect as the OCP hardware Reset signal, and the RESETDONE bit in GPIO_SYSSTATUS is updated in the same condition.

### *10.2.3 Interrupt Features*

#### 10.2.3.1 Functional Description

In order to generate an interrupt request to a host processor upon a defined event (level or logic transition) occurring on a GPIO pin, the GPIO configuration registers have to be programmed as follows:

- Interrupts for the GPIO channel must be enabled in the GPIO_IRQENABLE_SET_0 and/or GPIO_IRQENABLE_SET_1 registers.
- The expected event(s) on input GPIO to trigger the interrupt request has to be selected in the GPIO_LEVELDETECT0, GPIO_LEVELDETECT1, GPIO_RISINGDETECT, and GPIO_FALLINGDETECT registers.

For instance, interrupt generation on both edges on input k is configured by setting to 1 the *k*th bit in registers GPIO_RISINGDETECT and GPIO_FALLINGDETECT along with the interrupt enabling for one or both interrupt lines (GPIO_IRQENABLE_SET_n).

**NOTE:** All interrupt sources (the 32 input GPIO channels) are merged together to issue two synchronous interrupt requests 0 and 1.


Copyright © 2011–2016, Texas Instruments Incorporated

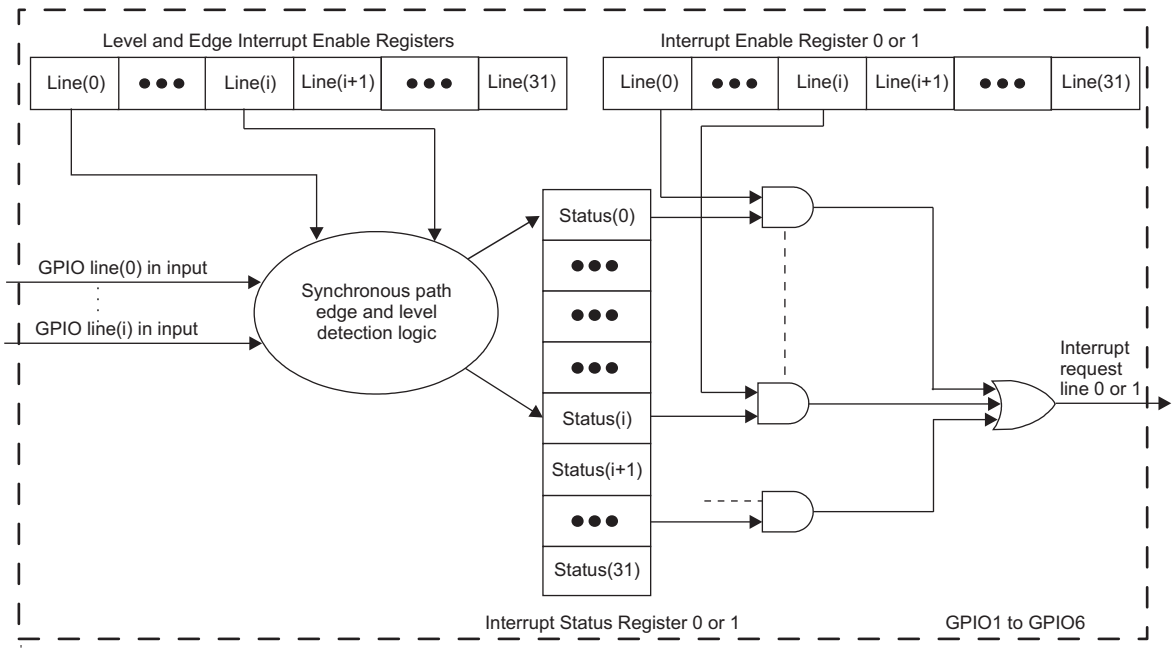
### 10.2.3.2 Synchronous Path: Interrupt Request Generation

In Active mode, once the GPIO configuration registers have been set to enable the interrupt generation, a synchronous path (Figure 10-3) samples the transitions and levels on the input GPIO with the internally gated interface clock. When an event matches the programmed settings, the corresponding bit in the GPIO_IRQSTATUS_RAW_n registers is set to 1 and, on the following interface clock cycle, the interrupt lines 0 and/or 1 are activated (depending on the GPIO_IRQENABLE_SET_n registers).

Due to the sampling operation, the minimum pulse width on the input GPIO to trigger a synchronous interrupt request is two times the internally gated interface clock period (the internally gated interface clock period is equal to N times the interface clock period). This minimum pulse width has to be met before and after any expected level transition detection. Level detection requires the selected level to be stable for at least two times the internally gated interface clock period to trigger a synchronous interrupt.

As the module is synchronous, latency is minimal between the expected event occurrence and the activation of the interrupt line(s). This should not exceed 3 internally gated interface clock cycles + 2 interface clock cycles when the debouncing feature is not used. When the debouncing feature is active, the latency depends on the GPIO_DEBOUNCINGTIME register value and should be less than 3 internally gated interface clock cycles + 2 interface clock cycles + GPIO_DEBOUNCINGTIME value debouncing clock cycles + 3 debouncing clock cycles.

**Figure 10-3. Interrupt Request Generation**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 10.2.3.3 Interrupt Line Release

When the host processor receives an interrupt request issued by the GPIO module, it can read the corresponding GPIO_IRQSTATUS_n register to find out which input GPIO has triggered the interrupt. After servicing the interrupt, the processor resets the status bit and releases the interrupt line by writing a 1 in the corresponding bit of the GPIO_IRQSTATUS_n register. If there is still a pending interrupt request to serve (all bits in the GPIO_IRQSTATUS_RAW_n register not masked by the GPIO_IRQENABLE_SET_n, which are not cleared by setting the GPIO_IRQENABLE_CLR_n), the interrupt line will be re-asserted.

### *10.2.4 General-Purpose Interface Basic Programming Model*

#### 10.2.4.1 Power Saving by Grouping the Edge/Level Detection

Each GPIO module implements four gated clocks used by the edge/level detection logic to save power. Each group of eight input GPIO pins generates a separate enable signal depending on the edge/level detection register setting (because the input is 32 bits, four groups of eight inputs are defined for each GPIO module). If a group requires no edge/level detection, then the corresponding clock is gated (cut off). Grouping the edge/level enable can save the power consumption of the module as described in the following example.

If any of the registers:

- GPIO_LEVELDETECT0
- GPIO_LEVELDETECT1
- GPIO_RISINGDETECT
- GPIO_FALLINGDETECT

are set to 0101 0101h, then all clocks are active (power consumption is high);

are set to 0000 00FFh, then a single clock is active.

**NOTE:** When the clocks are enabled by writing to the GPIO_LEVELDETECT0, GPIO_LEVELDETECT1, GPIO_RISINGDETECT, and GPIO_FALLINGDETECT registers, the detection starts after 5 clock cycles. This period is required to clean the synchronization edge/level detection pipeline.

The mechanism is independent of each clock group. If the clock has been started before a new setting is performed, the following is recommended: first, set the new detection required; second, disable the previous setting (if necessary). In this way, the corresponding clock is not gated and the detection starts immediately.

#### 10.2.4.2 Set and Clear Instructions

The GPIO module implements the set-and-clear protocol register update for the data output and interrupt enable registers. This protocol is an alternative to the atomic test and set operations and consists of writing operations at dedicated addresses (one address for setting bit[s] and one address for clearing bit[s]). The data to write is 1 at bit position(s) to clear (or to set) and 0 at unaffected bit(s).

Registers can be accessed in two ways:

- Standard: Full register read and write operations at the primary register address
- Set and clear (recommended): Separate addresses are provided to set (and clear) bits in registers. Writing 1 at these addresses sets (or clears) the corresponding bit into the equivalent register; writing a 0 has no effect.

Therefore, for these registers, three addresses are defined for one unique physical register. Reading these addresses has the same effect and returns the register value.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 10.2.4.2.1 Clear Instruction

##### 10.2.4.2.1.1 Clear Interrupt Enable Registers (GPIO_IRQENABLE_CLR_0 and GPIO_IRQENABLE_CLR_1):

- A write operation in the clear interrupt enable0 (or enable1) register clears the corresponding bit in the interrupt enable0 (or enable1) register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear interrupt enable0 (or enable1) register returns the value of the interrupt enable0 (or enable1) register.

##### 10.2.4.2.1.2 Clear Data Output Register (GPIO_CLEARDATAOUT):

- A write operation in the clear data output register clears the corresponding bit in the data output register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear data output register returns the value of the data output register.

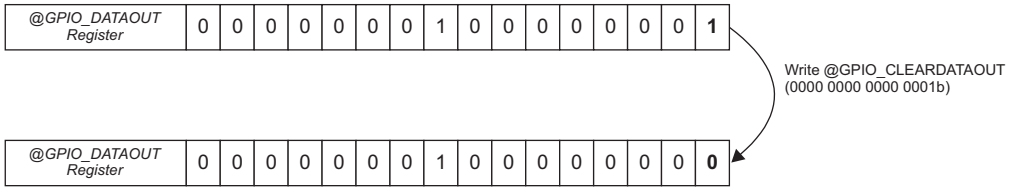
##### 10.2.4.2.1.3 Clear Instruction Example

Assume the data output register (or one of the interrupt enable registers) contains the binary value, 0000 0001 0000 0001h, and you want to clear bit 0.

With the clear instruction feature, write 0000 0000 0000 0001h at the address of the clear data output register (or at the address of the clear interrupt enable register). After this write operation, a reading of the data output register (or the interrupt enable register) returns 0000 0001 0000 0000h; bit 0 is cleared.

**NOTE:** Although the general-purpose interface registers are 32-bits wide, only the 16 least-significant bits are represented in this example.

**Figure 10-4. Write @ GPIO_CLEARDATAOUT Register Example**




#### 10.2.4.2.2 Set Instruction

##### 10.2.4.2.2.1 Set Interrupt Enable Registers (GPIO_IRQENABLE_SET_0 and GPIO_IRQENABLE_SET_1):

- A write operation in the set interrupt enable0 (or enable1) register sets the corresponding bit in the interrupt enable0 (or enable1) register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set interrupt enable0(or enable1) register returns the value of the interrupt enable0 (or enable1) register.

##### 10.2.4.2.2.2 Set Data Output Register (GPIO_SETDATAOUT):

- A write operation in the set data output register sets the corresponding bit in the data output register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set data output register returns the value of the data output register.


Copyright © 2011–2016, Texas Instruments Incorporated

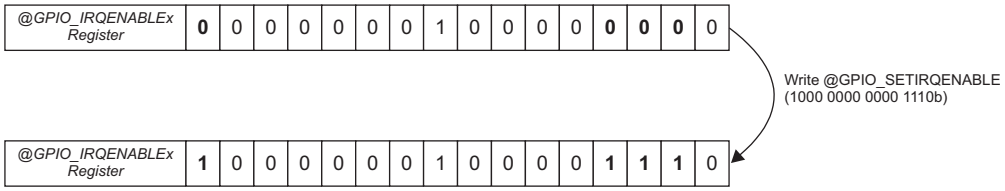
#### *10.2.4.2.2.3 Set Instruction Example*

Assume the interrupt enable0 (or enable1) register (or the data output register) contains the binary value, 0000 0001 0000 0000h, and you want to set bits 15, 3, 2, and 1.

With the set instruction feature, write 1000 0000 0000 1110h at the address of the set interrupt enable0 (or enable1) register (or at the address of the set data output register). After this write operation, a reading of the interrupt enable0 (or enable1) register (or the data output register) returns 1000 0001 0000 1110h; bits 15, 3, 2, and 1 are set.

**NOTE:** Although the general-purpose interface registers are 32-bits wide, only the 16 least-significant bits are represented in this example.

**Figure 10-5. Write @ GPIO_SETIRQENABLEx Register Example**




### 10.2.4.3 Data Input (Capture)/Output (Drive)

The output enable register (GPIO_OE) controls the output/input capability for each pin. At reset, all the GPIO-related pins are configured as input and output capabilities are disabled. This register is not used within the module; its only function is to carry the pads configuration.

When configured as an output (the desired bit reset in GPIO_OE), the value of the corresponding bit in the GPIO_DATAOUT register is driven on the corresponding GPIO pin. Data is written to the data output register synchronously with the interface clock. This register can be accessed with read/write operations or by using the alternate set and clear protocol register update feature. This feature lets you set or clear specific bits of this register with a single write access to the set data output register (GPIO_SETDATAOUT) or to the clear data output register (GPIO_CLEARDATAOUT) address. If the application uses a pin as an output and does not want interrupt generation from this pin, the application must properly configure the interrupt enable registers.

When configured as an input (the desired bit set to 1 in GPIO_OE), the state of the input can be read from the corresponding bit in the GPIO_DATAIN register. The input data is sampled synchronously with the interface clock and then captured in the data input register synchronously with the interface clock. When the GPIO pin levels change, they are captured into this register after two interface clock cycles (the required cycles to synchronize and to write data). If the application uses a pin as an input, the application must properly configure the interrupt enable registers to the interrupt as needed.

### 10.2.4.4 Debouncing Time

To enable the debounce feature for a pin, the GPIO configuration registers must be programmed as follows:

- The GPIO pin must be configured as input in the output enable register (write 1 to the corresponding bit of the GPIO_OE register).
- The debouncing time must be set in the debouncing value register (GPIO_DEBOUNCINGTIME). The GPIO_DEBOUNCINGTIME register is used to set the debouncing time for all input lines in the GPIO module. The value is global for all the ports of one GPIO module, so up to six different debouncing values are possible. The debounce cell is running with the debounce clock (32 kHz). This register represents the number of the clock cycle(s) (one cycle is 31 microseconds long) to be used.


Copyright © 2011–2016, Texas Instruments Incorporated

The following formula describes the required input stable time to be propagated to the debounced output:

*Debouncing time = (DEBOUNCETIME + 1) × 31 µs*

Where the DEBOUNCETIME field value in the GPIO_DEBOUNCINGTIME register is from 0 to 255.

- The debouncing feature must be enabled in the debouncing enable register (write 1 to the corresponding DEBOUNCEENABLE bit in the GPIO_DEBOUNCENABLE register).

#### 10.2.4.5 GPIO as a Keyboard Interface

The general-purpose interface can be used as a keyboard interface (Figure 10-6). You can dedicate channels based on the keyboard matrix size. Figure 10-6 shows row channels configured as inputs with the input debounce feature enabled. The row channels are driven high with an external pull-up. Column channels are configured as outputs and drive a low level.

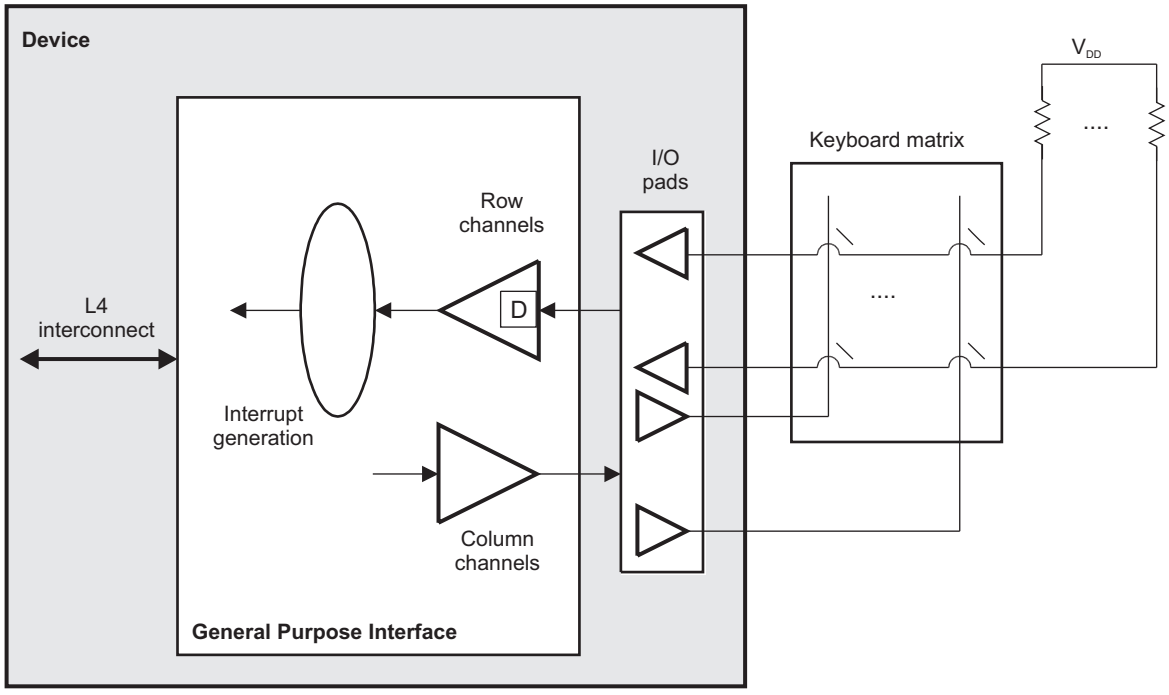
When a keyboard matrix key is pressed, the corresponding row and column lines are shorted together and a low level is driven on the corresponding row channel. This generates an interrupt based on the proper configuration (see Section 10.2.3).

When the keyboard interrupt is received, the processor can disable the keyboard interrupt and scan the column channels for the key coordinates.

- The scanning sequence has as many states as column channels: For each step in the sequence, the processor drives one column channel low and the others high.
- The processor reads the values of the row channels and thus detects which keys in the column are pressed.

At the end of the scanning sequence, the processor establishes which keys are pressed. The keyboard interface can then be reconfigured in the interrupt waiting state.

**Figure 10-6. General-Purpose Interface Used as a Keyboard Interface**





Copyright © 2011–2016, Texas Instruments Incorporated

## 10.3 GPIO Registers

All module registers are 8-, 16-, or 32-bit accessible through the L4 interconnect (little-endian encoding). Access to registers is direct; no shadow registers are implemented.

**Table 10-1. GPIO Registers**

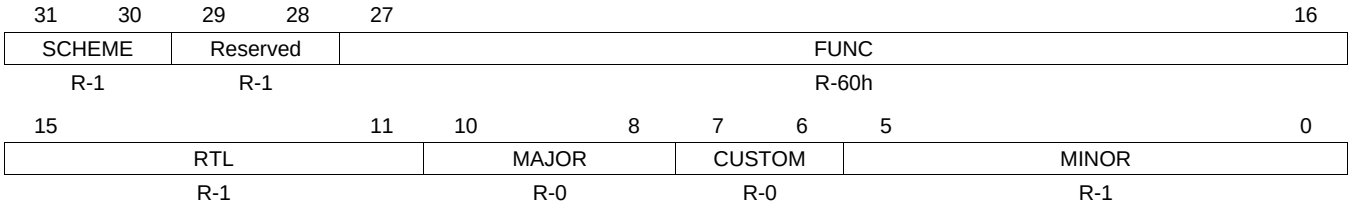
| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | GPIO_REVISION | GPIO Revision Register | Section 10.3.1 |
| 10h | GPIO_SYSCONFIG | System Configuration Register | Section 10.3.2 |
| 20h | GPIO_EOI | End of Interrupt Register | Section 10.3.3 |
| 24h | GPIO_IRQSTATUS_RAW_0 | Status Raw Register for Interrupt 0 | Section 10.3.4 |
| 28h | GPIO_IRQSTATUS_RAW_1 | Status Raw Register for Interrupt 1 | Section 10.3.5 |
| 2Ch | GPIO_IRQSTATUS_0 | Status Register for Interrupt 0 | Section 10.3.6 |
| 30h | GPIO_IRQSTATUS_1 | Status Register for Interrupt 1 | Section 10.3.7 |
| 34h | GPIO_IRQENABLE_SET_0 | Enable Set Register for Interrupt 0 | Section 10.3.8 |
| 38h | GPIO_IRQENABLE_SET_1 | Enable Set Register for Interrupt 1 | Section 10.3.9 |
| 3Ch | GPIO_IRQENABLE_CLR_0 | Enable Clear Register for Interrupt 0 | Section 10.3.10 |
| 40h | GPIO_IRQENABLE_CLR_1 | Enable Clear Register for Interrupt 1 | Section 10.3.11 |
| 114h | GPIO_SYSSTATUS | System Status Register | Section 10.3.12 |
| 130h | GPIO_CTRL | Module Control Register | Section 10.3.13 |
| 134h | GPIO_OE | Output Enable Register | Section 10.3.14 |
| 138h | GPIO_DATAIN | Data Input Register | Section 10.3.15 |
| 13Ch | GPIO_DATAOUT | Data Output Register | Section 10.3.16 |
| 140h | GPIO_LEVELDETECT0 | Low-level Detection Enable Register | Section 10.3.17 |
| 144h | GPIO_LEVELDETECT1 | High-level Detection Enable Register | Section 10.3.18 |
| 148h | GPIO_RISINGDETECT | Rising-edge Detection Enable Register | Section 10.3.19 |
| 14Ch | GPIO_FALLINGDETECT | Falling-edge Detection Enable Register | Section 10.3.20 |
| 150h | GPIO_DEBOUNCENABLE | Debounce Enable Register | Section 10.3.21 |
| 154h | GPIO_DEBOUNCINGTIME | Debouncing Time Register | Section 10.3.22 |
| 190h | GPIO_CLEARDATAOUT | Clear Data Output Register | Section 10.3.23 |
| 194h | GPIO_SETDATAOUT | Set Data Output Register | Section 10.3.24 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.1 GPIO_REVISION Register

The GPIO revision register is a read only register containing the revision number of the GPIO module. A write to this register has no effect, the same as the reset.

**Figure 10-7. GPIO_REVISION Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 10-2. GPIO_REVISION Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | SCHEME | R | 1h | Used to distinguish between old Scheme and current. |
| 29-28 | Reserved | R | 1h | Reserved |
| 27-16 | FUNC | R | 60h | Indicates a software compatible module family. |
| 15-11 | RTL | R | 1Fh | RTL version |
| 10-8 | MAJOR | R | 0h | Major Revision |
| 7-6 | CUSTOM | R | 0h | Indicates a special version for a particular device. |
| 5-0 | MINOR | R | 1Fh | Minor Revision |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.2 GPIO_SYSCONFIG Register

The GPIO_SYSCONFIG register controls the various parameters of the L4 interconnect.

**NOTE:** When the AUTOIDLE bit is set, the GPIO_DATAIN read command has a 3 OCP cycle latency due to the data in sample gating mechanism. When the AUTOIDLE bit is not set, the GPIO_DATAIN read command has a 2 OCP cycle latency.

**Figure 10-8. GPIO_SYSCONFIG Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | IDLEMODE | | Reserved | SOFTRESET | AUTOIDLE |
| R-0 | | R/W-0 | | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 10-3. GPIO_SYSCONFIG Register Field Descriptions**

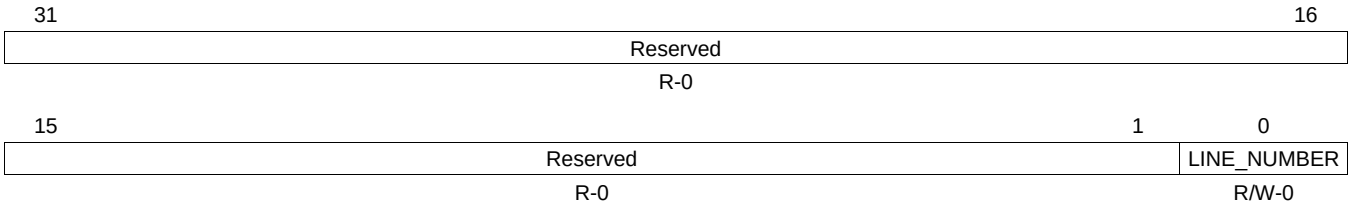
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | Reserved | R | 0h | Reserved |
| 4-3 | IDLEMODE | R/W | 0h | Power Management, Req/Ack control<br>0 = Force-idle. An idle request is acknowledged unconditionally<br>1 = No-idle. An idle request is never acknowledged<br>2 = Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module<br>3 = Smart-idle. |
| 2 | Reserved | R/W | 0h | Reserved. Always write the default value for future device compatibility. |
| 1 | SOFTRESET | R/W | 0h | Software reset. This bit is automatically reset by the hardware. During reads, it always returns 0.<br>0 = Normal mode<br>1 = The module is reset |
| 0 | AUTOIDLE | R/W | 0h | Internal interface clock gating strategy<br>0 = Internal Interface OCP clock is free-running<br>1 = Automatic internal OCP clock gating, based on the OCP interface activity |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.3 GPIO_EOI Register

The GPIO_EOI register provides software end of interrupt (EOI) control.

**Figure 10-9. GPIO_EOI Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 10-4. GPIO_EOI Register Field Descriptions**

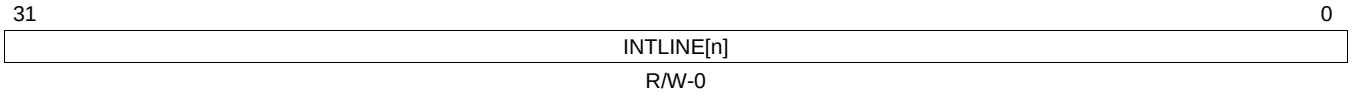
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | Reserved |
| 0 | LINE_NUMBER | R/W | 0h | Software End Of Interrupt (EOI) control. Write number of interrupt output.<br>Write 0 = EOI for interrupt output line #0<br>Write 1 = EOI for interrupt output line #1<br>Read 0 = Reads always 0 (no EOI memory) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.4 GPIO_IRQSTATUS_RAW_0 Register

The GPIO_IRQSTATUS_RAW_0 register provides core status information for the interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit sets it to 1, that is, triggers the IRQ (mostly for debug). Writing a 0 has no effect, that is, the register value is not be modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

**Figure 10-10. GPIO_IRQSTATUS_RAW_0 Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

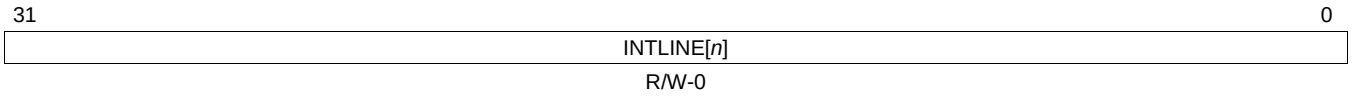
**Table 10-5. GPIO_IRQSTATUS_RAW_0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | INTLINE[*n*] | R/W | 0h | Interrupt n status.<br>0 = No effect<br>1 = IRQ is triggered. |

### 10.3.5 GPIO_IRQSTATUS_RAW_1 Register

The GPIO_IRQSTATUS_RAW_1 register provides core status information for the interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit sets it to 1, that is, triggers the IRQ (mostly for debug). Writing a 0 has no effect, that is, the register value is not be modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

**Figure 10-11. GPIO_IRQSTATUS_RAW_1 Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-6. GPIO_IRQSTATUS_RAW_1 Register Field Descriptions**

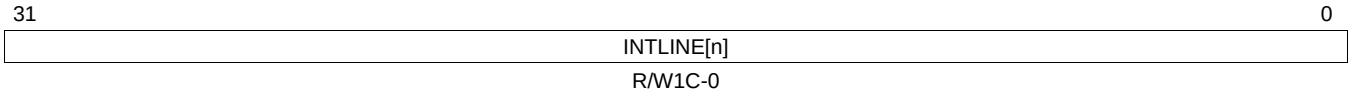
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | INTLINE[*n*] | R/W | 0h | Interrupt n status.<br>0 = No effect<br>1 = IRQ is triggered. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.6 GPIO_IRQSTATUS_0 Register

The GPIO_IRQSTATUS_0 register provides core status information for the interrupt handling, showing all enabled events (enabled interrupts only). The fields are read-write. Writing a 1 to a bit clears the bit to 0, that is, clears the IRQ. Writing a 0 has no effect, that is, the register value is not modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

**Figure 10-12. GPIO_IRQSTATUS_0 Register**




LEGEND: R/W = Read/Write; W1C = Write a 1 to clear to 0, a write of 0 has no effect; -*n* = value after reset

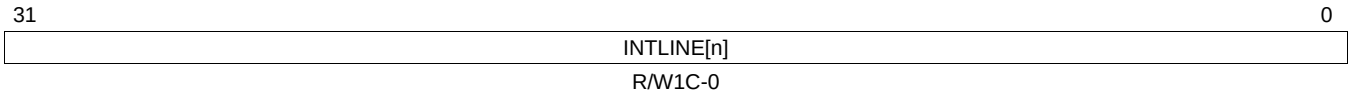
**Table 10-7. GPIO_IRQSTATUS_0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | INTLINE[*n*] | R/W1C | 0h | Interrupt n status.<br>0 = No effect<br>1 = IRQ is triggered. |

### 10.3.7 GPIO_IRQSTATUS_1 Register

The GPIO_IRQSTATUS_1 register provides core status information for the interrupt handling, showing all enabled events (enabled interrupts only). The fields are read-write. Writing a 1 to a bit clears the bit to 0, that is, clears the IRQ. Writing a 0 has no effect, that is, the register value is not modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line.

**Figure 10-13. GPIO_IRQSTATUS_1 Register**




LEGEND: R/W = Read/Write; W1C = Write a 1 to clear to 0, a write of 0 has no effect; -*n* = value after reset

**Table 10-8. GPIO_IRQSTATUS_1 Register Field Descriptions**

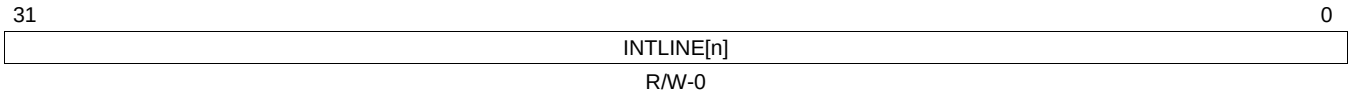
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | INTLINE[*n*] | R/W1C | 0h | Interrupt n status.<br>0 = No effect<br>1 = IRQ is triggered. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.8 GPIO_IRQENABLE_SET_0 Register

All 1-bit fields in the GPIO_IRQENABLE_SET_0 register enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit enables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

**Figure 10-14. GPIO_IRQENABLE_SET_0 Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

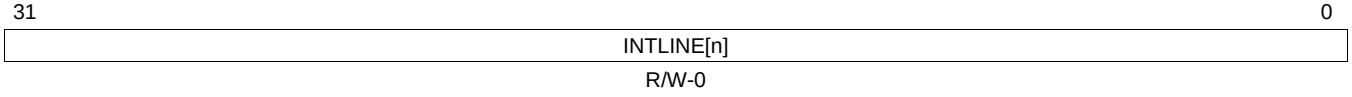
**Table 10-9. GPIO_IRQENABLE_SET_0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | INTLINE[*n*] | R/W | 0h | Interrupt n enable<br>0 = No effect<br>1 = Enable IRQ generation. |

### 10.3.9 GPIO_IRQENABLE_SET_1 Register

All 1-bit fields in the GPIO_IRQENABLE_SET_1 register enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit enables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

**Figure 10-15. GPIO_IRQENABLE_SET_1 Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-10. GPIO_IRQENABLE_SET_1 Register Field Descriptions**

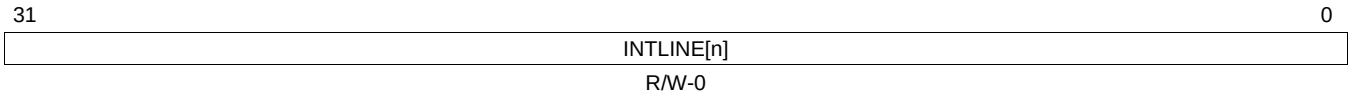
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | INTLINE[*n*] | R/W | 0h | Interrupt n enable<br>0 = No effect<br>1 = Enable IRQ generation. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.10 GPIO_IRQENABLE_CLR_0 Register

All 1-bit fields in the GPIO_IRQENABLE_CLR_0 register clear a specific interrupt event. Writing a 1 to a bit disables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

**Figure 10-16. GPIO_IRQENABLE_CLR_0 Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

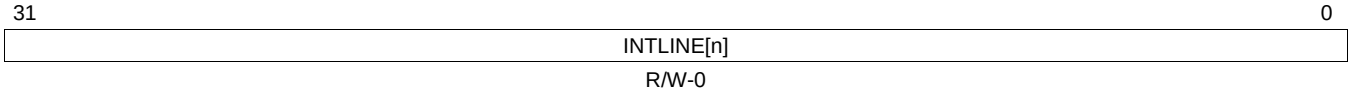
**Table 10-11. GPIO_IRQENABLE_CLR_0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | INTLINE[*n*] | R/W | 0h | Interrupt n enable<br>0 = No effect<br>1 = Disable IRQ generation. |

### 10.3.11 GPIO_IRQENABLE_CLR_1 Register

All 1-bit fields in the GPIO_IRQENABLE_CLR_1 register clear a specific interrupt event. Writing a 1 to a bit disables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified.

**Figure 10-17. GPIO_IRQENABLE_CLR_1 Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-12. GPIO_IRQENABLE_CLR_1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | INTLINE[*n*] | R/W | 0h | Interrupt n enable<br>0 = No effect<br>1 = Disable IRQ generation. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.12 GPIO_SYSSTATUS Register

The GPIO_SYSSTATUS register provides the reset status information about the GPIO module. It is a read-only register; a write to this register has no effect.

**Figure 10-18. GPIO_SYSSTATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | RESETDONE |
| R-0 | | | | | | | | | | | | | | | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 10-13. GPIO_SYSSTATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | Reserved |
| 0 | RESETDONE | R | 0h | Reset status information.<br>0 = Internal Reset is on-going<br>1 = Reset completed |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.13 GPIO_CTRL Register

The GPIO_CTRL register controls the clock gating functionality. The DISABLEMODULE bit controls a clock gating feature at the module level. When set, this bit forces the clock gating for all internal clock paths. Module internal activity is suspended. System interface is not affected by this bit. System interface clock gating is controlled with the AUTOIDLE bit in the system configuration register (GPIO_SYSCONFIG). This bit is to be used for power saving when the module is not used because of the multiplexing configuration selected at the chip level. This bit has precedence over all other internal configuration bits.

**Figure 10-19. GPIO_CTRL Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | GATINGRATIO | | DISABLEMODULE |
| R-0 | | R/W-1 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 10-14. GPIO_CTRL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0h | Reserved |
| 2-1 | GATINGRATIO | R/W | 1h | Gating Ratio. Controls the clock gating for the event detection logic.<br>0 = Functional clock is interface clock.<br>1 = Functional clock is interface clock divided by 2.<br>2 = Functional clock is interface clock divided by 4.<br>3 = Functional clock is interface clock divided by 8. |
| 0 | DISABLEMODULE | R/W | 0h | Module Disable<br>0 = Module is enabled, clocks are not gated.<br>1 = Module is disabled, clocks are gated. |

### 10.3.14 GPIO_OE Register

The GPIO_OE register is used to enable the pins output capabilities. At reset, all the GPIO related pins are configured as input and output capabilities are disabled. This register is not used within the module, its only function is to carry the pads configuration. When the application is using a pin as an output and does not want interrupt generation from this pin, the application can/has to configure properly the Interrupt Enable registers.

**Figure 10-20. GPIO_OE Register**

| 31 | 0 |
|---|---|
| OUTPUTEN[n] | |
| R/W-FFFF FFFFh | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-15. GPIO_OE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | OUTPUTEN[*n*] | R/W | FFFF FFFFh | Output Data Enable<br>0 = The corresponding GPIO port is configured as an output.<br>1 = The corresponding GPIO port is configured as an input. |

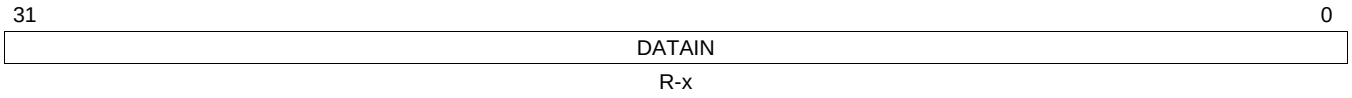

Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.15 GPIO_DATAIN Register

The GPIO_DATAIN register is used to register the data that is read from the GPIO pins. The GPIO_DATAIN register is a read-only register. The input data is sampled synchronously with the interface clock and then captured in the GPIO_DATAIN register synchronously with the interface clock. So, after changing, GPIO pin levels are captured into this register after two interface clock cycles (the required cycles to synchronize and to write the data).

**NOTE:** When the AUTOIDLE bit in the system configuration register (GPIO_SYSCONFIG) is set, the GPIO_DATAIN read command has a 3 OCP cycle latency due to the data in sample gating mechanism. When the AUTOIDLE bit is not set, the GPIO_DATAIN read command has a 2 OCP cycle latency.

**Figure 10-21. GPIO_DATAIN Register**




LEGEND: R = Read only; -*n* = value after reset

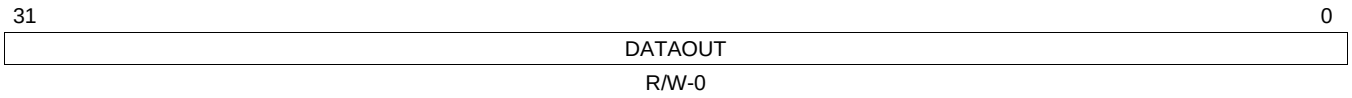
**Table 10-16. GPIO_DATAIN Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATAIN | R | x | Sampled Input Data |

### 10.3.16 GPIO_DATAOUT Register

The GPIO_DATAOUT register is used for setting the value of the GPIO output pins. Data is written to the GPIO_DATAOUT register synchronously with the interface clock. This register can be accessed with direct read/write operations or using the alternate 'Set/Clear' feature. This feature enables to set or clear specific bits of this register with a single write access to the set data output register (GPIO_SETDATAOUT) or to the clear data output register (GPIO_CLEARDATAOUT) address.

**Figure 10-22. GPIO_DATAOUT Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-17. GPIO_DATAOUT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATAOUT | R/W | 0 | Data to set on output pins |

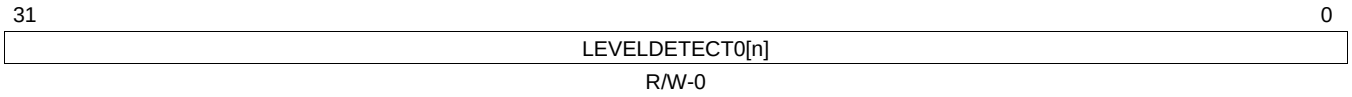

Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.17 GPIO_LEVELDETECT0 Register

The GPIO_LEVELDETECT0 register is used to enable/disable for each input lines the low-level (0) detection to be used for the interrupt request generation.

**NOTE:** Enabling at the same time high-level detection and low-level detection for one given pin makes a constant interrupt generator.

**Figure 10-23. GPIO_LEVELDETECT0 Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-18. GPIO_LEVELDETECT0 Register Field Descriptions**

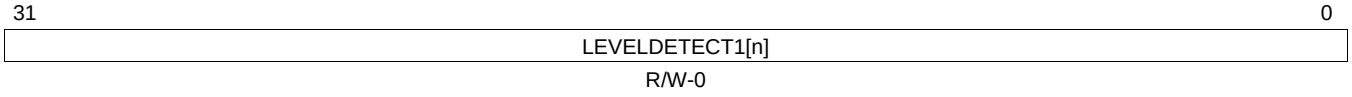
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | LEVELDETECT0[*n*] | R/W | 0h | Low Level Interrupt Enable<br>0 = Disable the IRQ assertion on low-level detect.<br>1 = Enable the IRQ assertion on low-level detect. |

### 10.3.18 GPIO_LEVELDETECT1 Register

The GPIO_LEVELDETECT1 register is used to enable/disable for each input lines the high-level (1) detection to be used for the interrupt request generation.

**NOTE:** Enabling at the same time high-level detection and low-level detection for one given pin makes a constant interrupt generator.

**Figure 10-24. GPIO_LEVELDETECT1 Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-19. GPIO_LEVELDETECT1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | LEVELDETECT1[*n*] | R/W | 0h | High Level Interrupt Enable<br>0 = Disable the IRQ assertion on high-level detect.<br>1 = Enable the IRQ assertion on high-level detect. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.19 GPIO_RISINGDETECT Register

The GPIO_RISINGDETECT register is used to enable/disable for each input lines the rising-edge (transition 0 to 1) detection to be used for the interrupt request generation.

**Figure 10-25. GPIO_RISINGDETECT Register**

| 31 | 0 |
|---|---|
| RISINGDETECT[n] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-20. GPIO_RISINGDETECT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RISINGDETECT[*n*] | R/W | 0h | Rising Edge Interrupt Enable<br>0 = Disable IRQ on rising-edge detect.<br>1 = Enable IRQ on rising-edge detect. |

### 10.3.20 GPIO_FALLINGDETECT Register

The GPIO_FALLINGDETECT register is used to enable/disable for each input lines the falling-edge (transition 1 to 0) detection to be used for the interrupt request generation.

**Figure 10-26. GPIO_FALLINGDETECT Register**

| 31 | 0 |
|---|---|
| FALLINGDETECT[n] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-21. GPIO_FALLINGDETECT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | FALLINGDETECT[*n*] | R/W | 0h | Falling Edge Interrupt Enable<br>0 = Disable IRQ on falling-edge detect.<br>1 = Enable IRQ on falling-edge detect. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.21 GPIO_DEBOUNCENABLE Register

The GPIO_DEBOUNCENABLE register is used to enable/disable the debouncing feature for each input line.

**Figure 10-27. GPIO_DEBOUNCENABLE Register**

| 31 | 0 |
|---|---|
| DEBOUNCEENABLE[n] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-22. GPIO_DEBOUNCENABLE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DEBOUNCEENABLE[*n*] | R/W | 0h | Input Debounce Enable<br>0 = Disable debouncing feature on the corresponding input port.<br>1 = Enable debouncing feature on the corresponding input port. |

### 10.3.22 GPIO_DEBOUNCINGTIME Register

The GPIO_DEBOUNCINGTIME register controls debouncing time (the value is global for all ports). The debouncing cell is running with the debouncing clock (32 kHz), this register represents the number of the clock cycle(s) (31µs long) to be used.

**Figure 10-28. GPIO_DEBOUNCINGTIME Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DEBOUNCETIME | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-23. GPIO_DEBOUNCINGTIME Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | Reserved | R/W | 0h | Reserved |
| 7-0 | DEBOUNCETIME | R/W | 0h | Input Debouncing Value in 31 microsecond steps.<br>Debouncing time = (DEBOUNCETIME + 1) × 31 microseconds |


Copyright © 2011–2016, Texas Instruments Incorporated

### 10.3.23 GPIO_CLEARDATAOUT Register

Writing a 1 to a bit in the GPIO_CLEARDATAOUT register clears to 0 the corresponding bit in the GPIO_DATAOUT register; writing a 0 has no effect. A read of the GPIO_CLEARDATAOUT register returns the value of the data output register (GPIO_DATAOUT).

**Figure 10-29. GPIO_CLEARDATAOUT Register**

| 31 | 0 |
|---|---|
| INTLINE [n] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-24. GPIO_CLEARDATAOUT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | INTLINE [*n*] | R/W | 0h | Clear Data Output Register<br>0 = No effect<br>1 = Clear the corresponding bit in the GPIO_DATAOUT register. |

### 10.3.24 GPIO_SETDATAOUT Register

Writing a 1 to a bit in the GPIO_SETDATAOUT register sets to 1 the corresponding bit in the GPIO_DATAOUT register; writing a 0 has no effect. A read of the GPIO_SETDATAOUT register returns the value of the data output register (GPIO_DATAOUT).

**Figure 10-30. GPIO_SETDATAOUT Register**

| 31 | 0 |
|---|---|
| INTLINE [n] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 10-25. GPIO_SETDATAOUT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | INTLINE [*n*] | R/W | 0h | Set Data Output Register<br>0 = No effect<br>1 = Set the corresponding bit in the GPIO_DATAOUT register. |


Copyright © 2011–2016, Texas Instruments Incorporated

# General-Purpose Memory Controller (GPMC)

This chapter describes the general-purpose memory controller (GPMC).

**Topic** **Page**

**11.1 Introduction** ..... **1631**
**11.2 Architecture** ..... **1633**
**11.3 Basic Programming Model** ..... **1712**
**11.4 Use Cases And Tips** ..... **1731**
**11.5 Registers** ..... **1743**



Copyright © 2011–2016, Texas Instruments Incorporated

## 11.1 Introduction

### 11.1.1 Overview

The general-purpose memory controller (GPMC) is an unified memory controller dedicated to interfacing external memory devices:

- Asynchronous SRAM-like memories and application-specific integrated circuit (ASIC) devices
- Asynchronous, synchronous, and page mode (only available in non-multiplexed mode) burst NOR flash devices
- NAND Flash
- Pseudo-SRAM devices

### 11.1.2 Block Diagram

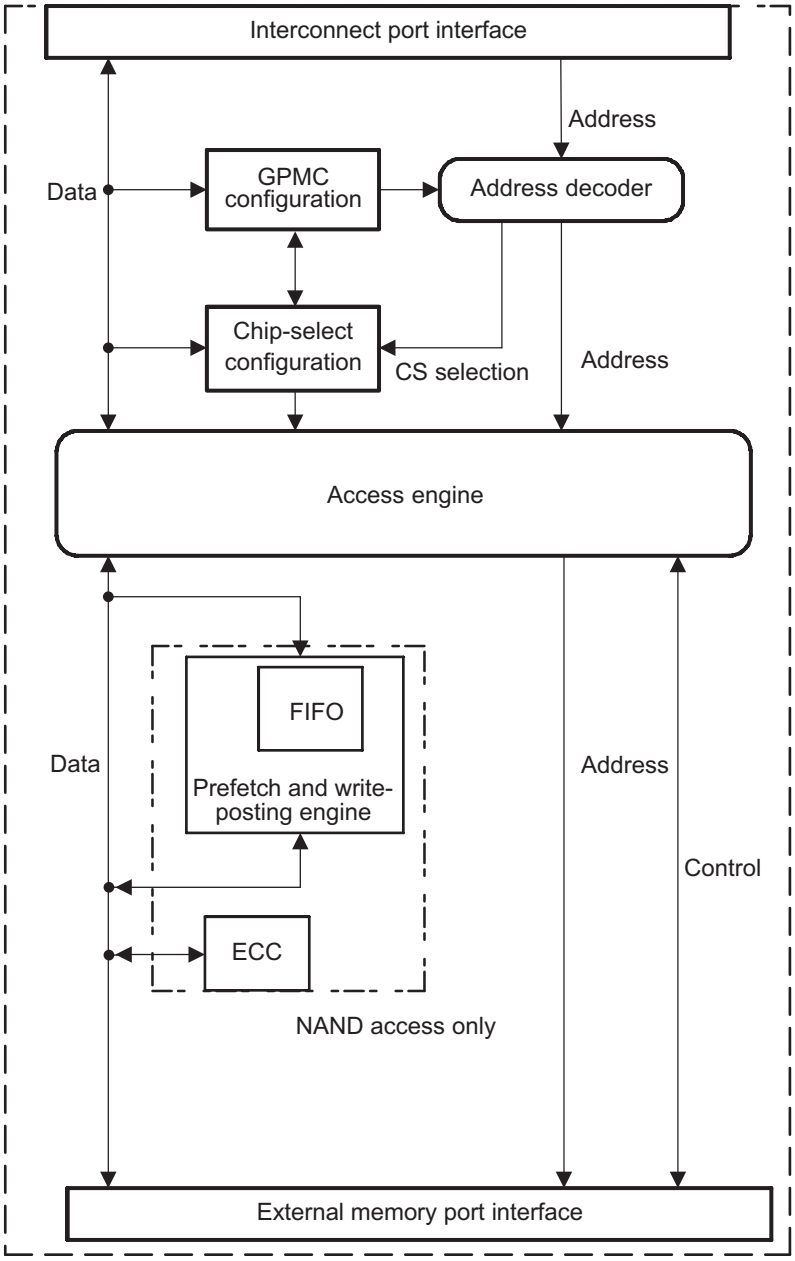
The GPMC can access various external devices through the L3 Slow Interconnect. The flexible programming model allows a wide range of attached device types and access schemes. Based on the programmed configuration bit fields stored in the GPMC registers, the GPMC is able to generate all control signals timing depending on the attached device and access type. Given the chip-select decoding and its associated configuration registers, the GPMC selects the appropriate device type control signals timing.

Figure 11-1 shows the GPMC functional block diagram. The GPMC consists of six blocks:

- Interconnect port interface
- Address decoder, GPMC configuration, and chip-select configuration register file
- Access engine
- Prefetch and write-posting engine
- Error correction code engine (ECC)
- External device/memory port interface


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-1. GPMC Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

## 11.2 Architecture

### 11.2.1 GPMC Signals

Table 11-1 lists the GPMC subsystem I/O pins.

**NOTE:** In this chapter, the i in GPMC_CONFIGx_i represents GPMC chip-select i, where i = 0 to 5.

**Table 11-1. GPMC I/O Description**

| Pin Name | I/O | Description |
|---|---|---|
| GPMC_A[27:0] | O | Address |
| GPMC_D[15:0] | I/O | Data |
| GPMC_CS[5:0] | O | Chip-selects (active low) |
| GPMC_CLK | O | Clock generated for the external memory or device |
| GPMC_ADV_ALE | O | Address valid (active low). Also used as address latch enable (active high) for NAND protocol memories. |
| GPMC_OE_RE | O | Output enable (active low). Also used as read enable (active low) for NAND protocol memories. |
| GPMC_WE | O | Write enable (active low) |
| GPMC_BE0_CLE | O | Lower-byte enable (active low). Also used as command latch enable for NAND protocol memories. |
| GPMC_BE1 | O | Upper-byte enable (active low) |
| GPMC_WAIT | I | External wait signal for NOR and NAND protocol memories. The wait signal can be mapped on any of the chip-selects. |

Table 11-2 shows the use of address and data GPMC controller pins based on the type of external device.

**Table 11-2. GPMC Pin Multiplexing Options**

| GPMC Pin | Non Multiplexed Address Data 16-Bit Device | Non Multiplexed Address Data 8-Bit Device | Multiplexed Address Data 16-Bit Device | 16-Bit NAND Device | 8-Bit NAND Device |
|---|---|---|---|---|---|
| GPMC_A[27] | A26 | A27 | A27 | Not Used | Not Used |
| GPMC_A[26] | A25 | A26 | Not Used | Not Used | Not Used |
| GPMC_A[25] | A24 | A25 | Not Used | Not Used | Not Used |
| GPMC_A[24] | A23 | A24 | Not Used | Not Used | Not Used |
| GPMC_A[23] | A22 | A23 | Not Used | Not Used | Not Used |
| GPMC_A[22] | A21 | A22 | Not Used | Not Used | Not Used |
| GPMC_A[21] | A20 | A21 | Not Used | Not Used | Not Used |
| GPMC_A[20] | A19 | A20 | Not Used | Not Used | Not Used |
| GPMC_A[19] | A18 | A19 | Not Used | Not Used | Not Used |
| GPMC_A[18] | A17 | A18 | Not Used | Not Used | Not Used |
| GPMC_A[17] | A16 | A17 | Not Used | Not Used | Not Used |
| GPMC_A[16] | A15 | A16 | Not Used | Not Used | Not Used |
| GPMC_A[15] | A14 | A15 | Not Used | Not Used | Not Used |
| GPMC_A[14] | A13 | A14 | Not Used | Not Used | Not Used |
| GPMC_A[13] | A12 | A13 | Not Used | Not Used | Not Used |
| GPMC_A[12] | A11 | A12 | Not Used | Not Used | Not Used |
| GPMC_A[11] | A10 | A11 | Not Used | Not Used | Not Used |
| GPMC_A[10] | A9 | A10 | A26 | Not Used | Not Used |
| GPMC_A[9] | A8 | A9 | A25 | Not Used | Not Used |
| GPMC_A[8] | A7 | A8 | A24 | Not Used | Not Used |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-2. GPMC Pin Multiplexing Options (continued)**

| GPMC Pin | Non Multiplexed Address Data 16-Bit Device | Non Multiplexed Address Data 8-Bit Device | Multiplexed Address Data 16-Bit Device | 16-Bit NAND Device | 8-Bit NAND Device |
|---|---|---|---|---|---|
| GPMC_A[7] | A6 | A7 | A23 | Not Used | Not Used |
| GPMC_A[6] | A5 | A6 | A22 | Not Used | Not Used |
| GPMC_A[5] | A4 | A5 | A21 | Not Used | Not Used |
| GPMC_A[4] | A3 | A4 | A20 | Not Used | Not Used |
| GPMC_A[3] | A2 | A3 | A19 | Not Used | Not Used |
| GPMC_A[2] | A1 | A2 | A18 | Not Used | Not Used |
| GPMC_A[1] | A0 | A1 | A17 | Not Used | Not Used |
| GPMC_A[0] | Not Used | A0 | Not Used | Not Used | Not Used |
| GPMC_D[15] | D15 | Not Used | A16/D15 | D15 | Not Used |
| GPMC_D[14] | D14 | Not Used | A15/D14 | D14 | Not Used |
| GPMC_D[13] | D13 | Not Used | A14/D13 | D13 | Not Used |
| GPMC_D[12] | D12 | Not Used | A13/D12 | D12 | Not Used |
| GPMC_D[11] | D11 | Not Used | A12/D11 | D11 | Not Used |
| GPMC_D[10] | D10 | Not Used | A11/D10 | D10 | Not Used |
| GPMC_D[9] | D9 | Not Used | A10/D9 | D9 | Not Used |
| GPMC_D[8] | D8 | Not Used | A9/D8 | D8 | Not Used |
| GPMC_D[7] | D7 | D7 | A8/D7 | D7 | D7 |
| GPMC_D[6] | D6 | D6 | A7/D6 | D6 | D6 |
| GPMC_D[5] | D5 | D5 | A6/D5 | D5 | D5 |
| GPMC_D[4] | D4 | D4 | A5/D4 | D4 | D4 |
| GPMC_D[3] | D3 | D3 | A4/D3 | D3 | D3 |
| GPMC_D[2] | D2 | D2 | A3/D2 | D2 | D2 |
| GPMC_D[1] | D1 | D1 | A2/D1 | D1 | D1 |
| GPMC_D[0] | D0 | D0 | A1/D0 | D0 | D0 |

With all device types, the GPMC does not drive unnecessary address lines. They stay at their reset value of 00.

Address mapping supports address/data-multiplexed 16-bit wide devices:

- The NOR flash memory controller still supports non-multiplexed address and data memory devices.
- Multiplexing mode can be selected through the GPMC_CONFIG1_i[9-8] MUXADDDATA bit field (i = 0 to 5).
- Asynchronous page mode is not supported for multiplexed address and data devices.



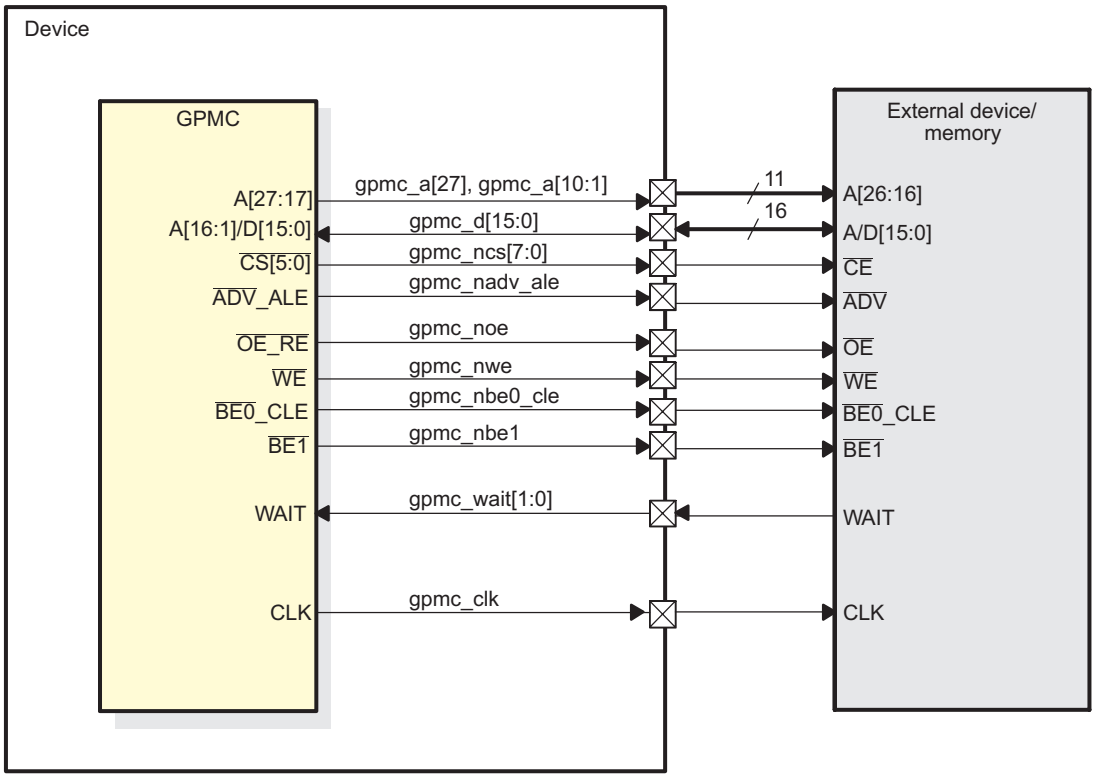
Copyright © 2011–2016, Texas Instruments Incorporated

### 11.2.2 GPMC Modes

This section shows three GPMC external connections options:

- Figure 11-2 shows a connection between the GPMC and a 16-bit synchronous address/data-multiplexed (or AAD-multiplexed, but this protocol use less address pins) external memory device.
- Figure 11-3 shows a connection between the GPMC and a 16-bit synchronous nonmultiplexed external memory device .
- Figure 11-4 shows a connection between the GPMC and a 8-bit NAND device

**Figure 11-2. GPMC to 16-Bit Address/Data-Multiplexed Memory**




A If ROM boot from the NOR device is required, some high-order address lines must be driven or pulled low by external hardware during ROM operation. See the *ROM Code Memory and Peripheral Booting* chapter, XIP Memory section, and the device data manual for details.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-3. GPMC to 16-Bit Nonmultiplexed Memory**

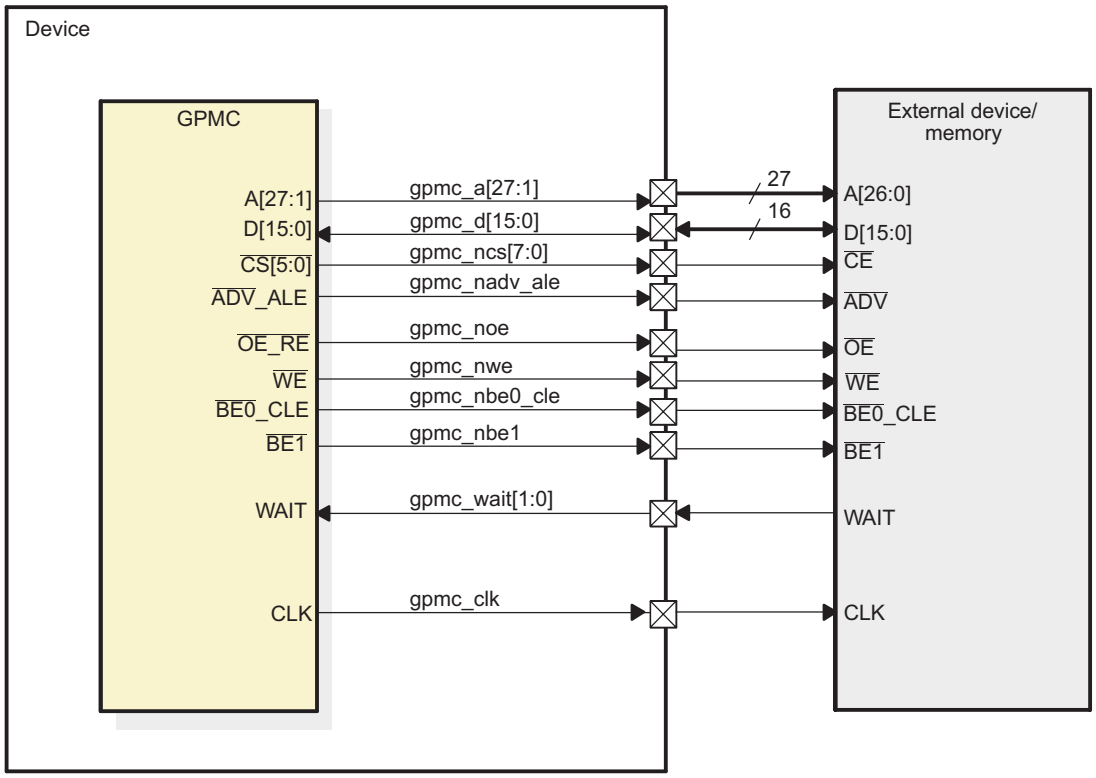
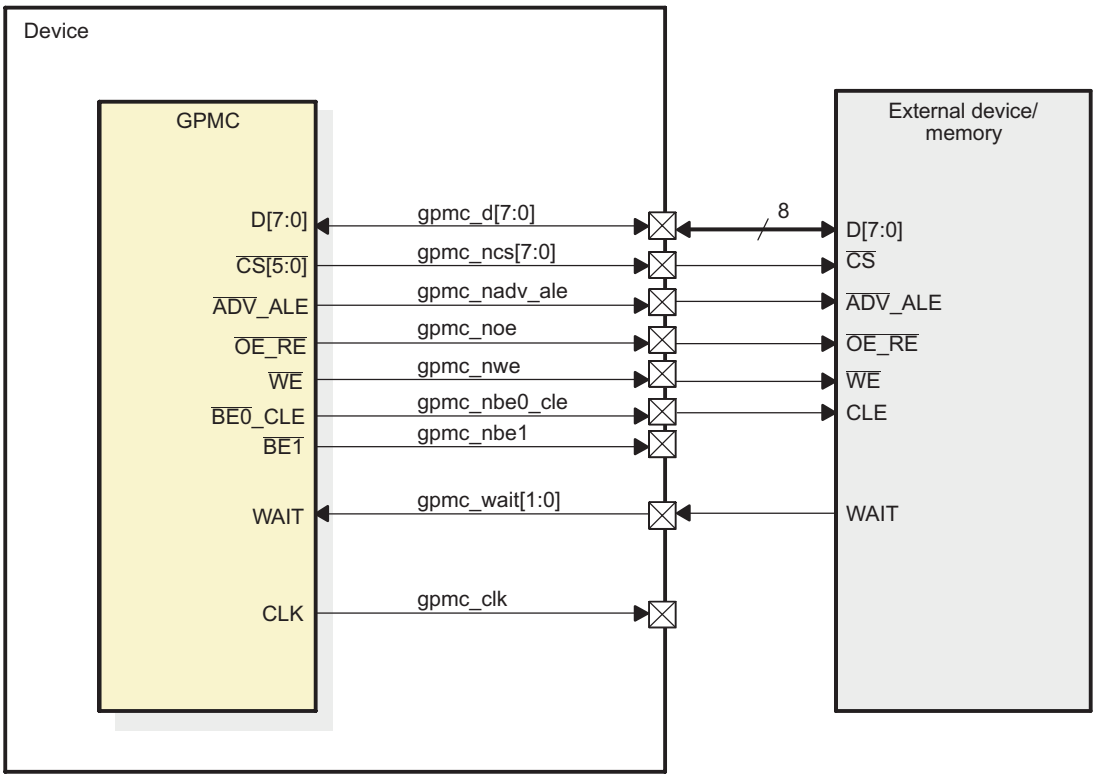



**Figure 11-4. GPMC to 8-Bit NAND Device**





Copyright © 2011–2016, Texas Instruments Incorporated

### 11.2.3 GPMC Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

- No STANDBY hardware handshake
- No wake-up request
- One system direct memory access (eDMA) request
- One interrupt request to the Cortex-A8 MPU Interrupt Controller (MA_IRQ)
- One clock for functional and interface domains

Figure 11-5 shows GPMC integration. Table 11-3 through Table 11-5 summarize the integration of the module in the device.

**Figure 11-5. GPMC Integration**

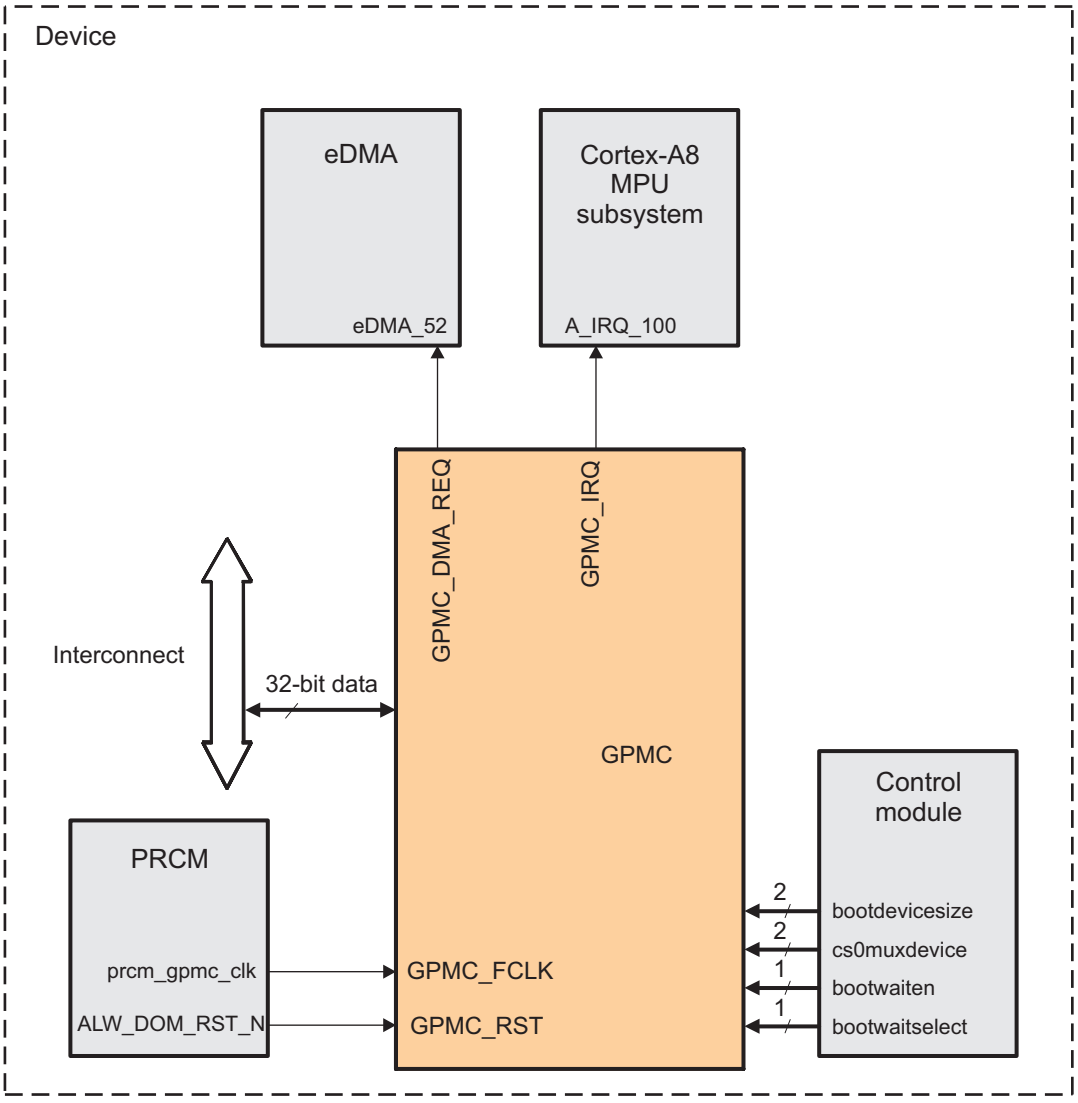



**Table 11-3. GPMC Integration Attributes**

| Module Instance | Attributes | | |
|---|---|---|---|
| | Power Domain | Wake-up Capability | Interconnect |
| GPMC | ALWAYS_ON | No | L3_Slow |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-4. GPMC Clocks and Resets**

| Module Instance | Destination Signal Name | Source Signal Name | Source | Description |
|---|---|---|---|---|
| **Clocks** | | | | |
| GPMC | GPMC_FCLK | prcm_GPMC_CLK | PRCM | Functional clock |
| **Resets** | | | | |
| GPMC | GPMC_RST | ALW_DOM_RST | PRCM | GPMC reset |

**Table 11-5. GPMC Hardware Requests**

| Module Instance | Source Signal Name | Destination Signal Name | Destination | Description |
|---|---|---|---|---|
| **Interrupt Requests** | | | | |
| GPMC | GPMC_IRQ | A_IRQ_100 | Cortex-A8 | GPMC interrupt to Cortex-A8 MPU subsystem |
| **DMA Requests** | | | | |
| GPMC | GPMC_DMA_REQ | e_DMA_52 | eDMA | GPMC request from Prefetch Engine to eDMA |

### *11.2.4 GPMC Functional Description*

The GPMC basic programming model offers maximum flexibility to support various access protocols for each of the six configurable chip-selects. Use optimal chip-select settings, based on the characteristics of the external device:

- Different protocols can be selected to support generic asynchronous or synchronous random-access devices (NOR flash, SRAM) or to support specific NAND devices.
- The address and the data bus can be multiplexed on the same external bus.
- Read and write access can be independently defined as asynchronous or synchronous.
- System requests (byte, 16-bit word, burst) are performed through single or multiple accesses. External access profiles (single, multiple with optimized burst length, native- or emulated-wrap) are based on external device characteristics (supported protocol, bus width, data buffer size, native-wrap support).
- System burst read or write requests are synchronous-burst (multiple-read or multiple-write). When neither burst nor page mode is supported by external memory or ASIC devices, system burst read or write requests are translated to successive single synchronous or asynchronous accesses (single reads or single writes). 8-bit wide devices are supported only in single synchronous or single asynchronous read or write mode.
- To simulate a programmable internal-wait state, an external wait pin can be monitored to dynamically control external access at the beginning (initial access time) of and during a burst access.

Each control signal is controlled independently for each chip-select. The internal functional clock of the GPMC (GPMC_FCLK) is used as a time reference to specify the following:

- Read- and write-access duration
- Most GPMC external interface control-signal assertion and deassertion times
- Data-capture time during read access
- External wait-pin monitoring time
- Duration of idle time between accesses, when required


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.1 GPMC Clock Configuration

Table 11-6 describes the GPMC clocks.

**Table 11-6. GPMC Clocks**

| Signal | I/O | Description |
|---|---|---|
| GPMC_FCLK | I | Functional and interface clock |
| GPMC_CLK | O | External clock provided to synchronous external memory devices. |

The GPMC_CLK is generated by the GPMC from the internal GPMC_FCLK clock. The source of the GPMC_FCLK is described in Table 11-4. The GPMC_CLK is configured via the GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER field (for i = 0 to 3) as shown in Table 11-7.

**Table 11-7. GPMC_CONFIG1_i Configuration**

| Source Clock | GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER | GPMC_CLK Generated Clock Provided to External Memory Device |
|---|---|---|
| GPMC_FCLK | 00 | GPMC_FCLK |
| | 01 | GPMC_FCLK/2 |
| | 10 | GPMC_FCLK/4 |
| | 11 | GPMC_FCLK/8 |

#### 11.2.4.2 GPMC Software Reset

The GPMC can be reset by software through the GPMC_SYSCONFIG[1] SOFTRESET bit. Setting the bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Hardware and software resets initialize all GPMC registers and the finite state-machine (FSM) immediately and unconditionally. The GPMC_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. The software must ensure that the software reset completes before doing GPMC operations.

#### 11.2.4.3 GPMC Power Management

GPMC power is supplied by the CORE power domain, and GPMC power management complies with system power-management guidelines. Table 11-8 describes power-management features available for the GPMC module.

**Table 11-8. GPMC Local Power Management Features**

| Feature | Registers | Description |
|---|---|---|
| Clock Auto Gating | GPMC_SYSCONFIG[0] AUTOIDLE] bit | This bit allows a local power optimization inside the module, by gating the GPMC_FCLK clock upon the internal activity. |
| Slave Idle Modes | GPMC_SYSCONFIG[4-3] SIDLEMODE bit field | Force-idle, No-idle and Smart-idle wakeup modes are available |
| Clock Activity | N/A | Feature not available |
| Master Standby Modes | N/A | Feature not available |
| Global Wake-up Enable | N/A | Feature not available |
| Wake-up Sources Enable | N/A | Feature not available |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.2.4.4 GPMC Interrupt Requests

The GPMC generates one interrupt event as shown in Figure 11-5.

- The interrupt request goes from GPMC (GPMC_IRQ) to the Cortex-A8 MPU subsystem: A_IRQ_100

Table 11-9 lists the event flags, and their mask, that can cause module interrupts.

**Table 11-9. GPMC Interrupt Events**

| Event Flag | Event Mask | Sensitivity | Map to | Description |
|---|---|---|---|---|
| GPMC_IRQSTATUS[9] WAIT1EDGEDETECTIONSTATUS | GPMC_IRQENABLE[9] WAIT1EDGEDETECTIONENABLE | Edge | A_IRQ_100 | Wait1 edge detection interrupt: Triggered if a rising or falling edge is detected on the GPMC_WAIT1 signal. The rising or falling edge detection of Wait1 is selected through GPMC_CONFIG[9] WAIT1PINPOLARITY bit. |
| GPMC_IRQSTATUS[8] WAIT0EDGEDETECTIONSTATUS | GPMC_IRQENABLE[8] WAIT0EDGEDETECTIONENABLE | Edge | A_IRQ_100 | Wait0 edge detection interrupt: Triggered if a rising or falling edge is detected on the GPMC_WAIT0 signal. The rising or falling edge detection of Wait0 is selected through GPMC_CONFIG[8] WAIT0PINPOLARITY bit. |
| GPMC_IRQSTATUS[1] TERMINALCOUNTSTATUS | GPMC_IRQENABLE[1] TERMINALCOUNTENABLE | Level | A_IRQ_100 | Terminal count event: Triggered on prefetch process completion, that is when the number of currently remaining data to be requested reaches 0. |
| GPMC_IRQSTATUS[0] FIFOEVENTSTATUS | GPMC_IRQENABLE[0] FIFOEVENTENABLE | Level | A_IRQ_100 | FIFO event interrupt: Indicates FIFO levels availability for in Write-Posting mode and prefetch mode. GPMC_PREFETCH_CONFIG[2] DMAMODE bit shall be cleared to 0. |

### 11.2.4.5 GPMC DMA Requests

The GPMC generates one DMA event as shown in Figure 11-5.

- From GPMC (GPMC_DMA_REQ) to the eDMA: e_DMA_53

### 11.2.4.6 L3 Slow Interconnect Interface

The GPMC L3 Slow interconnect interface is a pipelined interface including an 16 × 32-bit word write buffer. Any system host can issue external access requests through the GPMC. The device system can issue the following requests through this interface:

- One 8-bit / 16-bit / 32-bit interconnect access (read/write)
- Two incrementing 32-bit interconnect accesses (read/write)
- Two wrapped 32-bit interconnect accesses (read/write)
- Four incrementing 32-bit interconnect accesses (read/write)
- Four wrapped 32-bit interconnect accesses (read/write)
- Eight incrementing 32-bit interconnect accesses (read/write)
- Eight wrapped 32-bit interconnect accesses (read/write)

Only linear burst transactions are supported; interleaved burst transactions are not supported. Only power-of-two-length precise bursts 2 × 32, 4 × 32, 8 × 32 or 16 × 32 with the burst base address aligned on the total burst size are supported (this limitation applies to incrementing bursts only).

This interface also provides one interrupt and one DMA request line, for specific event control.

It is recommended to program the GPMC_CONFIG1_i ATTACHEDDEVICEPAGELENGTH field ([24-23]) according to the effective attached device page length and to enable the GPMC_CONFIG1_i WRAPBURST bit ([31]) if the attached device supports wrapping burst. However, it is possible to emulate wrapping burst on a non-wrapping memory by providing relevant addresses within the page or splitting transactions. Bursts larger than the memory page length are chopped into multiple bursts transactions. Due to the alignment requirements, a page boundary is never crossed.


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.2.4.7 GPMC Address and Data Bus

The current application supports GPMC connection to NAND devices and to address/data-multiplexed memories or devices. Depending on the GPMC configuration of each chip-select, address and data bus lines that are not required for a particular access protocol are not updated (changed from current value) and are not sampled when input (input data bus).

- For address/data-multiplexed and AAD-multiplexed NOR devices, the address is multiplexed on the data bus.
- 8-bit wide NOR devices do not use GPMC I/O: GPMC_D[15-8] for data (they are used for address if needed).
- 16-bit wide NAND devices do not use GPMC I/O: GPMC_A[27-0].
- 8-bit wide NAND devices do not use GPMC I/O: GPMC_A[27-0] and GPMC I/O: GPMC_D[15-8].

#### *11.2.4.7.1 GPMC I/O Configuration Setting*

To select a NAND device, program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 10
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 00

To select an address/data-multiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 00
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 10

To select an address/address/data-multiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 00
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 01

To select an address/data-nonmultiplexed device , program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 00
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 00

### 11.2.4.8 Address Decoder and Chip-Select Configuration

Addresses are decoded accordingly with the address request of the chip-select and the content of the chip-select base address register file, which includes a set of global GPMC configuration registers and eight sets of chip-select configuration registers.

The GPMC configuration register file is memory-mapped and can be read or written with byte, 16-bit word, or 32-bit word accesses. The register file should be configured as a noncacheable, nonbufferable region to prevent any desynchronization between host execution (write request) and the completion of register configuration (write completed with register updated). Section 11.5 provides the GPMC register locations. For the map of GPMC memory locations, see Table 11-54.

After the chip-select is configured, the access engine accesses the external device, drives the external interface control signals, and applies the interface protocol based on user-defined timing parameters and settings.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.8.1 Chip-Select Base Address and Region Size

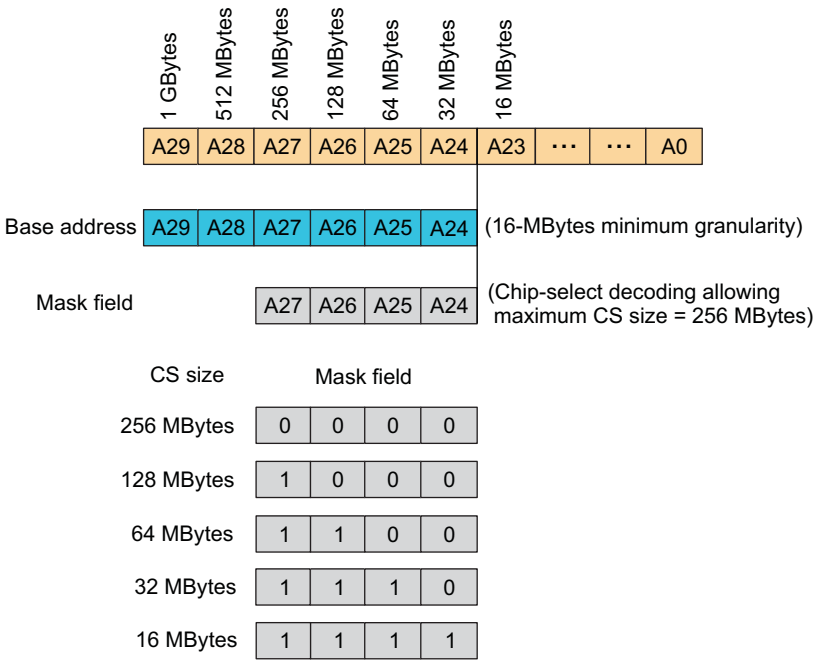
Any external memory or ASIC device attached to the GPMC external interface can be accessed by any device system host within the GPMC 512-Mbyte contiguous address space. For details, see Table 11-54.

The GPMC 512 Mbyte address space can be divided into a maximum of eight chip-select regions with programmable base address and programmable CS size. The CS size is programmable from 16 Mbytes to 256 Mbytes (must be a power-of-2) and is defined by the mask field. Attached memory smaller than the programmed CS region size is accessed through the entire CS region (aliasing).

Each chip-select has a 6-bit base address encoding and a 4-bit decoding mask, which must be programmed according to the following rules:

- The programmed chip-select region base address must be aligned on the chip-select region size address boundary and is limited to a power-of-2 address value. During access decoding, the register base address value is used for address comparison with the address-bit line mapping as described in Figure 11-6 (with A0 as the device system byte-address line). Base address is programmed through the GPMC_CONFIG7_i[5-0] BASEADDRESS bit field.
- The register mask is used to exclude some address lines from the decoding. A register mask bit field cleared to 0 suppresses the associated address line from the address comparison (incoming address bit line is don't care). The register mask value must be limited to the subsequent value, based on the desired chip-select region size. Any other value has an undefined result. When multiple chip-select regions with overlapping addresses are enabled concurrently, access to these chip-select regions is cancelled and a GPMC access error is posted. The mask field is programmed through the GPMC_CONFIG7_i[11-8] MASKADDRESS bit field.

**Figure 11-6. Chip-Select Address Mapping and Decoding Mask**




Chip-select configuration (base and mask address or any protocol and timing settings) must be performed while the associated chip-select is disabled through the GPMC_CONFIG7_i[6] CSVALID bit (i = 0 to 5). In addition, a chip-select configuration can only be disabled if there is no ongoing access to that chip-select. This requires activity monitoring of the prefetch or write-posting engine if the engine is active on the chip-select. Also, the write buffer state must be monitored to wait for any posted write completion to the chip-select.


Copyright © 2011–2016, Texas Instruments Incorporated

Any access attempted to a nonvalid GPMC address region (CSVALID disabled or address decoding outside a valid chip-select region) is not propagated to the external interface and a GPMC access error is posted. In case of chip-selects overlapping, an error is generated and no access will occur on either chip-select. Chip-select 0 is the only chip-select region enabled after either a power-up or a GPMC reset.

Although the GPMC interface can drive up to six chip-selects, the frequency specified for this interface is for a specific load. If this load is exceeded, the maximum frequency cannot be reached. One solution is to implement a board with buffers, to allow the slowest device to maintain the total load on the lines.

#### 11.2.4.8.2 Access Protocol

##### 11.2.4.8.2.1 Supported Devices

The access protocol of each chip-select can be independently specified through the GPMC_CONFIG1_i[11-10] DEVICETYPE parameter (where i = 0 to 5) for:

- Random-access synchronous or asynchronous memory like NOR flash, SRAM
- NAND flash asynchronous devices

For more information about the NAND flash GPMC basic programming model and NAND support, see Section 11.2.4.12 and Section 11.2.4.12.1.

##### 11.2.4.8.2.2 Access Size Adaptation and Device Width

Each chip-select can be independently configured through the GPMC_CONFIG1_i[13-12] DEVICESIZE field (i = 0 to 5) to interface with a 16-bit wide device or an 8-bit wide device. System requests with data width greater than the external device data bus width are split into successive accesses according to both the external device data-bus width and little-endian data organization.

The device does not provide the A0 byte address line required for random-byte addressable 8-bit wide device interfacing (for both multiplexed and nonmultiplexed protocol). It limits the use of 8-bit wide device interfacing to byte-alias accesses. This limitation is not applicable to NAND device interfacing (8-bit wide or 16-bit wide devices).

##### 11.2.4.8.2.3 Address/Data-Multiplexing Interface

For random synchronous or asynchronous memory interfacing (DEVICETYPE = 00), an address- and data-multiplexing protocol can be selected through the GPMC_CONFIG1_i[[9-8] MUXADDDATA bit (i = 0 to 5). The $\overline{\text{ADV}}$ signal must be used as the external device address latch control signal. For the associated chip-select configuration, $\overline{\text{ADV}}$ assertion and deassertion time and $\overline{\text{OE}}$ assertion time must be set to the appropriate value to meet the address latch setup/hold time requirements of the external device (see Section 11.2.3).

This address/data-multiplexing interface is not applicable to NAND device interfacing. NAND devices require a specific address, command, and data multiplexing protocol (see Section 11.2.4.12).

#### 11.2.4.8.3 External Signals

##### 11.2.4.8.3.1 WAIT Pin Monitoring Control

GPMC access time can be dynamically controlled using an external gpmc_wait pin when the external device access time is not deterministic and cannot be defined and controlled only using the GPMC internal RDACCESSTIME, WRACCESSTIME and PAGEBURSTACCESSTIME wait state generator.

The GPMC features two input wait pin:gpmc_wait1, and gpmc_wait0. This pin allow control of external devices with different wait-pin polarity. They also allow the overlap of wait-pin assertion from different devices without affecting access to devices for which the wait pin is not asserted.

- The GPMC_CONFIG1_i[17-16] WAITPINSELECT bit (where i = 0 to 5) selects which input gpmc_wait pin is used for the device attached to the corresponding chip-select.
- The polarity of the wait pin is defined through the WAITxPINPOLARITY bit of the GPMC_CONFIG register. A wait pin configured to be active low means that low level on the WAIT signal indicates that the data is not ready and that the data bus is invalid. When WAIT is inactive, data is valid.


Copyright © 2011–2016, Texas Instruments Incorporated

The GPMC access engine can be configured per CS to monitor the wait pin of the external memory device or not, based on the access type: read or write.

- The GPMC_CONFIG1_i[22] WAITREADMONITORING bit defines whether the wait pin should be monitored during read accesses or not.
- The GPMC_CONFIG1_i[21] WAITWRITEMONITORING bit defines whether the wait pin should be monitored during write accesses or not.

The GPMC access engine can be configured to monitor the wait pin of the external memory device asynchronously or synchronously with the GPMC_CLK clock, depending on the access type: synchronous or asynchronous (the GPMC_CONFIG1_i[29] READTYPE and GPMC_CONFIG1_i[27] WRITETYPE bits).

#### *11.2.4.8.3.2 Wait Monitoring During an Asynchronous Read Access*

When wait-pin monitoring is enabled for read accesses (WAITREADMONITORING), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the wait-deasserted state.

During asynchronous read accesses with wait-pin monitoring enabled, the wait pin must be at a valid level (asserted or deasserted) for at least two GPMC clock cycles before RDACCESSTIME completes, to ensure correct dynamic access-time control through wait-pin monitoring. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

In this context, RDACCESSTIME is used as a WAIT invalid timing window and is set to such a value that the wait pin is at a valid state two GPMC clock cycles before RDACCESSTIME completes.

Similarly, during a multiple-access cycle (for example, asynchronous read page mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the wait-deasserted state. Wait-monitoring pipelining is also applicable to multiple accesses (access within a page).

- WAIT monitored as active freezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as asserted extends the current access time in the page. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as inactive completes the current access time and starts the next access phase in the page. The data bus is considered valid, and data are captured during this clock cycle. In case of a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their related control timing value and according to the CYCLETIME counter status.

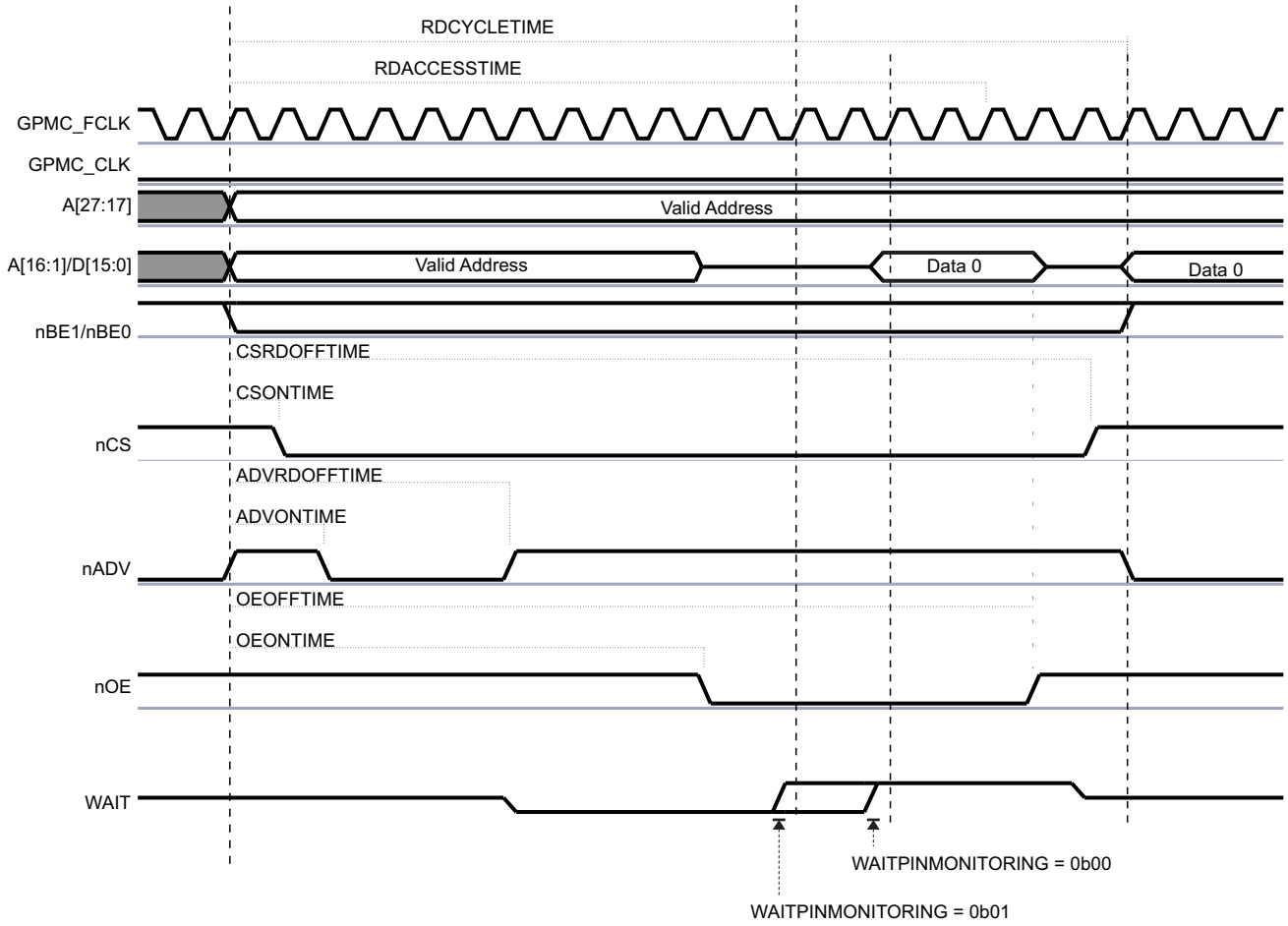
When a delay larger than two GPMC clocks must be observed between wait-pin deactivation time and data valid time (including the required GPMC and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data-capture time and the effective unlock of the CYCLETIME counter. This extra delay can be programmed in the GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME bit (i = 0 to 5).

- The WAITMONITORINGTIME parameter does not delay the wait-pin active or inactive detection, nor does it modify the two GPMC clocks pipelined detection delay.
- This extra delay is expressed as a number of GPMC_CLK clock cycles, even though the access is defined as asynchronous, and no GPMC_CLK clock is provided to the external device. Still, GPMCFCLKDIVIDER is used as a divider for the GPMC clock, so it must be programmed to define the correct WAITMONITORINGTIME delay.

Figure 11-7 shows wait behavior during an asynchronous single read access.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-7. Wait Behavior During an Asynchronous Single Read Access (GPMCFCLKDivider = 1)**




The WAIT signal is active low. GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME = 00b or 01b.

#### *11.2.4.8.3.3 Wait Monitoring During an Asynchronous Write Access*

When wait-pin monitoring is enabled for write accesses (GPMC_CONFIG1_i[21] WAITWRITEMONITORING bit = 1), the WAIT-invalid timing window is defined by the WRACCESSTIME field. WRACCESSTIME must be set so that the wait pin is at a valid state two GPMC clock cycles before WRACCESSTIME completes. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

- WAIT monitored as active freezes the CYCLETIME counter. This informs the GPMC that the data bus is not captured by the external device. The control signals are kept in their current state. The data bus still drives the data.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. This informs that the data bus is correctly captured by the external device. All signals, including the data bus, are controlled according to their related control timing value and to the CYCLETIME counter status.


Copyright © 2011–2016, Texas Instruments Incorporated

When a delay larger than two GPMC clock cycles must be observed between wait-pin deassertion time and the effective data write into the external device (including the required GPMC data setup time and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data write time into the external device and the effective unfreezing of the CYCLETIME counter. This extra delay can be programmed in the GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME bit (i = 0 to 5).

- The WAITMONITORINGTIME parameter does not delay the wait-pin assertion or deassertion detection, nor does it modify the two GPMC clock cycles pipelined detection delay.
- This extra delay is expressed as a number of GPMC_CLK clock cycles, even though the access is defined as asynchronous, and even though no clock is provided to the external device. Still, GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER is used as a divider for the GPMC clock and so it must be programmed to define the correct WAITMONITORINGTIME delay.

#### *11.2.4.8.3.4 Wait Monitoring During a Synchronous Read Access*

During synchronous accesses with wait-pin monitoring enabled, the wait pin is captured synchronously with GPMC_CLK, using the rising edge of this clock.

The WAIT signal can be programmed to apply to the same clock cycle it is captured in. Alternatively, it can be sampled one or two GPMC_CLK cycles ahead of the clock cycle it applies to. This pipelining is applicable to the entire burst access, and to all data phase in the burst access. This WAIT pipelining depth is programmed in the GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME bit (i = 0 to 5), and is expressed as a number of GPMC_CLK clock cycles.

In synchronous mode, when wait-pin monitoring is enabled (GPMC_CONFIG1_i[22] WAITREADMONITORING bit), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the WAIT deasserted-state detection.

Depending on the programmed WAITMONITORINGTIME value, the wait pin should be at a valid level, either asserted or deasserted:

- In the same clock cycle the data is valid if WAITMONITORINGTIME = 0 ( at RDACCESSTIME completion)
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC_FCLK clock cycles before RDACCESSTIME completion if WAITMONITORINGTIME not equal to 0

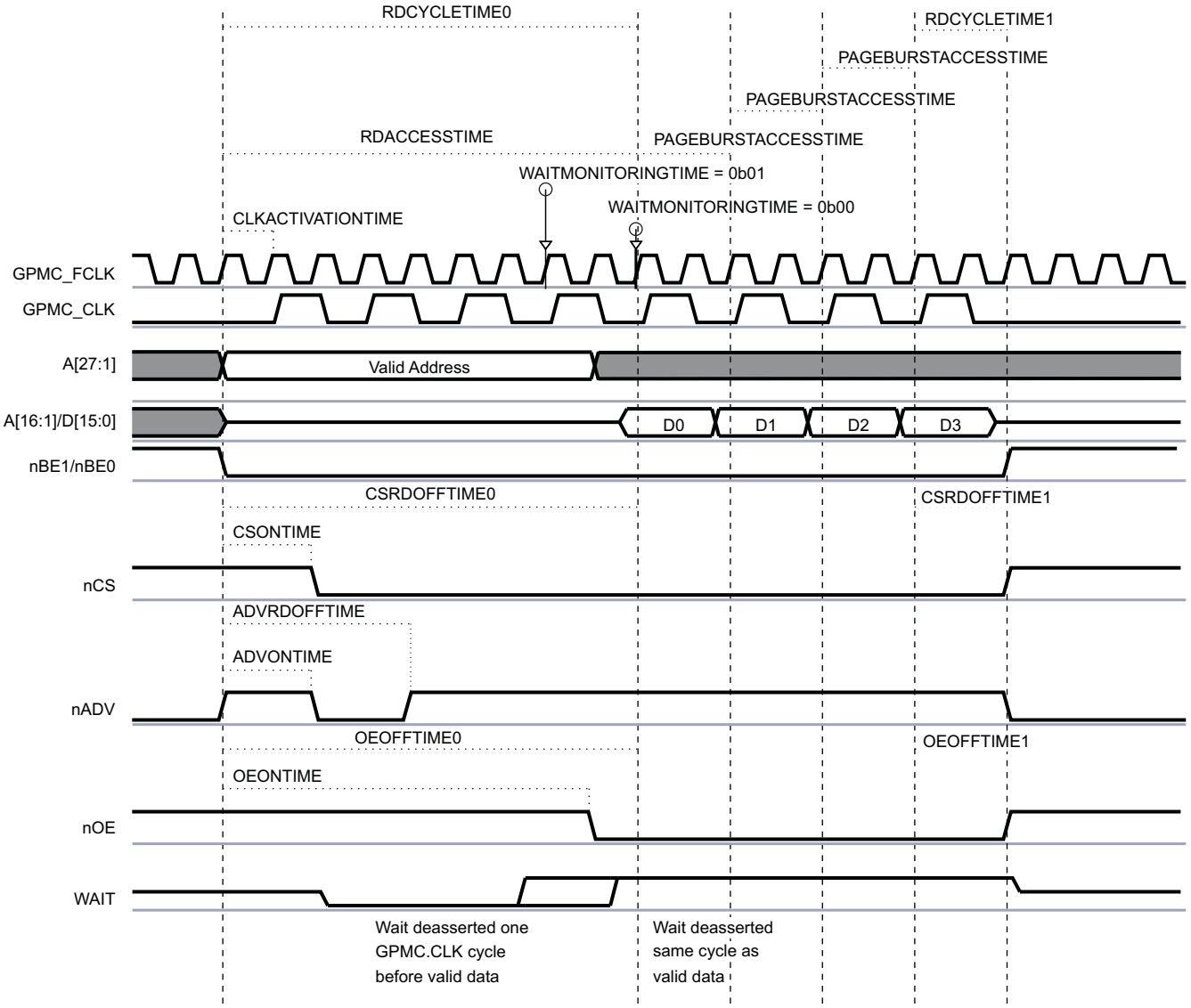
Similarly, during a multiple-access cycle (burst mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the wait-inactive state. The Wait pipelining depth programming applies to the whole burst access.

- WAIT monitored as active freezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in a lock state), WAIT monitored as active extends the current access time in the burst. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in lock state), WAIT monitored as inactive completes the current access time and starts the next access phase in the burst. The data bus is considered valid, and data are captured during this clock cycle. In a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their relative control timing value and the CYCLETIME counter status.

Figure 11-8 shows wait behavior during a synchronous read burst access.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-8. Wait Behavior During a Synchronous Read Burst Access**




The WAIT signal is active low. WAITMONITORINGTIME = 00b or 01b.

#### *11.2.4.8.3.5 Wait Monitoring During a Synchronous Write Access*

During synchronous accesses with wait-pin monitoring enabled (the WAITWRITEMONITORING bit), the wait pin is captured synchronously with GPMC_CLK, using the rising edge of this clock.

If enabled, external wait-pin monitoring can be used in combination with WRACCESSTIME to delay the effective memory device GPMC_CLK capture edge.

Wait-monitoring pipelining depth is similar to synchronous read access:

- At WRACCESSTIME completion if WAITMONITORINGTIME = 0
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC_FCLK cycles before WRACCESSTIME completion if WAITMONITORINGTIME not equal to 0.


Copyright © 2011–2016, Texas Instruments Incorporated

Wait-monitoring pipelining definition applies to whole burst accesses:

- WAIT monitored as active freezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as active indicates that the data bus is not being captured by the external device. Control signals are kept in their current state. The data bus is kept in its current state.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as inactive indicates the effective data capture of the bus by the external device and starts the next access of the burst. In case of a single access or if this was the last access in a multiple access cycle, all signals, including the data bus, are controlled according to their related control timing value and the CYCLETIME counter status.

Wait monitoring is supported for all configurations except for GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME = 0 (where i = 0 to 3) for write bursts with a clock divider of 1 or 2 (GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER field equal to 0 or 1, respectively).

#### 11.2.4.8.3.6 WAIT With NAND Device

For details about the use of the wait pin for communication with a NAND flash external device, see Section 11.2.4.12.2.

#### 11.2.4.8.3.7 Idle Cycle Control Between Successive Accesses

##### 11.2.4.8.3.7.1 Bus Turnaround (BUSTURNAROUND)

To prevent data-bus contention, an access that follows a read access to a slow memory/device must be delayed (in other words, control the $\overline{CS}$/$\overline{OE}$ de-assertion to data bus in high-impedance delay).

The bus turnaround is a time-out counter starting after $\overline{CS}$ or $\overline{OE}$ de-assertion time, whichever occurs first, and delays the next access start-cycle time. The counter is programmed through the GPMC_CONFIG6_i[3-0] BUSTURNAROUND bit (i = 0 to 5).

After a read access to a chip-select with a non zero BUSTURNAROUND, the next access is delayed until the BUSTURNAROUND delay completes, if the next access is one of the following:

- A write access to any chip-select (same or different from the chip-select data was read from)
- A read access to a different chip-select from the chip-select data was read access from
- A read or write access to a chip-select associated with an address/data-multiplexed device

Bus keeping starts after bus turnaround completion. The bus will not have enough time to go into high-impedance even though it could be driven with the same value before bus turnaround timing.

BUSTURNAROUND delay runs in parallel with GPMC_CONFIG6_i[3-0] CYCLE2CYCLEDELAY delays. It should be noted that BUSTURNAROUND is a timing parameter for the ending chip-select access while CYCLE2CYCLEDELAY is a timing parameter for the following chip-select access. The effective minimum delay between successive accesses is driven by these delay timing parameters and by the access type of the following access. See Figure 11-9 to Figure 11-11.

Another way to prevent bus contention is to define an earlier $\overline{CS}$ or $\overline{OE}$ deassertion time for slow devices or to extend the value of RDCYCLETIME. Doing this prevents bus contention, but affects all accesses of this specific chip-select.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-9. Read to Read for an Address-Data Multiplexed Device, On Different CS, Without Bus Turnaround (CS0 Attached to Fast Device)**

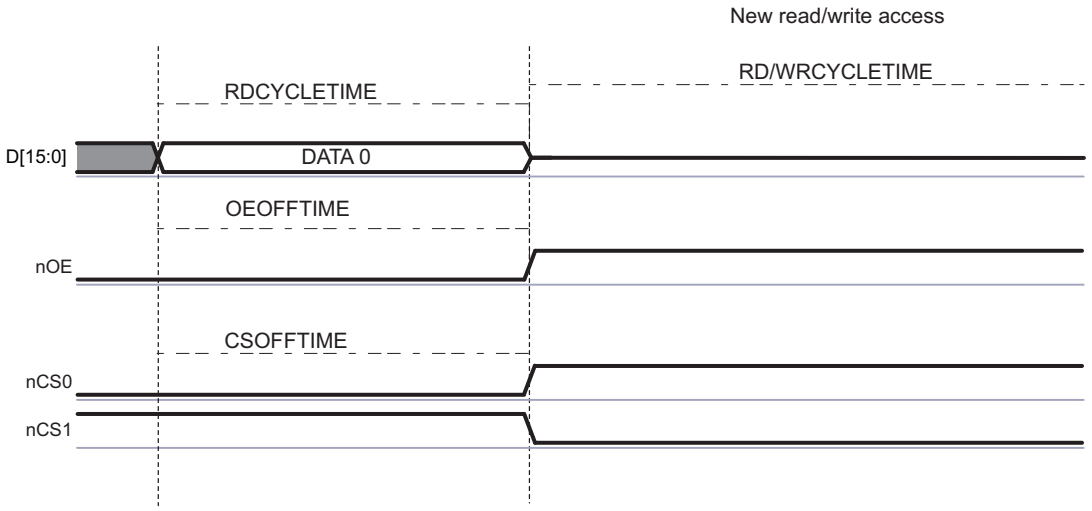
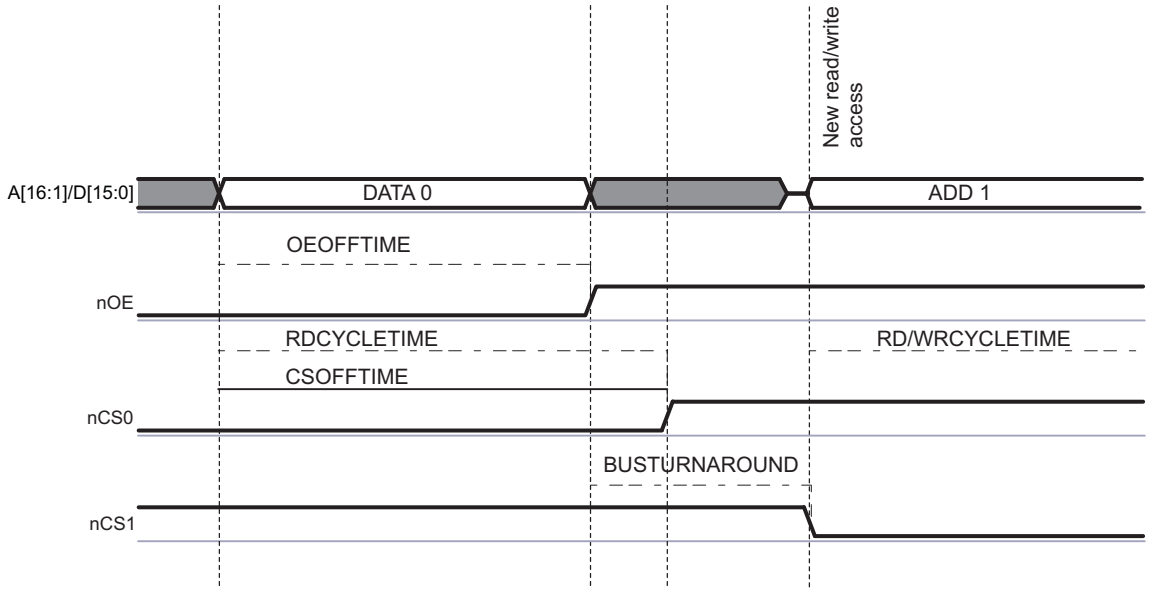


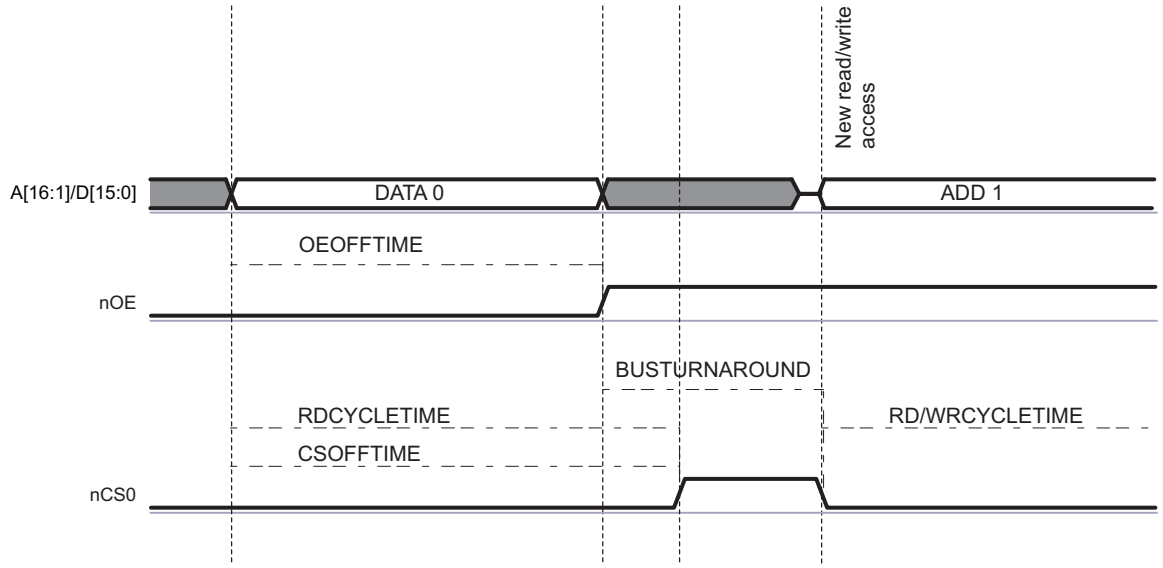

**Figure 11-10. Read to Read / Write for an Address-Data Multiplexed Device, On Different CS, With Bus Turnaround**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-11. Read to Read / Write for a Address-Data or AAD-Multiplexed Device, On Same CS, With Bus Turnaround**




#### *11.2.4.8.3.7.2 Idle Cycles Between Accesses to Same Chip-Select (CYCLE2CYCLESAMECSEN, CYCLE2CYCLEDELAY)*

Some devices require a minimum chip-select signal inactive time between accesses. The GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN bit (i = 0 to 5) enables insertion of a minimum number of GPMC_FCLK cycles, defined by the GPMC_CONFIG6_i[11-8] CYCLE2CYCLEDELAY field, between successive accesses of any type (read or write) to the same chip-select.

If CYCLE2CYCLESAMECSEN is enabled, any subsequent access to the same chip-select is delayed until its CYCLE2CYCLEDELAY completes. The CYCLE2CYCLEDELAY counter starts when CSRDOFFTIME/CSWROFFTIME completes.

The same applies to successive accesses occurring during 32-bit word or burst accesses split into successive single accesses when the single-access mode is used (GPMC_CONFIG1_i[30] READMULTIPLE = 0 or GPMC_CONFIG1_i[28] WRITEMULTIPLE = 0).

All control signals are kept in their default states during these idle GPMC_FCLK cycles. This prevents back-to-back accesses to the same chip-select without idle cycles between accesses.

#### *11.2.4.8.3.7.3 Idle Cycles Between Accesses to Different Chip-Select (CYCLE2CYCLEDIFFCSEN, CYCLE2CYCLEDELAY)*

Because of the pipelined behavior of the system, successive accesses to different chip-selects can occur back-to-back with no idle cycles between accesses. Depending on the control signals ($\overline{CS}$, $\overline{ADV}$_ALE, $\overline{BE0}$_CLE, $\overline{OE_RE}$, $\overline{WE}$) assertion and de-assertion timing parameters and on the IC timing parameters, some control signals assertion times may overlap between the successive accesses to different CS. Similarly, some control signals ($\overline{WE}$, $\overline{OE_RE}$) may not respect required transition times.

To work around the overlapping and to observe the required control-signal transitions, a minimum of CYCLE2CYCLEDELAY inactive cycles is inserted between the access being initiated to this chip-select and the previous access ending for a different chip-select. This applies to any type of access (read or write).

If GPMC_CONFIG6_i[6] CYCLE2CYCLEDIFFCSEN is enabled, the chip-select access is delayed until CYCLE2CYCLEDELAY cycles have expired since the end of a previous access to a different chip-select. CYCLE2CYCLEDELAY count starts at CSRDOFFTIME/CSWROFFTIME completion. All control signals are kept inactive during the idle GPMC_FCLK cycles.


Copyright © 2011–2016, Texas Instruments Incorporated

CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDIFFCSEN should be set in registers to respectively get idle cycles inserted between accesses on this chip-select and after accesses to a different chip-select.

The CYCLE2CYCLEDELAY delay runs in parallel with the BUSTURNAROUND delay. It should be noted that BUSTURNAROUND is a timing parameter defined for the ending chip-select access, whereas CYCLE2CYCLEDELAY is a timing parameter defined for the starting chip-select access. The effective minimum delay between successive accesses is based on the larger delay timing parameter and on access type combination, since bus turnaround does not apply to all access types. See Section 11.2.4.8.3.7.1 for more details on bus turnaround.

Table 11-10 describes the configuration required for idle cycle insertion.

**Table 11-10. Idle Cycle Insertion Configuration**

| First Access Type | BUSTURNAROUND Timing Parameter | Second Access Type | Chip-Select | Addr/Data Multiplexed | CYCLE2CYCLE SAMECSEN Parameter | CYCLE2CYCLE DIFFCSEN Parameter | Idle Cycle Insertion Between the Two Accesses |
|---|---|---|---|---|---|---|---|
| R/W | 0 | R/W | Any | Any | 0 | x | No idle cycles are inserted if the two accesses are well pipelined. |
| R | >0 | R | Same | Nonmuxed | x | 0 | No idle cycles are inserted if the two accesses are well pipelined. |
| R | >0 | R | Different | Nonmuxed | 0 | 0 | BUSTURNAROUND cycles are inserted. |
| R | >0 | R/W | Any | Muxed | 0 | 0 | BUSTURNAROUND cycles are inserted. |
| R | >0 | W | Any | Any | 0 | 0 | BUSTURNAROUND cycles are inserted. |
| W | >0 | R/W | Any | Any | 0 | 0 | No idle cycles are inserted if the two accesses are well pipelined. |
| R/W | 0 | R/W | Same | Any | 1 | x | CYCLE2CYCLEDELAY cycles are inserted. |
| R/W | 0 | R/W | Different | Any | x | 1 | CYCLE2CYCLEDELAY cycles are inserted. |
| R/W | >0 | R/W | Same | Any | 1 | x | CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is max (BUSTURNAROUND, CYCLE2CYCLEDELAY). |
| R/W | >0 | R/W | Different | Any | x | 1 | CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is maximum (BUSTURNAROUND, CYCLE2CYCLEDELAY). |

#### *11.2.4.8.3.8 Slow Device Support (TIMEPARAGRANULARITY Parameter)*

All access-timing parameters can be multiplied by 2 by setting the GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit (where i = 0 to 5). Increasing all access timing parameters allows support of slow devices.

#### *11.2.4.8.3.9 Reset*

No reset signal is sent to the external memory device by the GPMC. For more information about external-device reset, see Power, Reset, and Clock Management.


Copyright © 2011–2016, Texas Instruments Incorporated

The PRCM module provides an input pin, global_rst_n, to the GPMC:

- The global_rst_n pin is activated during device warm reset and cold reset.
- The global_rst_n pin initializes the internal state-machine and the internal configuration registers.

#### *11.2.4.8.3.10 Byte Enable ($\overline{BE1}$/$\overline{BE0}$)*

Byte enable signals ($\overline{BE1}$/$\overline{BE0}$) are:

- Valid (asserted or nonasserted according to the incoming system request) from access start to access completion for asynchronous and synchronous single accesses
- Asserted low from access start to access completion for asynchronous and synchronous multiple read accesses
- Valid (asserted or nonasserted, according to the incoming system request) synchronously to each written data for synchronous multiple write accesses

#### *11.2.4.8.4 Error Handling*

When an error occurs in the GPMC, the error information is stored in the GPMC_ERR_TYPE register and the address of the illegal access is stored in the GPMC_ERR_ADDRESS register. The GPMC keeps only the first error abort information until the GPMC_ERR_TYPE register is reset. Subsequent accesses that cause errors are not logged until the error is cleared by hardware with the GPMC_ERR_TYPE[0]ERRORVALID bit.

- ERRORNOTSUPPADD occurs when an incoming system request address decoding does not match any valid chip-select region, or if two chip-select regions are defined as overlapped, or if a register file access is tried outside the valid address range of 1KB.
- ERRORNOTSUPPMCMD occurs when an unsupported command request is decoded at the L3 Slow interconnect interface
- ERRORTIMEOUT: A time-out mechanism prevents the system from hanging. The start value of the

9-bit time-out counter is defined in the GPMC_TIMEOUT_CONTROL register and enabled with the GPMC_TIMEOUT_CONTROL[0] TIMEOUTENABLE bit. When enabled, the counter starts at start-cycle time until it reaches 0 and data is not responded to from memory, and then a time-out error occurs. When data are sent from memory, this counter is reset to its start value. With multiple accesses (asynchronous page mode or synchronous burst mode), the counter is reset to its start value for each data access within the burst.

The GPMC does not generate interrupts on these errors. True abort to the MPU or interrupt generation is handled at the interconnect level.

### 11.2.4.9 Timing Setting

The GPMC offers the maximum flexibility to support various access protocols. Most of the timing parameters of the protocol access used by the GPMC to communicate with attached memories or devices are programmable on a chip-select basis. Assertion and deassertion times of control signals are defined to match the attached memory or device timing specifications and to get maximum performance during accesses. For more information on GPMC_CLK and GPMC_FCLK see Section 11.2.4.9.6.

In the following sections, the start access time refer to the time at which the access begins.

#### *11.2.4.9.1 Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME)*

The GPMC_CONFIG5_i[4-0] RDCYCLETIME and GPMC_CONFIG5_i[12-8] WRCYCLETIME bit (i = 0 to 5) define the address bus and byte enables valid times for read and write accesses. To ensure a correct duty cycle of GPMC_CLK between accesses, RDCYCLETIME and WRCYCLETIME are expressed in GPMC_FCLK cycles and must be multiples of the GPMC_CLK cycle. RDCYCLETIME and WRCYCLETIME bit fields can be set with a granularity of 1 or 2 throught GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY.

When either RDCYCLETIME or WRCYCLETIME completes, if they are not already deasserted, all control signals ($\overline{CS}$, $\overline{ADV}$_ALE, $\overline{OE}$_$\overline{RE}$, $\overline{WE}$, and $\overline{BE0}$_CLE) are deasserted to their reset values, regardless of their deassertion time parameters.


Copyright © 2011–2016, Texas Instruments Incorporated

An exception to this forced deassertion occurs when a pipelined request to the same chip-select or to a different chip-select is pending. In such a case, it is not necessary to deassert a control signal with deassertion time parameters equal to the cycle-time parameter. This exception to forced deassertion prevents any unnecessary glitches. This requirement also applies to BE signals, thus avoiding an unnecessary BE glitch transition when pipelining requests.

If no inactive cycles are required between successive accesses to the same or to a different chip-select (GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN = 0 or GPMC_CONFIG6_i[6] CYCLE2CYCLEDIFFCSEN = 0, where i = 0 to 3), and if assertion-time parameters associated with the pipelined access are equal to 0, asserted control signals ($\overline{CS}$, $\overline{ADV}$_ALE, $\overline{BE0}$_CLE, $\overline{WE}$, and $\overline{OE_RE}$) are kept asserted. This applies to any read/write to read/write access combination.

If inactive cycles are inserted between successive accesses, that is, CYCLE2CYCLESAMECSEN = 1 or CYCLE2CYCLEDIFFCSEN = 1, the control signals are forced to their respective default reset values for the number of GPMC_FCLK cycles defined in CYCLE2CYCLEDELAY.

#### 11.2.4.9.2 $\overline{CS}$: Chip-Select Signal Control Assertion/Deassertion Time (CSONTIME / CSRDOFFTIME / CSWROFFTIME / CSEXTRADELAY)

The GPMC_CONFIG2_i[3-0] CSONTIME field (where i = 0 to 5) defines the $\overline{CS}$ signal-assertion time relative to the start access time. It is common for read and write accesses.

The GPMC_CONFIG2_i[12-8] CSRDOFFTIME (read access) and GPMC_CONFIG2_i[20-16] CSWROFFTIME (write access) bit fields define the $\overline{CS}$ signal deassertion time relative to start access time.

CSONTIME, CSRDOFFTIME and CSWROFFTIME parameters are applicable to synchronous and asynchronous modes. CSONTIME can be used to control an address and byte enable setup time before chip-select assertion. CSRDOFFTIME and CSWROFFTIME can be used to control an address and byte enable hold time after chip-select deassertion.

$\overline{CS}$ signal transitions as controlled through CSONTIME, CSRDOFFTIME, and CSWROFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG2_i[7] CSEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on the $\overline{CS}$ assertion and deassertion time to guarantee proper setup and hold time relative to GPMC_CLK. CSEXTRADELAY is especially useful in configurations where GPMC_CLK and GPMC_FCLK have the same frequency, but can be used for all GPMC configurations. If enabled, CSEXTRADELAY applies to all parameters controlling $\overline{CS}$ transitions.

The CSEXTRADELAY bit must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than the $\overline{CS}$ signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

#### 11.2.4.9.3 $\overline{ADV}$_ALE: Address Valid/Address Latch Enable Signal Control Assertion/Deassertion Time (ADVONTIME / ADVRDOFFTIME / ADVWROFFTIME / ADVEXTRADELAY / ADVAADMUXONTIME / ADVAADMUXRDOFFTIME / ADVAADMUXWROFFTIME)

The GPMC_CONFIG3_i[3-0] ADVONTIME field (where i = 0 to 5) defines the $\overline{ADV}$_ALE signal-assertion time relative to start access time. It is common to read and write accesses.

The GPMC_CONFIG3_i[12-8] ADVRDOFFTIME (read access) and GPMC_CONFIG3_i[20-16] ADVWROFFTIME (write access) bit fields define the $\overline{ADV}$_ALE signal-deassertion time relative to start access time.

ADVONTIME can be used to control an address and byte enable valid setup time control before $\overline{ADV}$_ALE assertion. ADVRDOFFTIME and ADVWROFFTIME can be used to control an address and byte enable valid hold time control after $\overline{ADV}$_ALE de-assertion. ADVRDOFFTIME and ADVWROFFTIME are applicable to both synchronous and asynchronous modes.


Copyright © 2011–2016, Texas Instruments Incorporated

$\overline{\text{ADV}}$_ALE signal transitions as controlled through ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG3_i[7] ADVEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on $\overline{\text{ADV}}$_ALE assertion and deassertion time to assure proper setup and hold time relative to GPMC_CLK. The ADVEXTRADELAY configuration parameter is especially useful in configurations where GPMC_CLK and GPMC_FCLK have the same frequency, but can be used for all GPMC configurations. If enabled, ADVEXTRADELAY applies to all parameters controlling $\overline{\text{ADV}}$_ALE transitions.

ADVEXTRADELAY must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than $\overline{\text{ADV}}$_ALE signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

The GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME, GPMC_CONFIG3_i[26-24] ADVAADMUXRDOFFTIME, and GPMC_CONFIG3_i[30-28] ADVAADMUXWROFFTIME parameters have the same functions as ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME, but apply to the first address phase in the AAD-multiplexed protocol. It is the user responsibility to make sure ADVAADMUXxxOFFTIME is programmed to a value lower than or equal to ADVxxOFFTIME. Functionality in AAD-mux mode is undefined if the settings do not comply with this requirement. ADVAADMUXxxOFFTIME can be programmed to the same value as ADVONTIME if no high $\overline{\text{ADV}}$ pulse is needed between the two AAD-mux address phases, which is the typical case in synchronous mode. In this configuration, $\overline{\text{ADV}}$ is kept low until it reaches the correct ADVxxOFFTIME.

See Section 11.2.4.12 for more details on ADVONTIME, ADVRDOFFTIME, ADVWROFFTIME, and ADVAADMUXRDOFFTIME, ADVAADMUXWROFFTIME usage for CLE and ALE (Command / Address Latch Enable) usage for a NAND Flash interface.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.9.4 $\overline{OE_RE}$: Output Enable / Read Enable Signal Control Assertion / Deassertion Time (OEONTIME / OEOFFTIME / OEEXTRADELAY / OEAADMUXONTIME / OEAADMUXOFFTIME)

The GPMC_CONFIG4_i[3-0] OEONTIME bit (where i = 0 to 5) defines the $\overline{OE_RE}$ signal assertion time relative to start access time. It is applicable only to read accesses.

The GPMC_CONFIG4_i[12-8] OEOFFTIME field defines the $\overline{OE_RE}$ signal deassertion time relative to start access time. It is applicable only to read accesses. $\overline{OE_RE}$ is not asserted during a write cycle.

OEONTIME, OEOFFTIME, OEAADMUXONTIME and OEAADMUXOFFTIME parameters are applicable to synchronous and asynchronous modes. OEONTIME can be used to control an address and byte enable valid setup time control before $\overline{OE_RE}$ assertion. OEOFFTIME can be used to control an address and byte enable valid hold time control after $\overline{OE_RE}$ assertion.

OEAADMUXONTIME and OEAADMUXOFFTIME parameters have the same functions as OEONTIME and OEOFFTIME, but apply to the first OE assertion in the AAD-multiplexed protocol for a read phase, or to the only OE assertion for a write phase. It is the user responsibility to make sure OEAADMUXOFFTIME is programmed to a value lower than OEONTIME. Functionality in AAD-mux mode is undefined if the settings do not comply with this requirement. OEAADMUXOFFTIME shall never be equal to OEONTIME because the AAD-mux protocol requires a second address phase with the $\overline{OE}$ signal de-asserted before $\overline{OE}$ can be asserted again to define a read command.

The $\overline{OE_RE}$ signal transitions as controlled through OEONTIME, OEOFFTIME, OEAADMUXONTIME and OEAADMUXOFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG4_i[7] OEEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on $\overline{OE_RE}$ assertion and deassertion time to assure proper setup and hold time relative to GPMC_CLK. If enabled, OEEXTRADELAY applies to all parameters controlling $\overline{OE_RE}$ transitions.

OEEXTRADELAY must be used carefully, to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program RDCYCLETIME and WRCYCLETIME to be greater than $\overline{OE_RE}$ signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

When the GPMC generates a read access to an address-/data-multiplexed device, it drives the address bus until $\overline{OE}$ assertion time.

#### 11.2.4.9.5 $\overline{WE}$: Write Enable Signal Control Assertion / Deassertion Time (WEONTIME / WEOFFTIME / WEEXTRADELAY)

The GPMC_CONFIG4_i[19-16] WEONTIME field (where i = 0 to 3) defines the $\overline{WE}$ signal-assertion time relative to start access time. The GPMC_CONFIG4_i[28-24] WEOFFTIME field defines the $\overline{WE}$ signal-deassertion time relative to start access time. These bit fields only apply to write accesses. $\overline{WE}$ is not asserted during a read cycle.

WEONTIME can be used to control an address and byte enable valid setup time control before $\overline{WE}$ assertion. WEOFFTIME can be used to control an address and byte enable valid hold time control after $\overline{WE}$ assertion.

$\overline{WE}$ signal transitions as controlled through WEONTIME, and WEOFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG4_i[23] WEEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on $\overline{WE}$ assertion and deassertion time to guaranty proper setup and hold time relative to GPMC_CLK. If enabled, WEEXTRADELAY applies to all parameters controlling $\overline{WE}$ transitions.

The WEEXTRADELAY bit must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the WRCYCLETIME bit field to be greater than the $\overline{WE}$ signal-deassertion time, including the extra half-GPMC_FCLK-period delay.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.9.6 GPMC_CLK

GPMC_CLK is the external clock provided to the attached synchronous memory or device.

- The GPMC_CLK clock frequency is the GPMC_FCLK functional clock frequency divided by 1, 2, 4, or 8, depending on the GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER bit (where i = 0 to 3), with an assured 50-percent duty cycle.
- The GPMC_CLK clock is only activated when the access in progress is defined as synchronous (read or write access).
- The GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME bit (i = 0 to 5) defines the number of GPMC_FCLK cycles from start access time to GPMC_CLK activation.
- The GPMC_CLK clock is stopped when cycle time completes and is asserted low between accesses.
- The GPMC_CLK clock is kept low when access is defined as asynchronous.

When cycle time completes, the GPMC_CLK may be high because of the GPMCFCLKDIVIDER bit field. To ensure correct stoppage of the GPMC_CLK clock within the 50-percent required duty cycle, it is the user's responsibility to extend the RDCYCLETIME or WRCYCLETIME value.

To ensure a correct external clock cycle, the following rules must be applied:

- (RDCYCLETIME - CLKACTIVATIONTIME) must be a multiple of (GPMCFCLKDIVIDER + 1).
- The PAGEBURSTACCESSTIME value must be a multiple of (GPMCFCLKDIVIDER + 1).

#### 11.2.4.9.7 GPMC_CLK and Control Signals Setup and Hold

Control-signal transition (assertion and deassertion) setup and hold values with respect to the GPMC_CLK edge can be controlled in the following ways:

- For the GPMC_CLK signal, the GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME bit (i = 0 to 5) allows setup and hold control of control-signal assertion time.
- The use of a divided GPMC_CLK allows setup and hold control of control-signal assertion and deassertion times.
- When GPMC_CLK runs at the GPMC_FCLK frequency so that GPMC_CLK edge and control-signal transitions refer to the same GPMC_FCLK edge, the control-signal transitions can be delayed by half of a GPMC_FCLK period to provide minimum setup and hold times. This half-GPMC_FCLK delay is enabled with the CSEXTRADELAY, ADVEXTRADELAY, OEEXTRADELAY, or WEEXTRADELAY parameter. This delay must be used carefully to prevent control-signal overlap between successive accesses to different chip-selects. This implies that the RDCYCLETIME and WRCYCLETIME are greater than the last control-signal deassertion time, including the extra half-GPMC_FCLK cycle.

#### 11.2.4.9.8 Access Time (RDACCESSTIME / WRACCESSTIME)

The read access time and write access time durations can be programmed independently through GPMC_CONFIG5_i[20-16] RDACCESSTIME and GPMC_CONFIG6_i[28-24] WRACCESSTIME bit (i = 0 to 5). This allows $\overline{OE}$ and GPMC data capture timing parameters to be independent of $\overline{WE}$ and memory device data capture timing parameters. RDACCESSTIME and WRACCESSTIME bit fields can be set with a granularity of 1 or 2 throught GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY.

##### 11.2.4.9.8.1 Access Time on Read Access

In asynchronous read mode, for single and paged accesses, GPMC_CONFIG5_i[[20-16] RDACCESSTIME bit (i = 0 to 5) defines the number of GPMC_FCLK cycles from start access time to the GPMC_FCLK rising edge used for the first data capture. RDACCESSTIME must be programmed to the rounded greater value (in GPMC_FCLK cycles) of the read access time of the attached memory device.

In synchronous read mode, for single or burst accesses, RDACCESSTIME defines the number of GPMC_FCLK cycles from start access time to the GPMC_FCLK rising edge corresponding to the GPMC_CLK rising edge used for the first data capture.


Copyright © 2011–2016, Texas Instruments Incorporated

GPMC_CLK which is sent to the memory device for synchronization with the GPMC controller, is internally retimed to correctly latch the returned data. GPMC_CONFIG5_i[4-0] RDCYCLETIME must be greater than RDACCESSTIME in order to let the GPMC latch the last return data using the internally retimed GPMC_CLK.

The external WAIT signal can be used in conjunction with RDACCESSTIME to control the effective GPMC data-capture GPMC_FCLK edge on read access in both asynchronous mode and synchronous mode. For details about wait monitoring, see Section 11.2.4.8.1.

#### 11.2.4.9.8.2 Access Time on Write Access

In asynchronous write mode, the GPMC_CONFIG6_i[[28-24] WRACCESSTIME timing parameter is not used to define the effective write access time. Instead, it is used as a WAIT invalid timing window, and must be set to a correct value so that the gpmc_wait pin is at a valid state two GPMC_CLK cycles before WRACCESSTIME completes. For details about wait monitoring, see Section 11.2.4.8.1.

In synchronous write mode , for single or burst accesses, WRACCESSTIME defines the number of GPMC_FCLK cycles from start access time to the GPMC_CLK rising edge used by the memory device for the first data capture.

The external WAIT signal can be used in conjunction with WRACCESSTIME to control the effective memory device data capture GPMC_CLK edge for a synchronous write access. For details about wait monitoring, see Section 11.2.4.8.1.

### 11.2.4.9.9 Page Burst Access Time (PAGEBURSTACCESSTIME)

GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME bit (i = 0 to 5) can be set with a granularity of 1 or 2 throught the GPMC_CONFIG1_i[[4] TIMEPARAGRANULARITY.

#### 11.2.4.9.9.1 Page Burst Access Time on Read Access

In asynchronous page read mode, the delay between successive word captures in a page is controlled through the PAGEBURSTACCESSTIME bit field. The PAGEBURSTACCESSTIME parameter must be programmed to the rounded greater value (in GPMC_FCLK cycles) of the read access time of the attached device.

In synchronous burst read mode, the delay between successive word captures in a burst is controlled through the PAGEBURSTACCESSTIME field.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective GPMC data capture GPMC_FCLK edge on read access. For details about wait monitoring, see Section 11.2.4.8.1.

#### 11.2.4.9.9.2 Page Burst Access Time on Write Access

Asynchronous page write mode is not supported. PAGEBURSTACCESSTIME is irrelevant in this case.

In synchronous burst write mode, PAGEBURSTACCESSTIME controls the delay between successive memory device word captures in a burst.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective memory-device data capture GPMC_CLK edge in synchronous write mode. For details about wait monitoring, see Section 11.2.4.8.1.

### 11.2.4.9.10 Bus Keeping Support

At the end-cycle time of a read access, if no other access is pending, the GPMC drives the bus with the last data read after RDCYCLETIME completion time to prevent bus floating and reduce power consumption.

After a write access, if no other access is pending, the GPMC keeps driving the data bus after WRCYCLETIME completes with the same data to prevent bus floating and power consumption.


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.2.4.10 NOR Access Description

For each chip-select configuration, the read access can be specified as either asynchronous or synchronous access through the GPMC_CONFIG1_i[29] READTYPE bit (i = 0 to 5). For each chip-select configuration, the write access can be specified as either synchronous or asynchronous access through the GPMC_CONFIG1_i[27] WRITETYPE bit (i = 0 to 5).

Asynchronous and synchronous read and write access time and related control signals are controlled through timing parameters that refer to GPMC_FCLK. The primary difference of synchronous mode is the availability of a configurable clock interface (GPMC_CLK) to control the external device. Synchronous mode also affects data-capture and wait-pin monitoring schemes in read access.

For details about asynchronous and synchronous access, see the descriptions of GPMC_CLK, RdAccessTime, WrAccessTime, and wait-pin monitoring.

For more information about timing-parameter settings, see the sample timing diagrams in this chapter.

The address bus and $\overline{\text{BE}}$[1:0] are fixed for the duration of a synchronous burst read access, but they are updated for each beat of an asynchronous page-read access.

#### *11.2.4.10.1 Asynchronous Access Description*

This section describes:

- Asynchronous single read operation on an address/data multiplexed device
- Asynchronous single write operation on an address/data-multiplexed device
- Asynchronous single read operation on an AAD-multiplexed device
- Asynchronous single write operation on an AAD-multiplexed device
- Asynchronous multiple (page) read operation on a non-multiplexed device

In asynchronous operations GPMC_CLK is not provided outside the GPMC and is kept low.

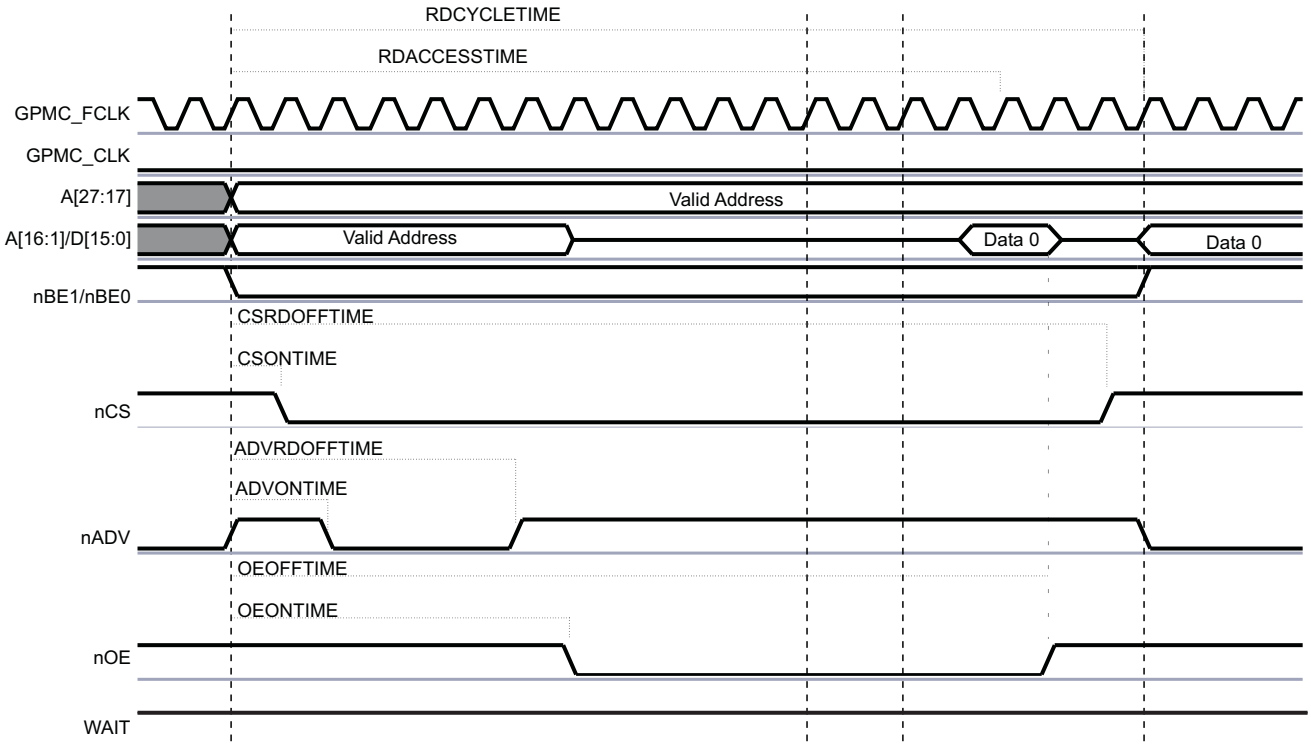

Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.1.1 Access on Address/Data Multiplexed Devices

##### 11.2.4.10.1.1.1 Asynchronous Single-Read Operation on an Address/Data Multiplexed Device

Figure 11-12 shows an asynchronous single read operation on an address/data-multiplexed device.

**Figure 11-12. Asynchronous Single Read Operation on an Address/Data Multiplexed Device**




##### 11.2.4.10.1.1.2 Asynchronous Single Read on an Address/Data-Multiplexed Device

See Section 11.3.6.1 for formulas to calculate timing parameters.

Table 11-40 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until $\overline{OE}$ assertion time. For details, see Section 11.2.4.8.2.3.

Address bits (A[16:1] from a GPMC perspective, A[15:0] from an external device perspective) are placed on the address/data bus, and the remaining address bits GPMC_A[25:16] are placed on the address bus.

- Chip-select signal $\overline{CS}$
  - $\overline{CS}$ assertion time is controlled by the GPMC_CONFIG2_i[3-0] CSONTIME field. It controls the address setup time to $\overline{CS}$ assertion.
  - $\overline{CS}$ deassertion time is controlled by the GPMC_CONFIG2_i[12-8] CSRDOFFTIME field. It controls the address hold time from $\overline{CS}$ deassertion
- Address valid signal $\overline{ADV}$
  - $\overline{ADV}$ assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ deassertion time is controlled by the GPMC_CONFIG3_i[[12-8] ADVRDOFFTIME field.


Copyright © 2011–2016, Texas Instruments Incorporated

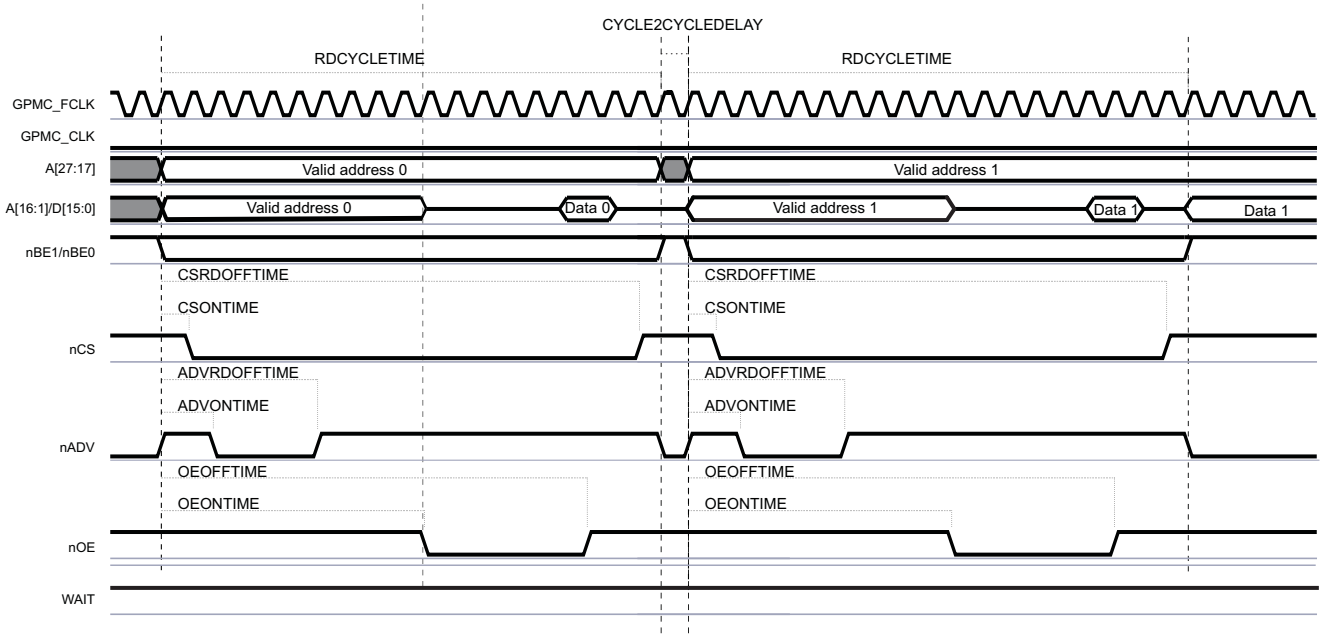
- Output enable signal $\overline{OE}$
  - $\overline{OE}$ assertion indicates a read cycle.
  - $\overline{OE}$ assertion time is controlled by the GPMC_CONFIG4_i[[3-0] OEONTIME field.
  - OE deassertion time is controlled by the GPMC_CONFIG4_i[[12-8] OEOFFTIME field.
- Read data is latched when RDACCESSTIME completes. Access time is defined in the GPMC_CONFIG5_i[20-16] RDACCESSTIME field.
- The end of the access is defined by the GPMC_CONFIG5_i[4-0] RDCYCLETIME parameter.

In the GPMC, when a 16-bit wide device is attached to the controller, a 32-bit word write access is split into two 16-bit word write accesses. For more information about GPMC access size and type adaptation, see Section 11.2.4.10.5. Between two successive accesses, if an $\overline{CS}$ pulse is needed:

- The GPMC_CONFIG6_i[[11-8] CYCLE2CYCLEDELAY field can be programmed with GPMC_CONFIG6_i[[7] CYCLE2CYCLESAMECSEN enabled.
- The CSWROFFTIME and CSONTIME parameters also allow a chip-select pulse, but this affects all other types of access.

Figure 11-13 shows two asynchronous single-read accesses on an address/data-multiplexed devices.

**Figure 11-13. Two Asynchronous Single Read Accesses on an Address/Data Multiplexed Device (32-Bit Read Split Into 2 × 16-Bit Read)**



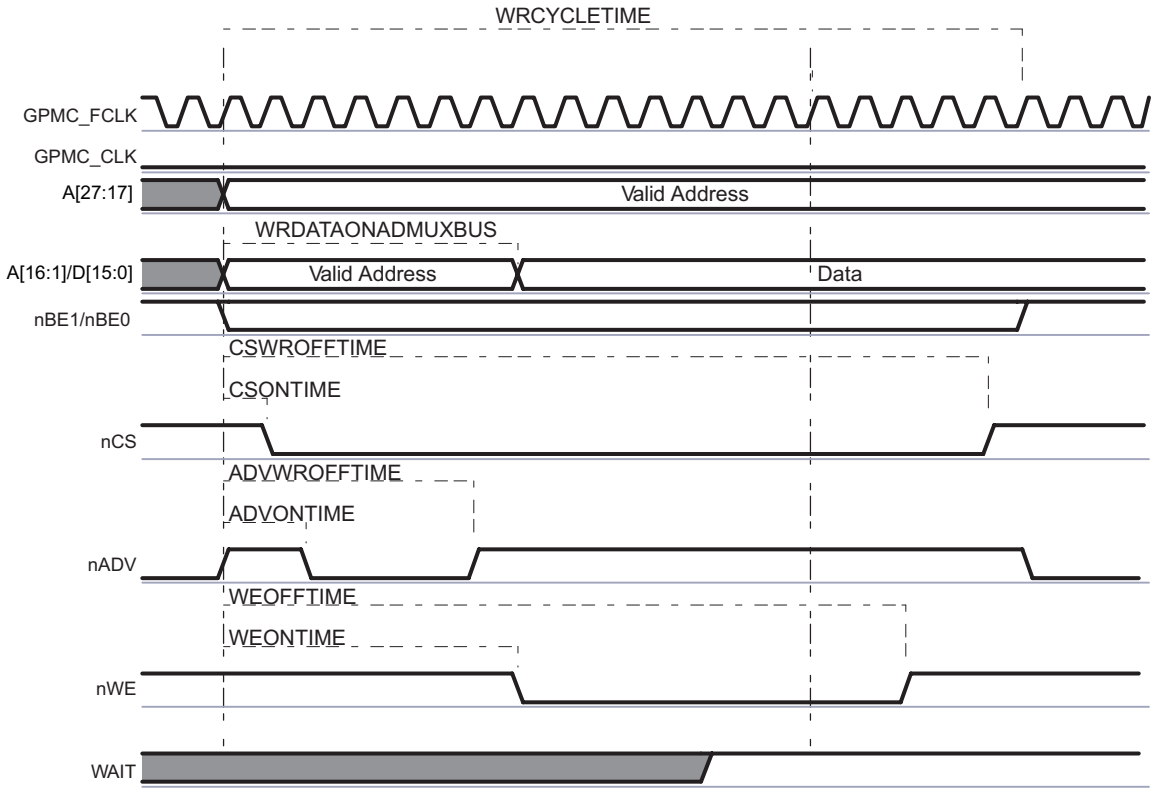


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.1.1.3 Asynchronous Single Write Operation on an Address/Data-Multiplexed Device

Figure 11-14 shows an asynchronous single write operation on an address/data-multiplexed device.

**Figure 11-14. Asynchronous Single Write on an Address/Data-Multiplexed Device**




#### 11.2.4.10.1.1.4 Asynchronous Single Write on an Address/Data-Multiplexed Device

See Section 11.3.6.1 for formulas to calculate timing parameters.

Table 11-40 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single write mode. When the GPMC generates a write access to an address/data-multiplexed device, it drives the address bus until $\overline{WE}$ assertion time. For more information, see Section 11.2.4.8.2.3.

The $\overline{CS}$ and $\overline{ADV}$ signals are controlled in the same way as for asynchronous single read operation on an address/data-multiplexed device.

- Write enable signal $\overline{WE}$
  - $\overline{WE}$ assertion indicates a write cycle.
  - $\overline{WE}$ assertion time is controlled by the GPMC_CONFIG4_i[19-16] WEONTIME field.
  - $\overline{WE}$ deassertion time is controlled by the GPMC_CONFIG4_i[28-24] WEOFFTIME field.
- The end of the access is defined by the GPMC_CONFIG5_i[12-8] WRCYCLETIME parameter.

Address bits A[16:1] (GPMC point of view) are placed on the address/data bus at the start of cycle time, and the remaining address bits A[26:17] are placed on the address bus.

Data is driven on the address/data bus at a GPMC_CONFIG6_i[19-16] WRDATAONADMUXBUS time.

Write multiple access in asynchronous mode is not supported. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.

After a write operation, if no other access (read or write) is pending, the data bus keeps its previous value. See Section 11.2.4.9.10.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.1.1.5 Asynchronous Multiple (Page) Write Operation on an Address/Data-Multiplexed Device

Write multiple (page) access in asynchronous mode is not supported for address/data-multiplexed devices. If GPMC_CONFIG1_i[28] WRITEMULTIPLE is enabled (1) with GPMC_CONFIG1_i[27] WRITETYPE as asynchronous (0), the GPMC processes single asynchronous accesses.

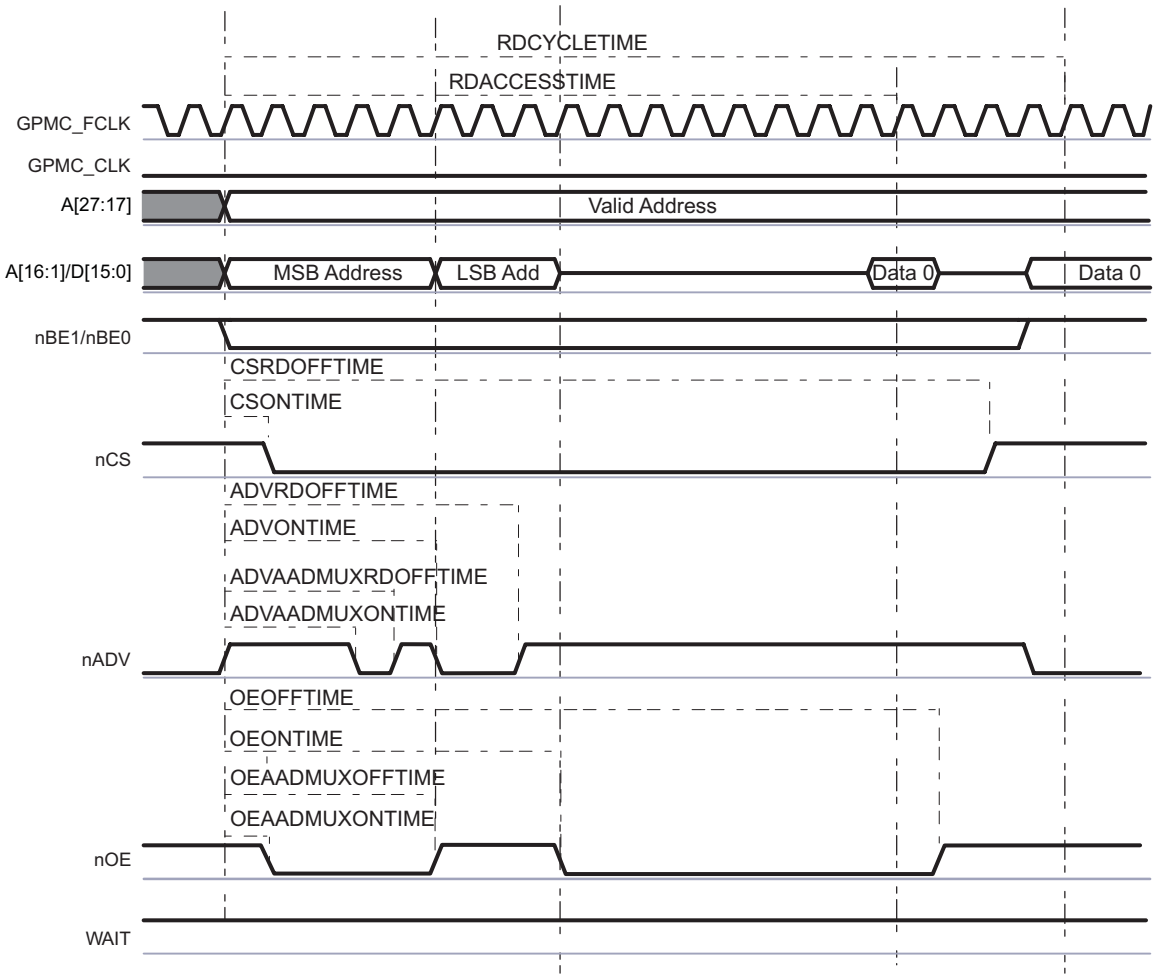
For accesses on non-multiplexed devices, see Section 11.2.4.10.3.

### 11.2.4.10.1.2 Access on Address/Address/Data (AAD) Multiplexed Devices

#### 11.2.4.10.1.2.1 Asynchronous Single Read Operation on an AAD-Multiplexed Device

Figure 11-15 shows an asynchronous single read operation on an AAD-multiplexed device.

**Figure 11-15. Asynchronous Single-Read on an AAD-Multiplexed Device**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.1.2.2 Asynchronous Single Read on an AAD-Multiplexed Device

See Section 11.3.6.1 for formulas to calculate timing parameters.

Table 11-40 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single write mode.

When the GPMC generates a read access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with $\overline{OE}$ driven low. The first address phase ends at the first $\overline{OE}$ deassertion time. The second phase for LSB address is qualified with $\overline{OE}$ driven high. The second address phase ends at the second $\overline{OE}$ assertion time.

The $\overline{CS}$ signal is controlled in the same way as for asynchronous single read operation on an address/data-multiplexed device.

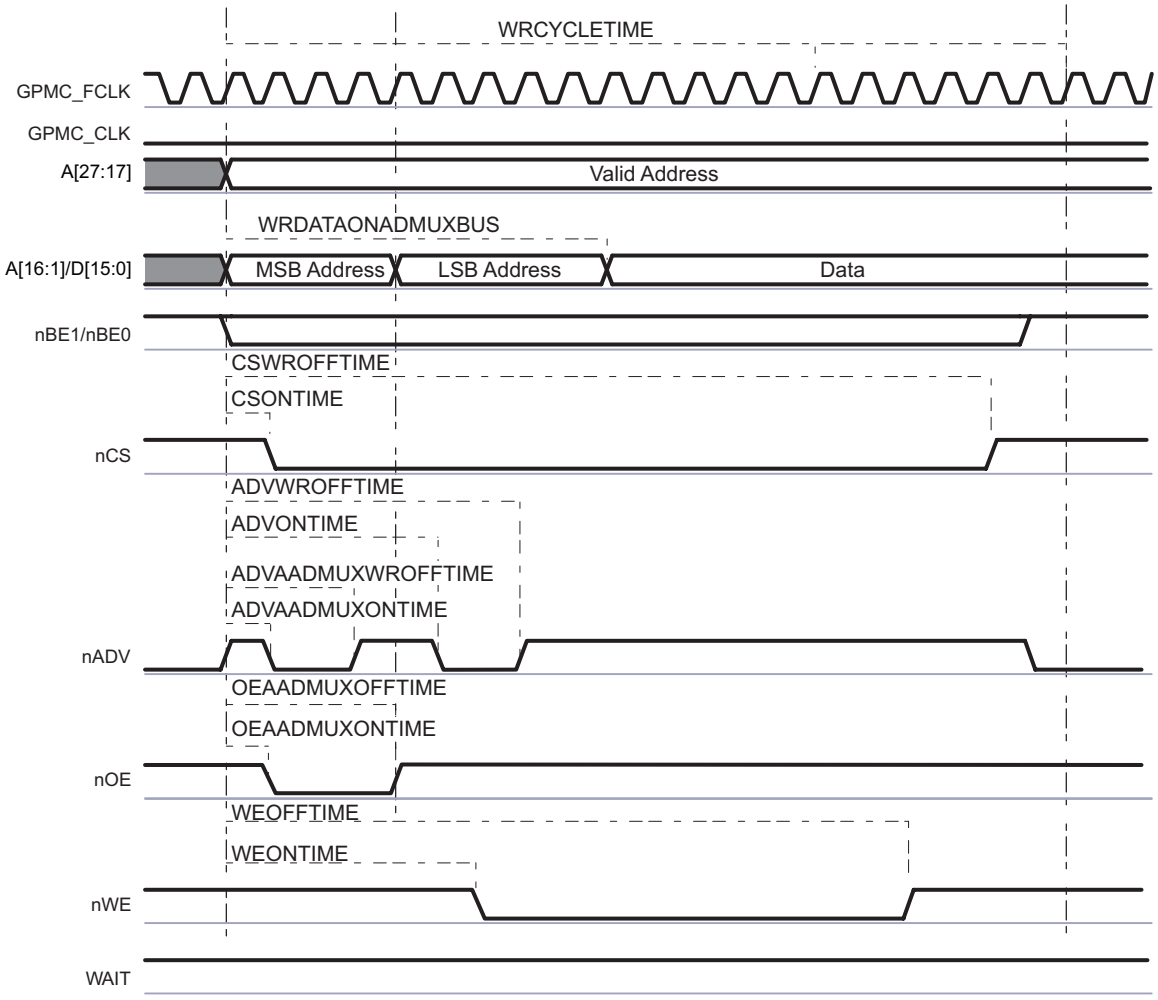
- Address valid signal $\overline{ADV}$. $\overline{ADV}$ is asserted and deasserted twice during a read transaction:
  - $\overline{ADV}$ first assertion time is controlled by the GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME field.
  - $\overline{ADV}$ first deassertion time is controlled by the GPMC_CONFIG3_i[26-24] ADVAADMUXRDOFFTIME field.
  - $\overline{ADV}$ second assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ second deassertion time is controlled by the GPMC_CONFIG3_i[12-8] ADVRDOFFTIME field.
- Output Enable signal $\overline{OE}$. $\overline{OE}$ is asserted and deasserted twice during a read transaction ($\overline{OE}$ second assertion indicates a read cycle):
  - $\overline{OE}$ first assertion time is controlled by the GPMC_CONFIG4_i[6-4] OEAADMUXONTIME field.
  - $\overline{OE}$ first deassertion time is controlled by the GPMC_CONFIG3_i[15-13] OEAADMUXOFFTIME field.
  - $\overline{OE}$ second assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - $\overline{OE}$ second deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.1.2.3 Asynchronous Single Write Operation on an AAD-Multiplexed Device

Figure 11-16 shows an asynchronous single write operation on an AAD-multiplexed device.

**Figure 11-16. Asynchronous Single Write on an AAD-Multiplexed Device**






Copyright © 2011–2016, Texas Instruments Incorporated

See Section 11.3.6.1 for formulas to calculate timing parameters.

Table 11-40 lists the timing bit fields to set up to configure the GPMC in asynchronous single write mode.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with $\overline{OE}$ driven low. The second phase for LSB address is qualified with $\overline{OE}$ driven high. The address phase ends at $\overline{WE}$ assertion time.

The $\overline{CS}$ and $\overline{WE}$ signals are controlled in the same way as for asynchronous single write operation on an address/data-multiplexed device.

- Address valid signal $\overline{ADV}$ is asserted and deasserted twice during a write transaction
  - $\overline{ADV}$ first assertion time is controlled by the GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME field.
  - $\overline{ADV}$ first deassertion time is controlled by the GPMC_CONFIG3_i[30-28] ADVAADMUXWROFFTIME field.
  - $\overline{ADV}$ second assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ second deassertion time is controlled by the GPMC_CONFIG3_i[20-16] ADVWROFFTIME field.
- Output Enable signal $\overline{OE}$ is asserted during the address phase of a write transaction
  - $\overline{OE}$ assertion time is controlled by the GPMC_CONFIG4_i[6-4] OEAADMUXONTIME field.
  - $\overline{OE}$ deassertion time is controlled by the GPMC_CONFIG3_i[15-13] OEAADMUXOFFTIME field.

The address bits for the first address phase are driven onto the data bus until $\overline{OE}$ deassertion. Data is driven onto the address/data bus at the clock edge defined by the GPMC_CONFIG6_i[19-16] WRDATAONADMUXBUS parameter.

#### *11.2.4.10.1.2.4 Asynchronous Multiple (Page) Read Operation on an AAD-Multiplexed Device*

Write multiple (page) access in asynchronous mode is not supported for AAD-multiplexed devices.

If GPMC_CONFIG1_i[28] WRITEMULTIPLE is enabled (1) with GPMC_CONFIG1_i[27] WRITETYPE as asynchronous (0), the GPMC processes single asynchronous accesses.

For accesses on non-multiplexed devices, see Section 11.2.4.10.3.

### *11.2.4.10.2 Synchronous Access Description*

This section details read and write synchronous accesses on address/data multiplexed. All information in this section can be applied to any type of memory - non-multiplexed, address and data multiplexed or AAD-multiplexed - with a difference limited to the address phase. For accesses on non-multiplexed devices, see Section 11.2.4.10.3.

In synchronous operations:

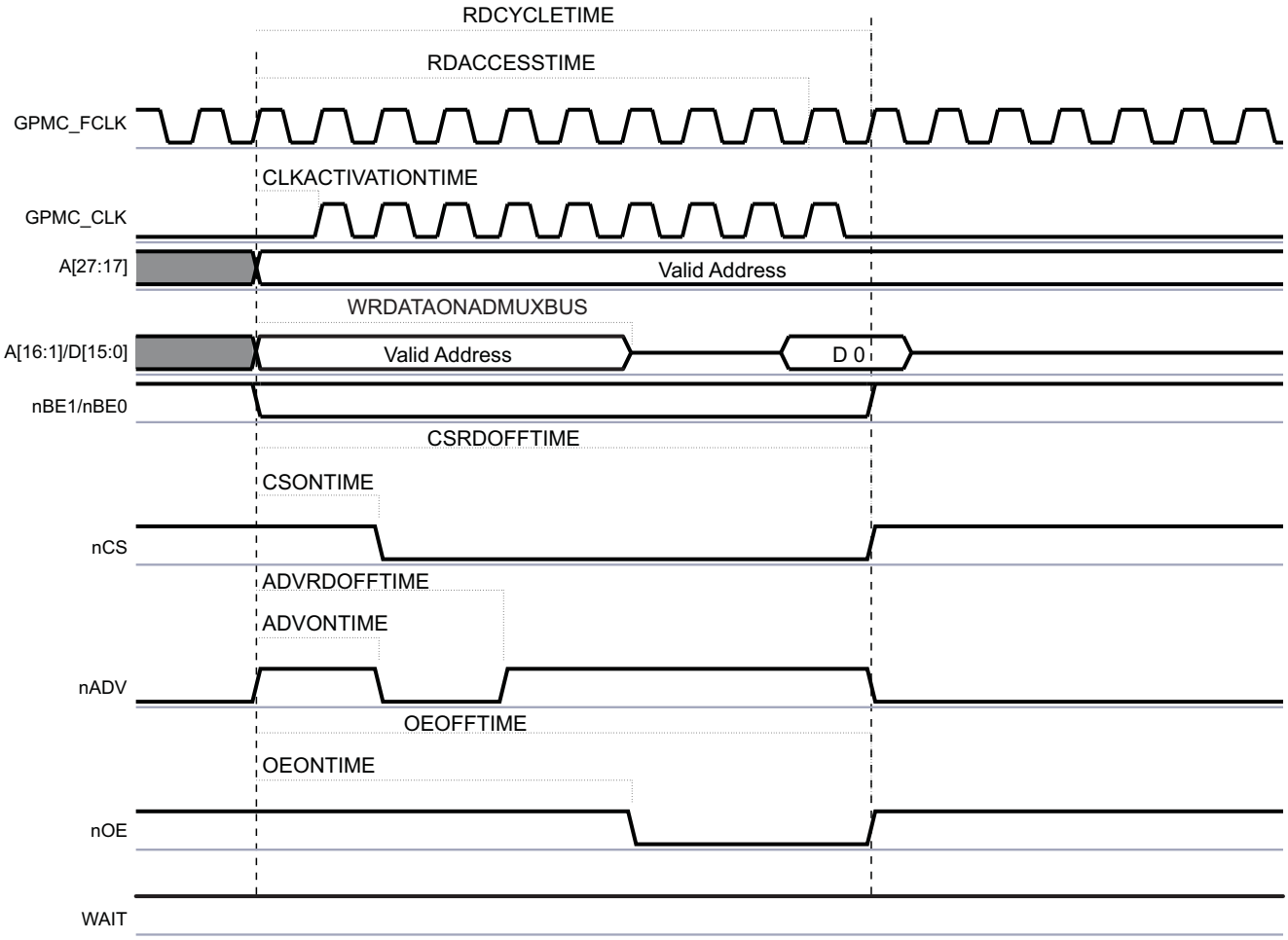
- The GPMC_CLK clock is provided outside the GPMC when accessing the memory device.
- The GPMC_CLK clock is derived from the GPMC_FCLK clock using the GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER field. In the following section, i stands for the chip-select number, i = 0 to 3.
- The GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME field specifies that the GPMC_CLK is provided outside the GPMC 0, 1, or 2 GPMC_FCLK cycles after start access time until RDCYCLETIME or WRCYCLETIME completion.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.2.1 Synchronous Single Read

Figure 11-17 and Figure 11-18 show a synchronous single-read operation with GPMCFCLKDIVIDER equal to 0 and 1, respectively.

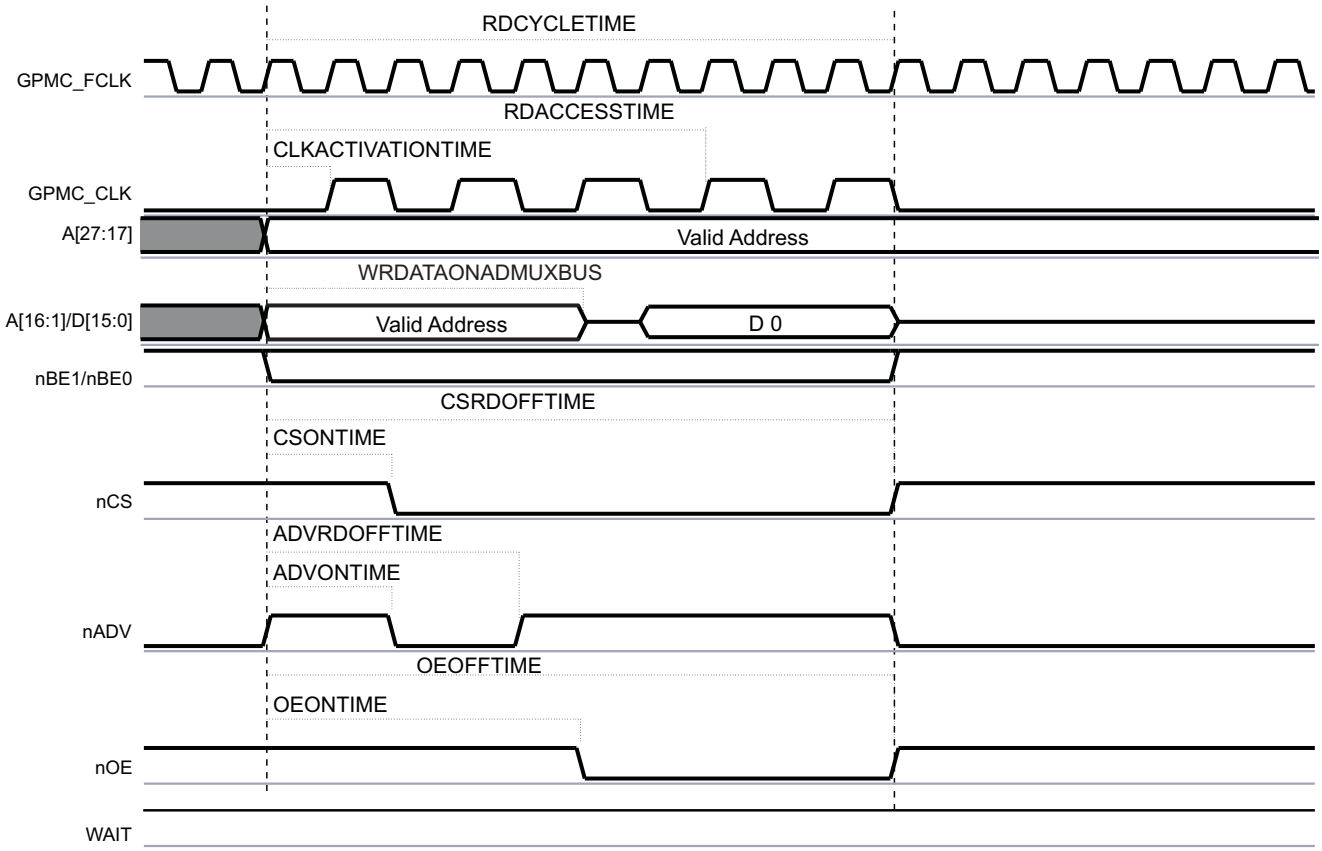
**Figure 11-17. Synchronous Single Read (GPMCFCLKDIVIDER = 0)**






Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-18. Synchronous Single Read (GPMCFCLKDIVIDER = 1)**




See Section 11.3.6.1 for formulas to calculate timing parameters.

Table 11-40 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until $\overline{OE}$ assertion time. For details, see Section 11.2.4.8.2.3.

- Chip-select signal $\overline{CS}$
  - $\overline{CS}$ assertion time is controlled by the GPMC_CONFIG2_i[3-0] CSONTIME field and ensures address setup time to $\overline{CS}$ assertion.
  - $\overline{CS}$ deassertion time is controlled by the GPMC_CONFIG2_i[12-8] CSRDOFFTIME field and ensures address hold time to $\overline{CS}$ deassertion.
- Address valid signal $\overline{ADV}$
  - $\overline{ADV}$ assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ deassertion time is controlled by the GPMC_CONFIG3_i[12-8] ADVRDOFFTIME field.
- Output enable signal $\overline{OE}$
  - $\overline{OE}$ assertion indicates a read cycle.
  - $\overline{OE}$ assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - $\overline{OE}$ deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.
- Initial latency for the first read data is controlled by GPMC_CONFIG5_i[20-16] RDACCESSTIME or by monitoring the WAIT signal.
- Total access time (GPMC_CONFIG5_i[4-0] RDCYCLETIME) corresponds to RDACCESSTIME plus the address hold time from $\overline{CS}$ deassertion, plus time from RDACCESSTIME to CSRDOFFTIME.


Copyright © 2011–2016, Texas Instruments Incorporated

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with $\overline{OE}$ driven low. The second phase for LSB address is qualified with $\overline{OE}$ driven high. The address phase ends at $\overline{WE}$ assertion time.

The $\overline{CS}$ signal is controlled in the same way as for synchronous single read operation on an address/data-multiplexed device.

- Address valid signal $\overline{ADV}$ is asserted and deasserted twice during a read transaction
  - $\overline{ADV}$ first assertion time is controlled by the GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME field.
  - $\overline{ADV}$ first deassertion time is controlled by the GPMC_CONFIG3_i[26-24] ADVAADMUXRDOFFTIME field.
  - $\overline{ADV}$ second assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ second deassertion time is controlled by the GPMC_CONFIG3_i[12-8] ADVRDOFFTIME field.
- Output Enable signal $\overline{OE}$ is asserted and deasserted twice during a read transaction ($\overline{OE}$ second assertion indicates a read cycle)
  - $\overline{OE}$ first assertion time is controlled by the GPMC_CONFIG4_i[6-4] OEAADMUXONTIME field.
  - $\overline{OE}$ first deassertion time is controlled by the GPMC_CONFIG3_i[15-13] OEAADMUXOFFTIME field.
  - $\overline{OE}$ second assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - $\overline{OE}$ second deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.

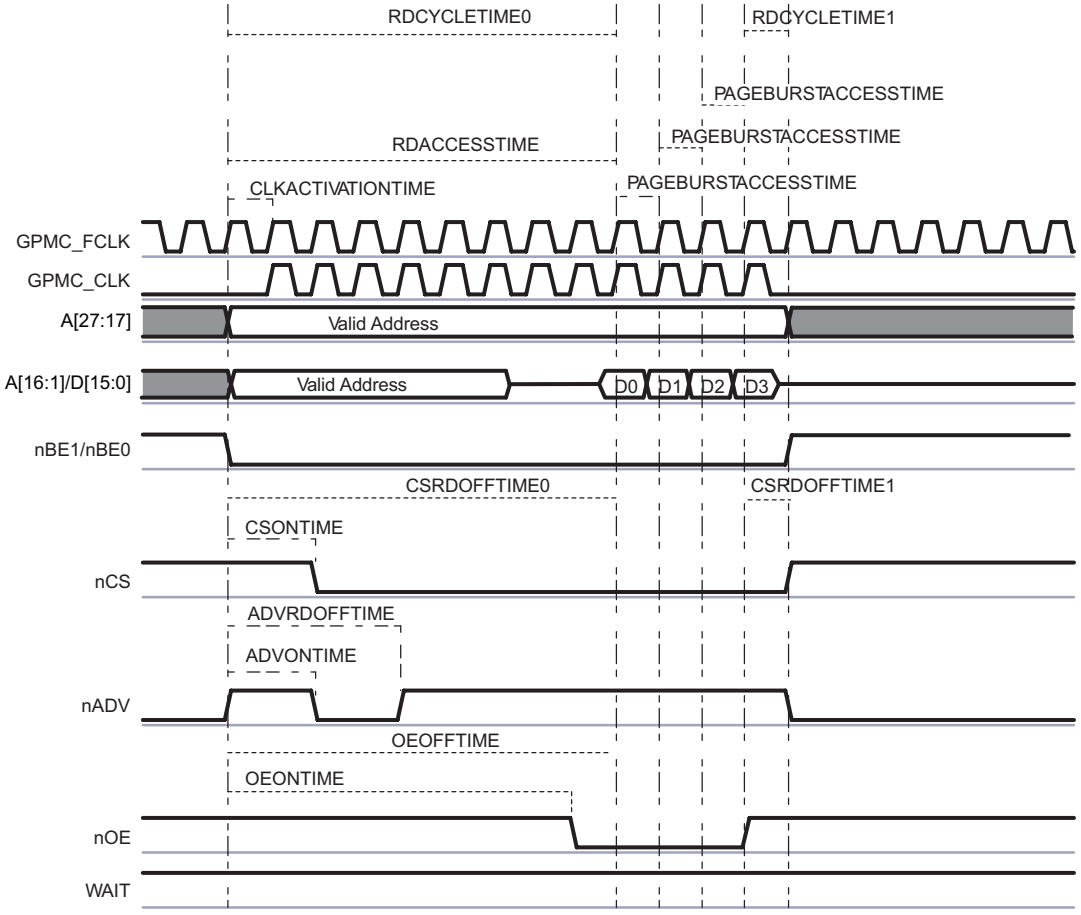
After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See Section 11.2.4.9.10.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.2.2 Synchronous Multiple (Burst) Read (4-, 8-, 16-Word16 Burst With Wraparound Capability)

Figure 11-19 and Figure 11-20 show a synchronous multiple read operation with GPMCFCLKDivider equal to 0 and 1, respectively.

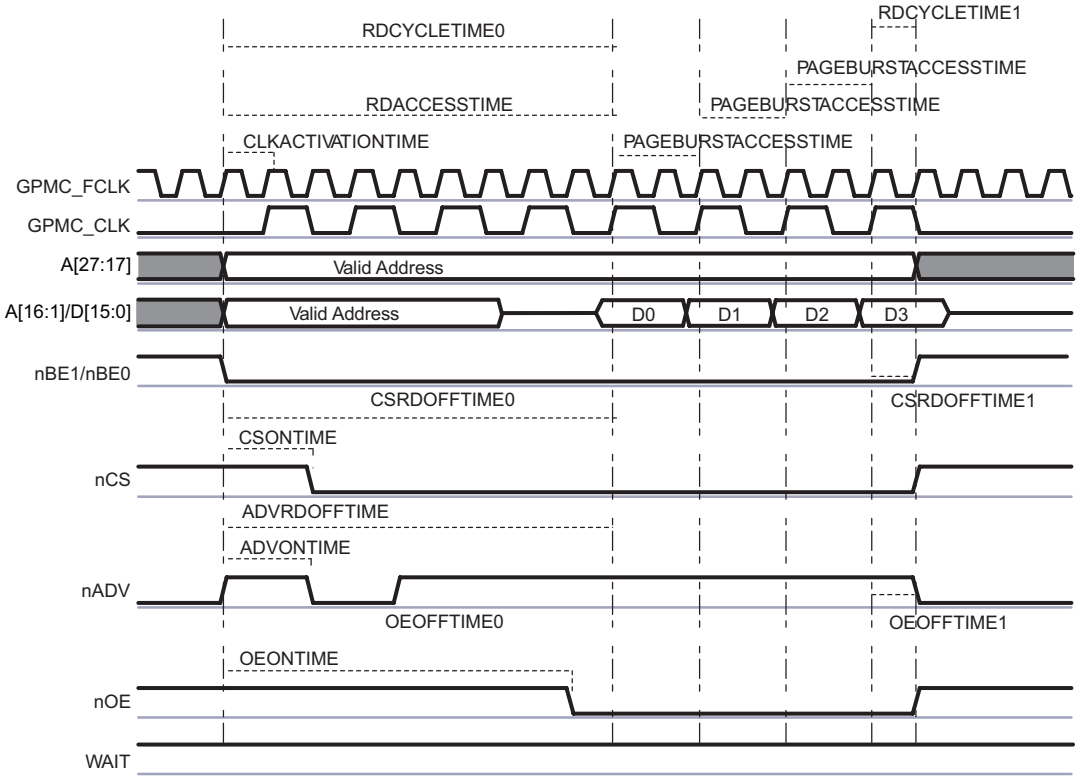
**Figure 11-19. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 0)**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-20. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 1)**




When GPMC_CONFIG5_i[20-16] RDACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

The $\overline{CS}$, $\overline{ADV}$, and $\overline{OE}$ signals are controlled in the same way as for synchronous single read operation. See Section 11.2.4.10.2.1.

Initial latency for the first read data is controlled by RDACCESSTIME or by monitoring the WAIT signal. Successive read data are provided by the memory device each one or two GPMC_CLK cycles. The PAGEBURSTACCESSTIME parameter must be set accordingly with GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER and the memory-device internal configuration. Depending on the device page length, the GPMC checks device page crossing during a new burst request and purposely insert initial latency (of RDACCESSTIME) when required.

Total access time GPMC_CONFIG5_i[4-0] RDCYCLETIME corresponds to RDACCESSTIME plus the address hold time from $\overline{CS}$ deassertion. In Figure 11-19, RDCYCLETIME programmed value equals to RDCYCLETIME0 + RDCYCLETIME1.

After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See Section 11.2.4.9.10.

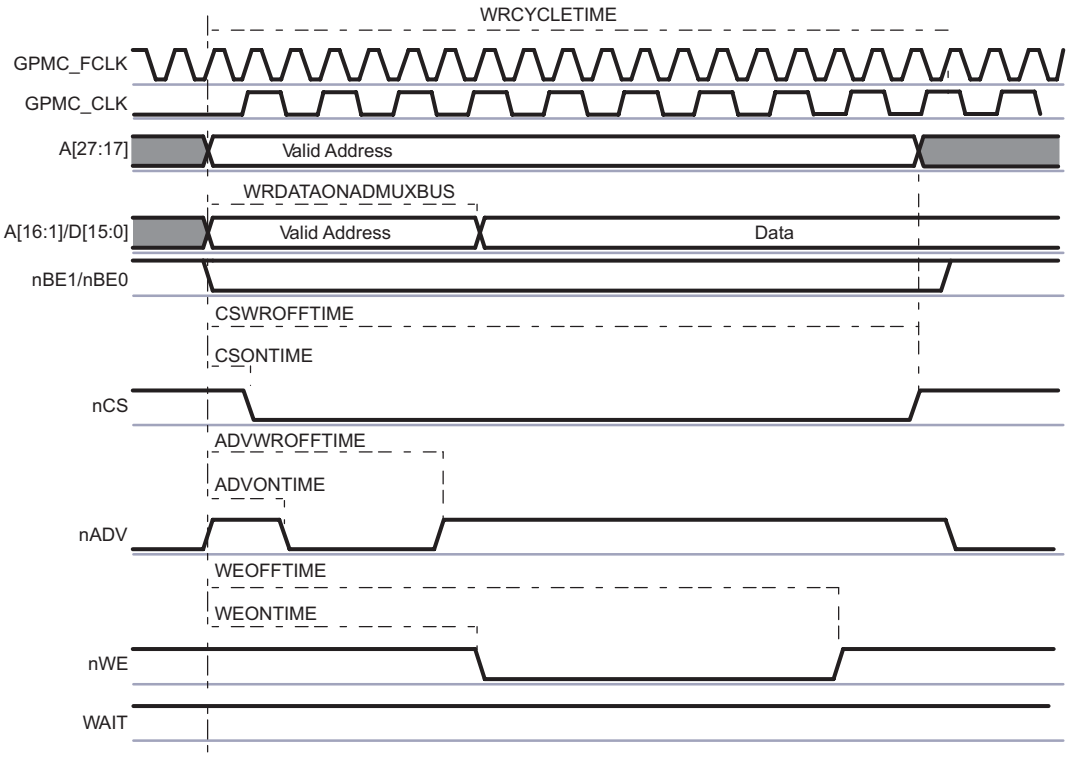
Burst wraparound is enabled through the GPMC_CONFIG1_i[31] WRAPBURST bit and allows a 4-, 8-, or 16-Word16 linear burst access to wrap within its burst-length boundary through GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.2.3 Synchronous Single Write

Burst write mode is used for synchronous single or burst accesses (see Figure 11-21).

**Figure 11-21. Synchronous Single Write on an Address/Data-Multiplexed Device**




When the GPMC generates a write access to an address/data-multiplexed device, it drives the data bus (with address bits A[16:1]) until [19:16] WRDATAONADMUXBUS time. First data of the burst is driven on the address/data bus at WRDATAONADMUXBUS time.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.2.4 Synchronous Multiple (Burst) Write

Synchronous burst write mode provides synchronous single or consecutive accesses. Figure 11-22 shows a synchronous burst write access when the chip-select is configured in address/data-multiplexed mode.

**Figure 11-22. Synchronous Multiple Write (Burst Write) in Address/Data-Multiplexed Mode**

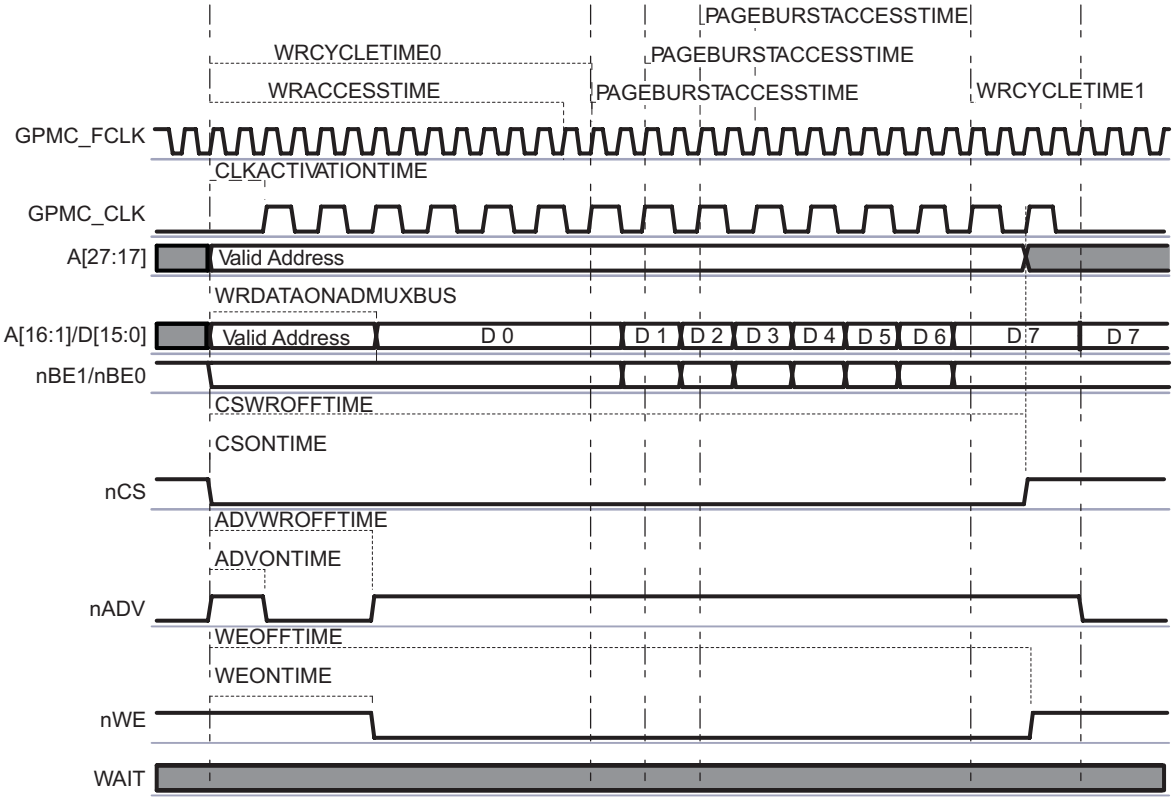
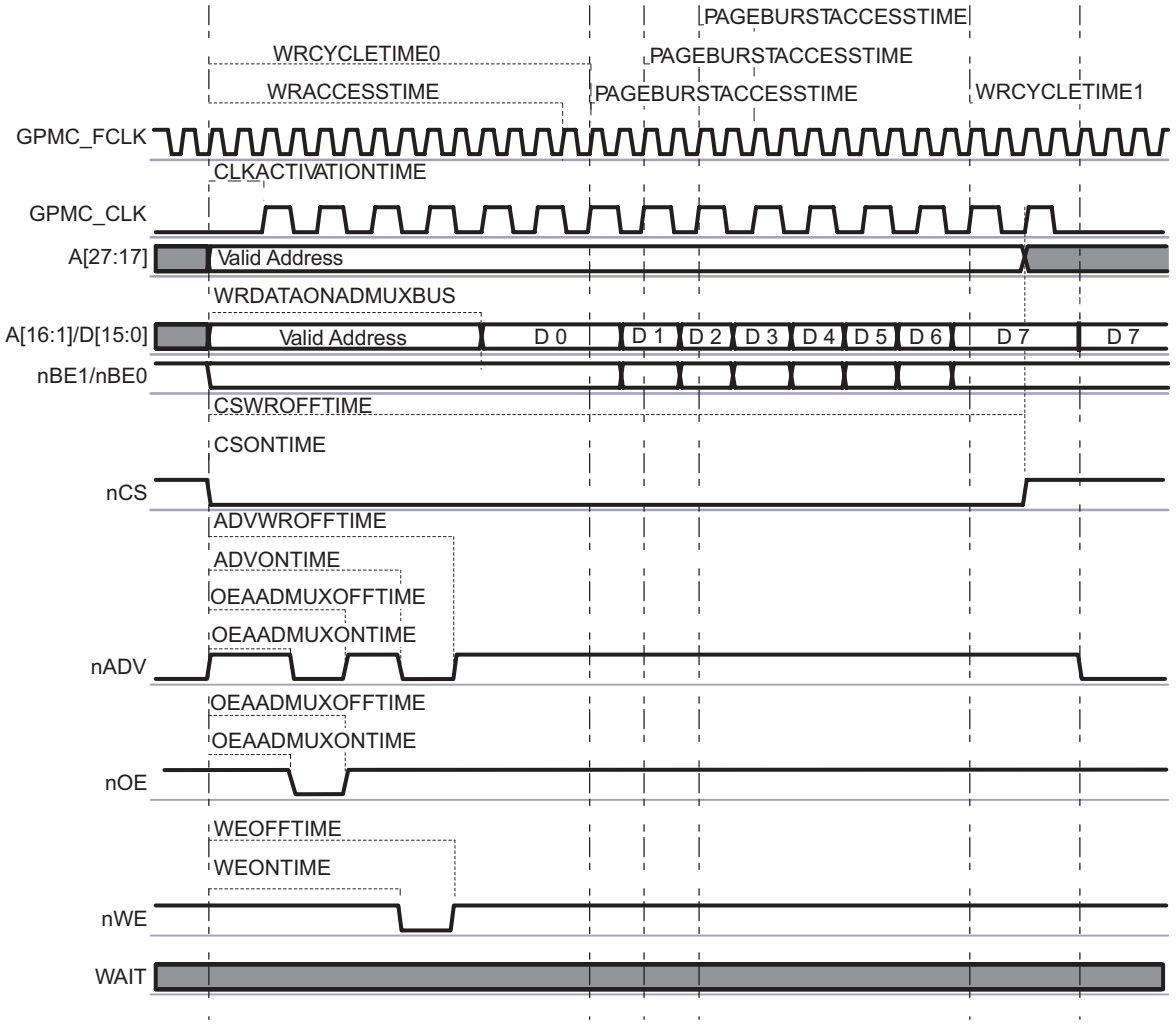



Figure 11-23 shows the same synchronous burst write access when the chip-select is configured in address/address/data-multiplexed (AAD-multiplexed) mode.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-23. Synchronous Multiple Write (Burst Write) in Address/Address/Data-Multiplexed Mode**




The first data of the burst is driven on the A/D bus at GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS.

When WRACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to the GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until $\overline{OE}$ assertion time. For details, see Section 11.2.4.8.2.3.

- Chip-select signal $\overline{CS}$
  - $\overline{CS}$ assertion time is controlled by the GPMC_CONFIG2_i[3-0] CSONTIME bit (i = 0 to 5) and ensures address setup time to $\overline{CS}$ assertion.
  - $\overline{CS}$ deassertion time controlled by the GPMC_CONFIG2_i[20-16] CSWROFFTIME field and ensures address hold time to $\overline{CS}$ deassertion.
- Address valid signal $\overline{ADV}$
  - $\overline{ADV}$ assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ deassertion time is controlled by the GPMC_CONFIG3_i[20-16] ADVWROFFTIME field.


Copyright © 2011–2016, Texas Instruments Incorporated

- Write enable signal $\overline{WE}$
  - $\overline{WE}$ assertion indicates a read cycle.
  - $\overline{WE}$ assertion time is controlled by the GPMC_CONFIG4_i[19-16] WEONTIME field.
  - $\overline{WE}$ deassertion time is controlled by the GPMC_CONFIG4_i[28-24] WEOFFTIME field.

  The $\overline{WE}$ falling edge must not be used to control the time when the burst first data is driven in the address/data bus because some new devices require the $\overline{WE}$ signal at low during the address phase.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with $\overline{OE}$ driven low. The second phase for LSB address is qualified with $\overline{OE}$ driven high. The address phase ends at $\overline{WE}$ assertion time.

The $\overline{CS}$ signal is controlled as detailled above.

- Address valid signal $\overline{ADV}$ is asserted and deasserted twice during a read transaction
  - $\overline{ADV}$ first assertion time is controlled by the GPMC_CONFIG3_i[[6-4] ADVAADMUXONTIME field.
  - $\overline{ADV}$ first deassertion time is controlled by the GPMC_CONFIG3_i[[26-24] ADVAADMUXRDOFFTIME field.
  - $\overline{ADV}$ second assertion time is controlled by the GPMC_CONFIG3_i[[3-0] ADVONTIME field.
  - $\overline{ADV}$ second deassertion time is controlled by the GPMC_CONFIG3_i[[12-8] ADVRDOFFTIME field.
- Output Enable signal $\overline{OE}$ is asserted and deasserted twice during a read transaction ($\overline{OE}$ second assertion indicates a read cycle)
  - $\overline{OE}$ first assertion time is controlled by the GPMC_CONFIG4_i[[6-4] OEAADMUXONTIME field.
  - $\overline{OE}$ first deassertion time is controlled by the GPMC_CONFIG4_i[15-13] OEAADMUXOFFTIME field.
  - $\overline{OE}$ second assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - $\overline{OE}$ second deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.

First write data is driven by the GPMC at GPMC_CONFIG6_i[19-16] WRDATAONADMUXBUS, when in address/data mux configuration. The next write data of the burst is driven on the bus at WRACCESSTIME + 1 during GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME GPMC_FCLK cycles. The last data of the synchronous burst write is driven until GPMC_CONFIG5_i[12-8] WRCYCLETIME completes.

- WRACCESSTIME is defined in the GPMC_CONFIG6_i[28-24] register.
- The PAGEBURSTACCESSTIME parameter must be set accordingly with GPMCFCLKDIVIDER and the memory-device internal configuration.

Total access time GPMC_CONFIG5_i[12-8] WRCYCLETIME corresponds to WRACCESSTIME plus the address hold time from $\overline{CS}$ deassertion. In Figure 11-23 the WRCYCLETIME programmed value equals WRCYCLETIME0 + WRCYCLETIME1. WRCYCLETIME0 and WRCYCLETIME1 delays are not actual parameters and are only a graphical representation of the full WRCYCLETIME value.

After a write operation, if no other access (read or write) is pending, the data bus keeps the previous value. See Section 11.2.4.9.10.

#### *11.2.4.10.3 Asynchronous and Synchronous Accesses in Nonmultiplexed Mode*

Page mode is only available in non-multiplexed mode.

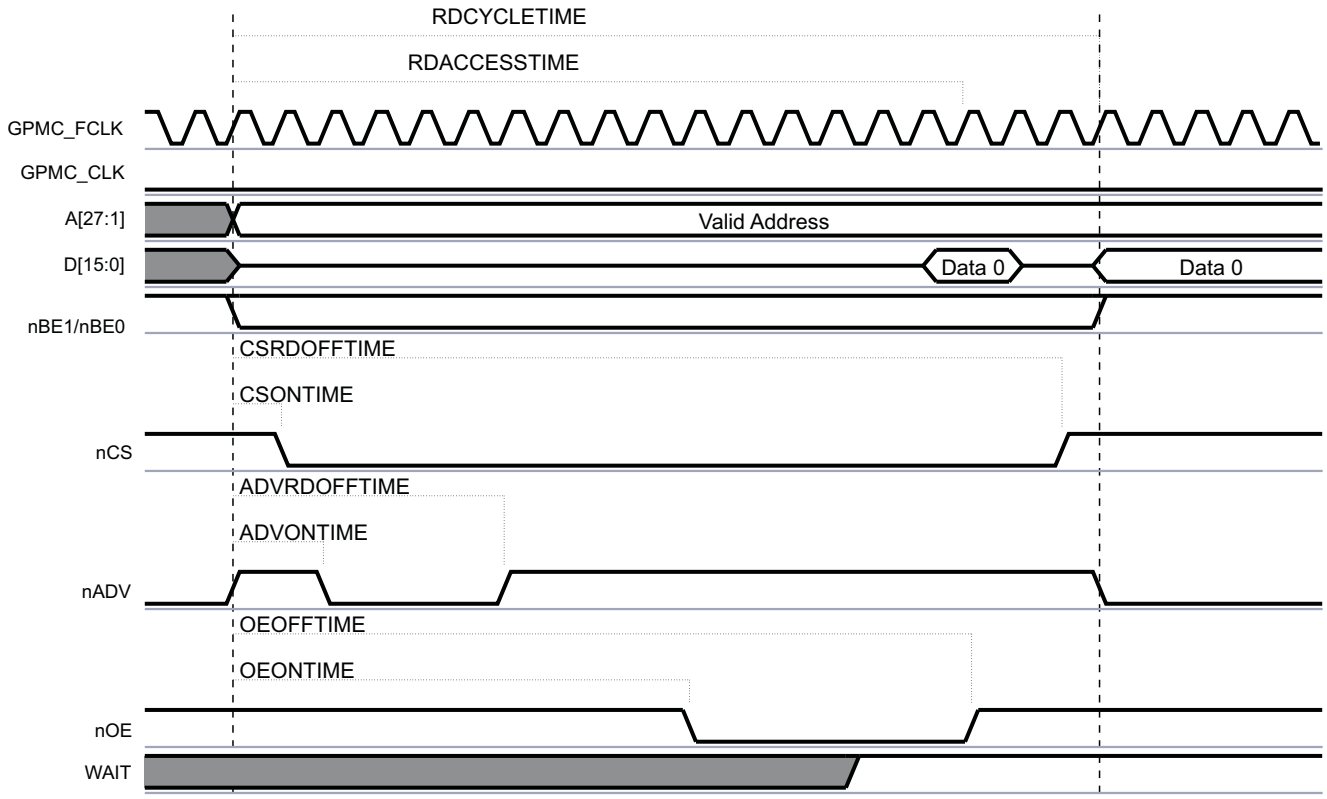
- Asynchronous single read operation on a nonmultiplexed device
- Asynchronous single write operation on a nonmultiplexed device
- Asynchronous multiple (page mode) read operation on a nonmultiplexed device
- Synchronous operations on a nonmultiplexed device


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.3.1 Asynchronous Single Read Operation on a Nonmultiplexed Device

Figure 11-24 shows an asynchronous single read operation on a nonmultiplexed device.

**Figure 11-24. Asynchronous Single Read on an Address/Data-Nonmultiplexed Device**




The 27-bit address is driven onto the address bus A[27:1] and the 16-bit data is driven onto the data bus D[15:0].

Read data is latched at GPMC_CONFIG1_5[20-16] RDACCESSTIME completion time. The end of the access is defined by the GPMC_CONFIG1_5[4-0] RDCYCLETIME parameter.

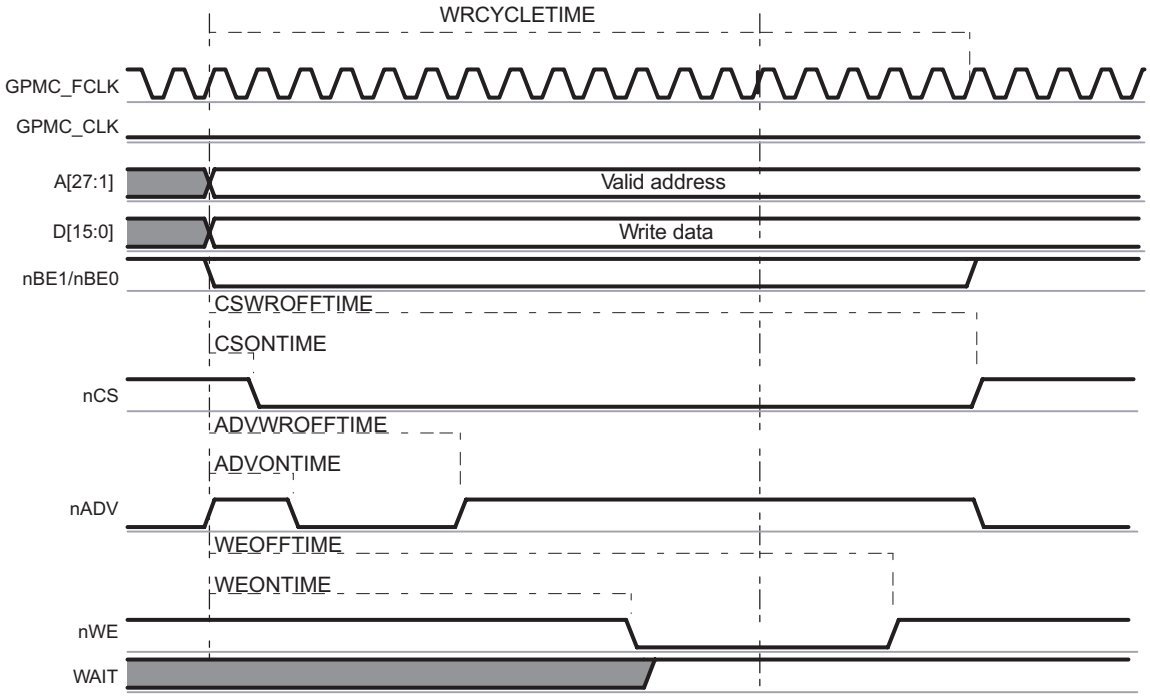
$\overline{CS}$, $\overline{ADV}$, and $\overline{OE}$ signals are controlled in the same way as address/data multiplexed accesses, see Section 11.2.4.10.1.1.2.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.3.2 Asynchronous Single Write Operation on a Nonmultiplexed Device

Figure 11-25 shows an asynchronous single write operation on a nonmultiplexed device.

**Figure 11-25. Asynchronous Single Write on an Address/Data-Nonmultiplexed Device**




The 27-bit address is driven onto the address bus A[27:1] and the 16-bit data is driven onto the data bus D[15:0].

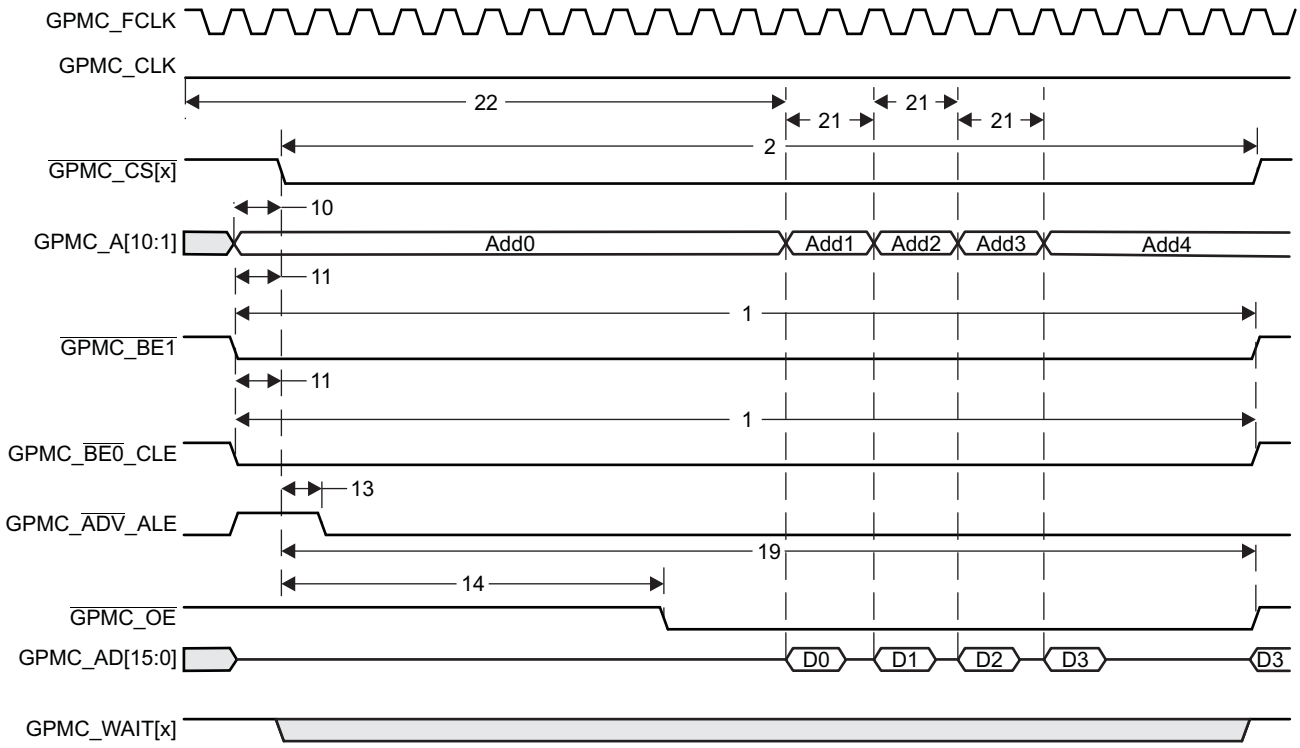
$\overline{CS}$, $\overline{ADV}$, and $\overline{WE}$ signals are controlled in the same way as address/data multiplexed accesses, see Section 11.2.4.10.1.1.3.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.3.3 Asynchronous Multiple (Page Mode) Read Operation on a Nonmultiplexed Device

Figure 11-26 shows an asynchronous multiple read operation on a Nonmultiplexed Device, in which two word32 host read accesses to the GPMC are split into one multiple (page mode of 4 word 16) read access to the attached device.

**Figure 11-26. Asynchronous Multiple (Page Mode) Read**




The WAIT signal is active low.

$\overline{CS}$, $\overline{ADV}$, and $\overline{OE}$ signals are controlled in the same way as address/data multiplexed accesses, see Section 11.2.4.10.1.1.2.

When RDACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

Read data is latched at GPMC_CONFIG5_i[20-16] RDACCESSTIME completion time (i = 0 to 5). The end of the access is defined by the GPMC_CONFIG5_i[4-0] RDCYCLETIME parameter.

During consecutive accesses, the GPMC increments the address after each data read completes.

Delay between successive read data in the page is controlled by the GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME parameter. Depending on the device page length, the GPMC can control device page crossing during a burst request and insert initial RDACCESSTIME latency. Note that page crossing is only possible with a new burst access, meaning a new initial access phase is initiated.

Total access time RDCYCLETIME corresponds to RDACCESSTIME plus the address hold time starting from the $\overline{CS}$ deassertion.

- The read cycle time is defined in the GPMC_CONFIG5_i[4-0] RDCYCLETIME field.
- In Figure 11-26, the RDCYCLETIME programmed value equals RDCYCLETIME0 (before paged accesses) + RDCYCLETIME1 (after paged accesses).

#### 11.2.4.10.3.4 Synchronous Operations on a Nonmultiplexed Device

All information for this section is equivalent to similar operations for address/data- or AAD-multiplexed accesses. The only difference resides in the address phase. See Section 11.3.3.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.10.4 Page and Burst Support

Each chip-select can be configured to process system single or burst requests into successive single accesses or asynchronous page/synchronous burst accesses, with appropriate access size adaptation.

Depending on the external device page or burst capability, read and write accesses can be independently configured through the GPMC. The GPMC_CONFIG1_i[30] READMULTIPLE and GPMC_CONFIG1_i[28] WRITMULTIPLE bits (i = 0 to 5) are associated with the READTYPE and WRITETYPE parameters.

- Asynchronous write page mode is not supported.
- 8-bit wide device support is limited to nonburstable devices (READMULTIPLE and WRITEMULTIPLE are ignored).
- Not applicable to NAND device interfacing.

#### 11.2.4.10.5 System Burst Versus External Device Burst Support

The device system can issue the following requests to the GPMC:

- Byte, 16-bit word, 32-bit word requests (byte enable controlled). This is always a single request from the interconnect point of view.
- Incrementing fixed-length bursts of two words, four words, and eight words
- Wrapped (critical word access first) fixed-length burst of two, four, or eight words

To process a system request with the optimal protocol, the READMULTIPLE (and READTYPE) and WRITEMULTIPLE (and WRITETYPE) parameters must be set according to the burstable capability (synchronous or asynchronous) of the attached device.

The GPMC access engine issues only fixed-length burst. The maximum length that can be issued is defined per CS by the GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH bit (i = 0 to 5). When the ATTACHEDDEVICEPAGELENGTH value is less than the system burst request length (including the appropriate access size adaptation according to the device width), the GPMC splits the system burst request into multiple bursts. Within the specified 4-, 8-, or 16-word value, the ATTACHEDDEVICEPAGELENGTH field value must correspond to the maximum-length burst supported by the memory device configured in fixed-length burst mode (as opposed to continuous burst mode).

To get optimal performance from memory devices that natively support 16 Word16-length-wrapping burst capability (critical word access first), the ATTACHEDDEVICEPAGELENGTH parameter must be set to 16 words and the GPMC_CONFIG1_i[31] WRAPBURST bit (i = 0 to 5) must be set to 1. Similarly DEVICEPAGELENGTH is set to 4 and 8 for memories supporting respectively 4 and 8 Word16-length-wrapping burst.

When the memory device does not offer (or is not configured to offer) native 16 Word16-length-wrapping burst, the WRAPBURST parameter must be cleared, and the GPMC access engine emulates the wrapping burst by issuing the appropriate burst sequences according to the ATTACHEDDEVICEPAGELENGTH value.

When the memory device does not support native-wrapping burst, there is usually no difference in behavior between a fixed burst length mode and a continuous burst mode configuration (except for a potential power increase from a memory-speculative data prefetch in a continuous burst read). However, even though continuous burst mode is compatible with GPMC behavior, because the GPMC access engine issues only fixed-length burst and does not benefit from continuous burst mode, it is best to configure the memory device in fixed-length burst mode.

The memory device maximum-length burst (configured in fixed-length burst wrap or nonwrap mode) usually corresponds to the memory device data buffer size. Memory devices with a minimum of 16 half-word buffers are the most appropriate (especially with wrap support), but memory devices with smaller buffer size (4 or 8) are also supported, assuming that the GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH field is set accordingly to 4 or 8 words.


Copyright © 2011–2016, Texas Instruments Incorporated

The device system issues only requests with addresses or starting addresses for nonwrapping burst requests; that is, the request size boundary is aligned. In case of an eight-word-wrapping burst, the wrapping address always occurs on the eight-words boundary. As a consequence, all words requested must be available from the memory data buffer when the buffer size is equal to or greater than the ATTACHEDDEVICEPAGELENGTH value. This usually means that data can be read from or written to the buffer at a constant rate (number of cycles between data) without wait states between data accesses. If the memory does not behave this way (nonzero wait state burstable memory), wait-pin monitoring must be enabled to dynamically control data-access completion within the burst.

When the system burst request length is less than the ATTACHEDDEVICEPAGELENGTH value, the GPMC proceeds with the required accesses.

#### 11.2.4.11 pSRAM Access Specificities

pSRAM devices are SRAM-pin-compatible low-power memories that contain a self-refreshed DRAM memory array. The GPMC_CONFIG1_i[[11-10] DEVICETYPE field (i = 0 to 3) shall be cleared to 0b00.

The pSRAM devices uses the NOR protocol. It support the following operations:

- Asynchronous single read
- Asynchronous page read
- Asynchronous single write
- Synchronous single read and write
- Synchronous burst read
- Synchronous burst write (not supported by NOR Flash memory)

pSRAM devices must be powered up and initialized in a predefined manner according to the specifications of the attached device.

pSRAM devices can be programmed to use either mode: fixed or variable latency. pSRAM devices can either automatically schedule autorefresh operations, which force the GPMC to use its WAIT signal capability when read or write operations occur during an internal self-refresh operation, or pSRAM devices automatically include the autorefresh operation in the access time. These devices do not require additional WAIT signal capability or a minimum $\overline{CS}$ high pulse width between consecutive accesses to ensure that the correct internal refresh operation is scheduled.

#### 11.2.4.12 NAND Access Description

NAND (8-bit and 16-bit) memory devices using a standard NAND asynchronous address/data-multiplexing scheme can be supported on any chip-select with the appropriate asynchronous configuration settings

As for any other type of memory compatible with the GPMC interface, accesses to a chip-select allocated to a NAND device can be interleaved with accesses to chip-selects allocated to other external devices. This interleaved capability limits the system to *chip enable don't care* NAND devices, because the chip-select allocated to the NAND device must be de-asserted if accesses to other chip-selects are requested.

##### *11.2.4.12.1 NAND Memory Device in Byte or 16-Bit Word Stream Mode*

NAND devices require correct command and address programming before data array read or write accesses. The GPMC does not include specific hardware to translate a random address system request into a NAND-specific multiphase access. In that sense, GPMC NAND support, as opposed to random memory-map device support, is data-stream-oriented (byte or 16-bit word).

The GPMC NAND programming model relies on a software driver for address and command formatting with the correct data address pointer value according to the block and page structure. Because of NAND structure and protocol interface diversity, the GPMC does not support automatic command and address phase programming, and software drivers must access the NAND device ID to ensure that correct command and address formatting are used for the identified device.

NAND device data read and write accesses are achieved through an asynchronous read or write access. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.


Copyright © 2011–2016, Texas Instruments Incorporated

Any chip-select region can be qualified as a NAND region to constrain the $\overline{\text{ADV}}$_ALE signal as Address Latch Enable (ALE active high, default state value at low) during address program access, and the $\overline{\text{BE0}}$_CLE signal as Command Latch Enable (CLE active high, default state value at low) during command program access. GPMC address lines are not used (the previous value is not changed) during NAND access.

#### 11.2.4.12.1.1 Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode

The GPMC_CONFIG7_i register (where i = 0 to 5) associated with a NAND device region interfaced in byte or word stream mode can be initialized with a minimum size of 16 Mbytes, because any address location in the chip-select memory region can be used to access a NAND data array. The NAND Flash protocol specifies an address sequence where address bits are passed through the data bus in a series of write accesses with the ALE pin asserted. After this address phase, all operations are streamed and the system requests address is irrelevant.

To allow correct command, address, and data-access controls, the GPMC_CONFIG1_i register associated with a NAND device region must be initialized in asynchronous read and write modes with the parameters shown in Table 11-11. Failure to comply with these settings corrupts the NAND interface protocol.

The GPMC_CONFIG1_i to GPMC_CONFIG4_i register (where i = 0 to 5) associated with a NAND device region must be initialized with the correct control-signal timing value according to the NAND device timing parameters.

**Table 11-11. Chip-Select Configuration for NAND Interfacing**

| Bit Field | Register | Value | Comments |
|---|---|---|---|
| WRAPBURST | GPMC_CONFIG1_i | 0 | No wrap |
| READMULTIPLE | GPMC_CONFIG1_i | 0 | Single access |
| READTYPE | GPMC_CONFIG1_i | 0 | Asynchronous mode |
| WRITEMULTIPLE | GPMC_CONFIG1_i | 0 | Single access |
| WRITETYPE | GPMC_CONFIG1_i | 0 | Asynchronous mode |
| CLKACTIVATIONTIME | GPMC_CONFIG1_i | 0b00 | |
| ATTACHEDDEVICEPAGELENGTH | GPMC_CONFIG1_i | Don't care | Single-access mode |
| WAITREADMONITORING | GPMC_CONFIG1_i | 0 | Wait not monitored by GPMC access engine |
| WAITWRITEMONITORING | GPMC_CONFIG1_i | 0 | Wait not monitored by GPMC access engine |
| WAITMONITORINGTIME | GPMC_CONFIG1_i | Don't care | Wait not monitored by GPMC access engine |
| WAITPINSELECT | GPMC_CONFIG1_i | | Select which wait is monitored by edge detectors |
| DEVICESIZE | GPMC_CONFIG1_i | 0b00 or 0b01 | 8- or 16-bit interface |
| DEVICETYPE | GPMC_CONFIG1_i | 0b10 | NAND device in stream mode |
| MUXADDDATA | GPMC_CONFIG1_i | 0b00 | Nonmultiplexed mode |
| TIMEPARAGRANULARITY | GPMC_CONFIG1_i | 0 | Timing achieved with best GPMC clock granularity |
| GPMCFCLKDIVIDER | GPMC_CONFIG1_i | Don't care | Asynchronous mode |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.1.2 NAND Device Command and Address Phase Control

NAND devices require multiple address programming phases. The MPU software driver is responsible for issuing the correct number of command and address program accesses, according to the device command set and the device address-mapping scheme.

NAND device-command and address-phase programming is achieved through write requests to the GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i (where i = 0 to 5) register locations with the correct command and address values. These locations are mapped in the associated chip-select register region. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Command and address values are not latched during the access and cannot be read back at the register location.

- Only write accesses must be issued to these locations, but the GPMC does not discard any read access. Accessing a NAND device with $\overline{OE}$ and CLE or ALE asserted (read access) can produce undefined results.
- Write accesses to the GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i (where i = 0 to 5) register locations must be posted for faster operations. The GPMC_CONFIG[0] NANDFORCEPOSTEDWRITE bit enables write accesses to these locations as posted, even if they are defined as nonposted.

A write buffer is used to store write transaction information before the external device is accessed:

- Up to eight consecutive posted write accesses can be accepted and stored in the write buffer.
- For nonposted write, the pipeline is one deep.
- A GPMC_STATUS[0] EMPTYWRITEBUFFERSTATUS bit stores the empty status of the write buffer.

The GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i (where i = 0 to 5) registers are 32-bit word locations, which means any 32-bit word or 16-bit word access is split into 4- or 2-byte accesses if an 8-bit wide NAND device is attached. For multiple-command phase or multiple-address phase, the software driver can use 32-bit word or 16-bit word access to these registers, but it must account for the splitting and little-endian ordering scheme. When only one byte command or address phase is required, only byte write access to a GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i can be used, and any of the four byte locations of the registers are valid.

The same applies to GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i (where i = 0 to 5) 32-bit word write access to a 16-bit wide NAND device (split into two 16-bit word accesses). In the case of a 16-bit word write access, the MSByte of the 16-bit word value must be set according to the NAND device requirement (usually 0). Either 16-bit word location or any one of the four byte locations of the registers is valid


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.1.3 Command Latch Cycle

Writing data at the GPMC_NAND_COMMAND_i (where i = 0 to 5) register location places the data as the NAND command value on the bus, using a regular asynchronous write access.

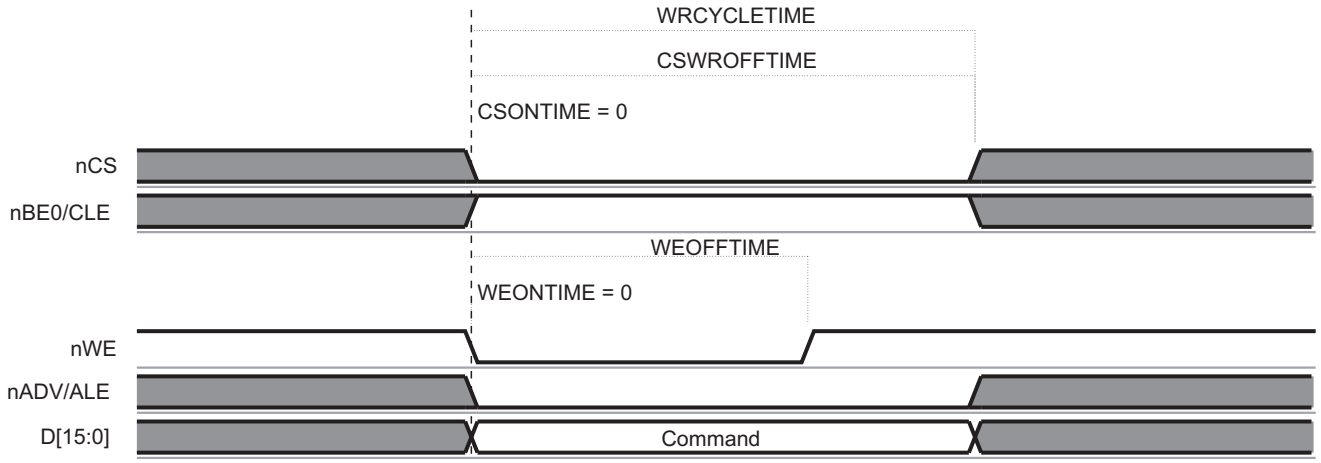
- $\overline{CE}$ is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- CLE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- WE is controlled by the WEONTIME and WEOFFTIME timing parameters.
- ALE and $\overline{RE}$ ($\overline{OE}$) are maintained inactive.

Figure 11-27 shows the NAND command latch cycle.

CLE is shared with the $\overline{BE0}$ output signal and has an inverted polarity from $\overline{BE0}$. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, $\overline{BE0}$ (also $\overline{BE1}$) must not toggle, because it is shared with CLE.

NAND Flash memories do not use byte enable signals at all.

**Figure 11-27. NAND Command Latch Cycle**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.1.4 Address Latch Cycle

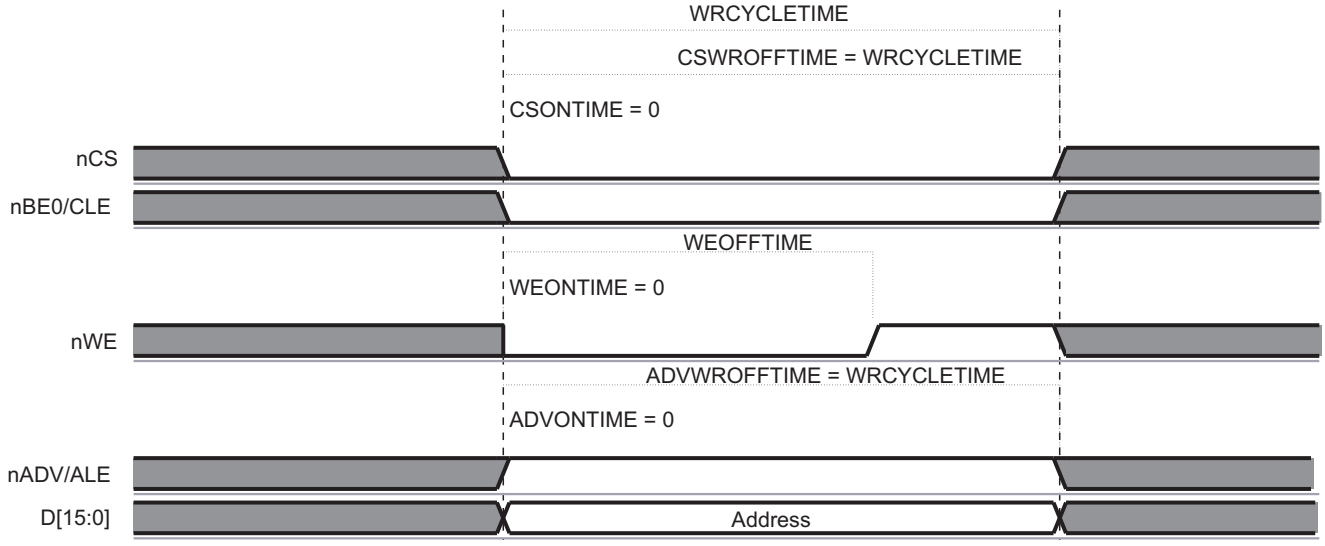
Writing data at the GPMC_NAND_ADDRESS_i (where i = 0 to 5) register location places the data as the NAND partial address value on the bus, using a regular asynchronous write access.

- $\overline{CS}$ is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- ALE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- $\overline{WE}$ is controlled by the WEONTIME and WEOFFTIME timing parameters.
- CLE and $\overline{RE}$ ($\overline{OE}$) are maintained inactive.

Figure 11-28 shows the NAND address latch cycle.

ALE is shared with the $\overline{ADV}$ output signal and has an inverted polarity from $\overline{ADV}$. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, ALE is kept stable.

**Figure 11-28. NAND Address Latch Cycle**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.1.5 NAND Device Data Read and Write Phase Control in Stream Mode

NAND device data read and write accesses are achieved through a read or write request to the chip-select-associated memory region at any address location in the region or through a read or write request to the GPMC_NAND_DATA_i (where i = 0 to 5) register location mapped in the chip-select-associated control register region. GPMC_NAND_DATA_i is not a true register, but an address location to enable $\overline{RE}$ or $\overline{WE}$ signal control. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

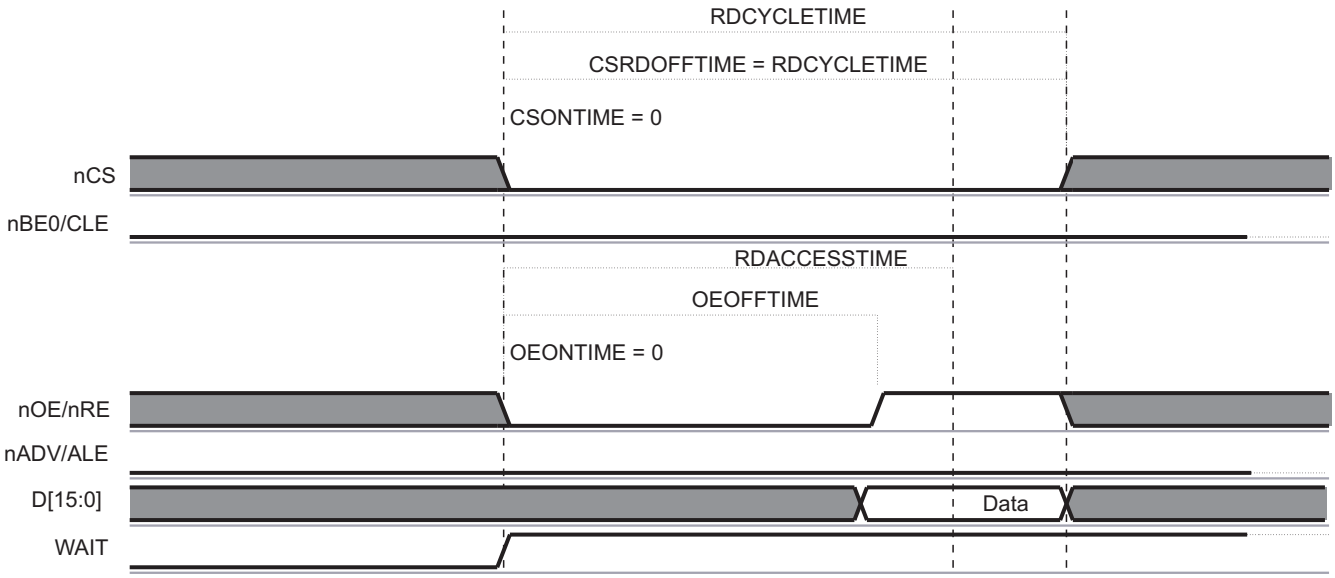
Reading data from the GPMC_NAND_DATA_i location or from any location in the associated chip-select memory region activates an asynchronous read access.

- $\overline{CS}$ is controlled by the CSONTIME and CSRDOFFTIME timing parameters.
- $\overline{RE}$ is controlled by the OEONTIME and OEOFFTIME timing parameters.
- To take advantage of $\overline{RE}$ high-to-data invalid minimum timing value, the RDACCESSTIME can be set so that data are effectively captured after $\overline{RE}$ deassertion. This allows optimization of NAND read access cycle time completion. For optimal timing parameter settings, see the NAND device and the device IC timing parameters.

ALE, CLE, and $\overline{WE}$ are maintained inactive.

Figure 11-29 shows the NAND data read cycle.

**Figure 11-29. NAND Data Read Cycle**



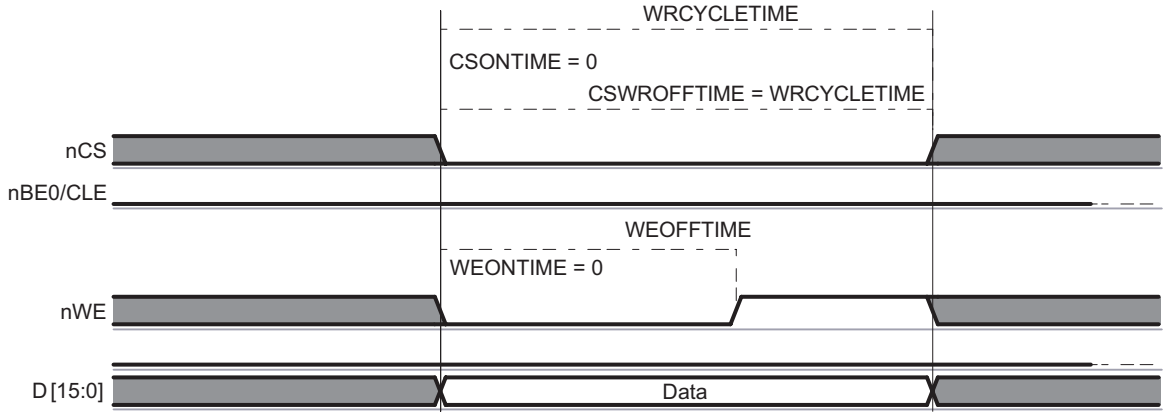

Writing data to the GPMC_NAND_DATA_i location or to any location in the associated chip-select memory region activates an asynchronous write access.

- $\overline{CS}$ is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- $\overline{WE}$ is controlled by the WEONTIME and WEOFFTIME timing parameters.
- ALE, CLE, and $\overline{RE}$ ($\overline{OE}$) are maintained inactive.

Figure 11-30 shows the NAND data write cycle.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-30. NAND Data Write Cycle**




#### 11.2.4.12.1.6 NAND Device General Chip-Select Timing Control Requirement

For most NAND devices, read data access time is dominated by $\overline{CS}$-to-data-valid timing and has faster $\overline{RE}$-to-data-valid timing. Successive accesses with $\overline{CS}$ deassertions between accesses are affected by this timing constraint. Because accesses to a NAND device can be interleaved with other chip-select accesses, there is no certainty that $\overline{CS}$ always stays low between two accesses to the same chip-select. Moreover, an $\overline{CS}$ deassertion time between the same chip-select NAND accesses is likely to be required as follows: the $\overline{CS}$ deassertion requires programming CYCLETIME and RDACCESSTIME according to the $\overline{CS}$-to-data-valid critical timing.

To get full performance from NAND read and write accesses, the prefetch engine can dynamically reduce RDCYCLETIME, WRCYCLETIME, RDACCESSTIME, WRACCESSTIME, CSRDOFFTIME, CSWROFFTIME, ADVRDOFFTIME, ADVWROFFTIME, OEOFFTIME, and WEOFFTIME on back-to-back NAND accesses (to the same memory) and suppress the minimum $\overline{CS}$ high pulse width between accesses. For more information about optimal prefetch engine access, see Section 11.2.4.12.4.

Some NAND devices require minimum write-to-read idle time, especially for device-status read accesses following status-read command programming (write access). If such write-to-read transactions are used, a minimum $\overline{CS}$ high pulse width must be set. For this, CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDELAY must be set according to the appropriate timing requirement to prevent any timing violation.

NAND devices usually have an important $\overline{RE}$ high to data bus in tristate mode. This requires a bus turnaround setting (BUSTURNAROUND = 1), so that the next access to a different chip-select is delayed until the BUSTURNAROUND delay completes. Back-to-back NAND read accesses to the same NAND Flash are not affected by the programmed bus turnaround delay.

#### 11.2.4.12.1.7 Read and Write Access Size Adaptation

##### 11.2.4.12.1.7.1 8-Bit Wide NAND Device

Host 16-bit word and 32-bit word read and write access requests to a chip-select associated with an 8-bit wide NAND device are split into successive read and write byte accesses to the NAND memory device. Byte access is ordered according to little-endian organization. A NAND 8-bit wide device must be interfaced on the D0D7 interface bus lane. GPMC data accesses are justified on this bus lane when the chip-select is associated with an 8-bit wide NAND device.

##### 11.2.4.12.1.7.2 16-Bit Wide NAND Device

Host 32-bit word read and write access requests to a chip-select associated with a 16-bit wide NAND device are split into successive read and write 16-bit word accesses to the NAND memory device. 16-bit word access is ordered according to little-endian organization.


Copyright © 2011–2016, Texas Instruments Incorporated

Host byte read and write access requests to a 16-bit wide NAND device are completed as 16-bit accesses on the device itself, because there is no byte-addressing capability on 16-bit wide NAND devices. This means that the NAND device address pointer is incremented on a 16-bit word basis and not on a byte basis. For a read access, only the requested byte is given back to the host, but the remaining byte is not stored or saved by the GPMC, and the next byte or 16-bit word read access gets the next 16-bit word NAND location. For a write access, the invalid byte part of the 16-bit word is driven to FF, and the next byte or 16-bit word write access programs the next 16-bit word NAND location.

Generally, byte access to a 16-bit wide NAND device should be avoided, especially when ECC calculation is enabled. 8-bit or 16-bit ECC-based computations are corrupted by a byte read to a 16-bit wide NAND device, because the nonrequested byte is considered invalid on a read access (not captured on the external data bus; FF is fed to the ECC engine) and is set to FF on a write access.

Host requests (read/write) issued in the chip-select memory region are translated in successive single or split accesses (read/write) to the attached device. Therefore, incrementing 32-bit burst requests are translated in multiple 32-bit sequential accesses following the access adaptation of the 32-bit to 8- or 16-bit device.

#### *11.2.4.12.2 NAND Device-Ready Pin*

The NAND memory device provides a ready pin to indicate data availability after a block/page opening and to indicate that data programming is complete. The ready pin can be connected to one of the WAIT GPMC input pins; data read accesses must not be tried when the ready pin is sampled inactive (device is not ready) even if the associated chip-select WAITREADMONITORING bit field is set. The duration of the NAND device busy state after the block/page opening is so long (up to 50 µs) that accesses occurring when the ready pin is sampled inactive can stall GPMC access and eventually cause a system time-out.

If a read access to a NAND flash is done using the wait monitoring mode, the device is blocked during a page opening, and so is the GPMC. If the correct settings are used, other chip-selects can be used while the memory processes the page opening command.

To avoid a time-out caused by a block/page opening delay in NAND flash, disable the wait pin monitoring for read and write accesses (that is, set the GPMC_CONFIG1_i[[21] WAITWRITEMONITORING and GPMC_CONFIG1_i[[22] WAITREADMONITORING bits to 0, where i = 0 to 5), and use one of the following methods instead:

- Use software to poll the WAITnSTATUS bit (n = 0 to 1) of the GPMC_STATUS register.
- Configure an interrupt that is generated on the WAIT signal change (through the GPMC_IRQENABLE [11-8] bits).

Even if the READWAITMONITORING bit is not set, the external memory nR/B pin status is captured in the programmed WAIT bit in the GPMC_STATUS register.

The READWAITMONITORING bit method must be used for other memories than NAND flash, if they require the use of a WAIT signal.

##### *11.2.4.12.2.1 Ready Pin Monitored by Software Polling*

The ready signal state can be monitored through the GPMC_STATUS WAITxSTATUS bit (x = 0 or 1). The software must monitor the ready pin only when the signal is declared valid. Refer to the NAND device timing parameters to set the correct software temporization to monitor ready only after the invalid window is complete from the last read command written to the NAND device.

##### *11.2.4.12.2.2 Ready Pin Monitored by Hardware Interrupt*

Each gpmc_wait input pin can generate an interrupt when a wait-to-no-wait transition is detected. Depending on whether the GPMC_CONFIG WAITxPINPOLARITY bits (x = 0 or 1) is active low or active high, the wait-to-no-wait transition is a low-to-high external WAIT signal transition or a high-to-low external WAIT signal transition, respectively.


Copyright © 2011–2016, Texas Instruments Incorporated

The wait transition pin detector must be cleared before any transition detection. This is done by writing 1 to the WAITxEDGEDETECTIONSTATUS bit (x = 0 or 1) of the GPMC_IRQSTATUS register according to the gpmc_wait pin used for the NAND device-ready signal monitoring. To detect a wait-to-no-wait transition, the transition detector requires a wait active time detection of a minimum of two GPMC_FCLK cycles. Software must incorporate precautions to clear the wait transition pin detector before wait (busy) time completes.

A wait-to-no-wait transition detection can issue a GPMC interrupt if the WAITxEDGEDETECTIONENABLE bit in the GPMC_IRQENABLE register is set and if the WAITxEDGEDETECTIONSTATUS bit field in the GPMC_IRQSTATUS register is set.

The WAITMONITORINGTIME field does not affect wait-to-no-wait transition time detection.

It is also possible to poll the WAITxEDGEDETECTIONSTATUS bit field in the GPMC_IRQSTATUS register according to the gpmc_wait pin used for the NAND device ready signal monitoring.

#### 11.2.4.12.3 ECC Calculator

The General Purpose Memory Controller includes an Error Code Correction (ECC) calculator circuitry that enables on the fly ECC calculation during data read or data program (that is, write) operations. The page size supported by the ECC calculator in one calculation/context is 512 bytes.

The user can choose from two different algorithms with different error correction capabilities through the GPMC_ECC_CONFIG[16] ECCALGORITHM bit:

- Hamming code for 1-bit error code correction on 8- or 16-bit NAND Flash organized with page size greater than 512 bytes
- BCH (Bose-Chaudhurl-Hocquenghem) code for 4- to 16-bit error correction

The GPMC does not directly handle the error code correction itself. During writes, the GPMC computes parity bits. During reads, the GPMC provides enough information for the processor to correct errors without reading the data buffer all over again.

The Hamming code ECC is based on a 2-dimensional (row and column) bit parity accumulation. This parity accumulation is either accomplished on the programmed number of bytes or 16-bit words read from the memory device, or written to the memory device in stream mode.

Because the ECC engine includes only one accumulation context, it can be allocated to only one chip-select at a time through the GPMC_ECC_CONFIG[3-1] ECCCS bit field. Even if two CS use different ECC algorithms, one the Hamming code and the other a BCH code, they must define separate ECC contexts because some of the ECC registers are common to all types of algorithms.

##### 11.2.4.12.3.1 Hamming Code

All references to Error Code Correction (ECC) in this subsection refer to the 1-bit error correction Hamming code.

The ECC is based on a two-dimensional (row and column) bit parity accumulation known as Hamming Code. The parity accumulation is done for a programmed number of bytes or 16-bit word read from the memory device or written to the memory device in stream mode.

There is no automatic error detection or correction, and it is the software NAND driver responsibility to read the multiple ECC calculation results, compare them to the expected code value, and take the appropriate corrective actions according to the error handling strategy (ECC storage in spare byte, error correction on read, block invalidation).

The ECC engine includes a single accumulation context. It can be allocated to a single designated chip-select at a time and parallel computations on different chip-selects are not possible. Since it is allocated to a single chip-select, the ECC computation is not affected by interleaved GPMC accesses to other chip-selects and devices. The ECC accumulation is sequentially processed in the order of data read from or written to the memory on the designated chip-select. The ECC engine does not differentiate read accesses from write accesses and does not differentiate data from command or status information. It is the software responsibility to make sure only relevant data are passed to the NAND flash memory while the ECC computation engine is active.


Copyright © 2011–2016, Texas Instruments Incorporated

The starting NAND page location must be programmed first, followed by an ECC accumulation context reset with an ECC enabling, if required. The NAND device accesses discussed in the following sections must be limited to data read or write until the specified number of ECC calculations is completed.

#### 11.2.4.12.3.1.1 ECC Result Register and ECC Computation Accumulation Size

The GPMC includes up to nine ECC result registers (GPMC_ECCj_RESULT, j = 1 to 9) to store ECC computation results when the specified number of bytes or 16-bit words has been computed.

The ECC result registers are used sequentially; one ECC result is stored in one ECC result register on the list, the next ECC result is stored in the next ECC result register on the list, and so forth, until the last ECC computation. The value of the GPMC_ECCj_RESULT register value is valid only when the programmed number of bytes or 16-bit words has been accumulated, which means that the same number of bytes or 16-bit words has been read from or written to the NAND device in sequence.

The GPMC_ECC_CONTROL[3-0] ECCPOINTER field must be set to the correct value to select the ECC result register to be used first in the list for the incoming ECC computation process. The ECCPointer can be read to determine which ECC register is used in the next ECC result storage for the ongoing ECC computation. The value of the GPMC_ECCj_RESULT register (j = 1 to 9) can be considered valid when ECCPOINTER equals j + 1. When the GPMC_ECCj_RESULT (where j = 9) is updated, ECCPOINTER is frozen at 10, and ECC computing is stopped (ECCENABLE = 0).

The ECC accumulator must be reset before any ECC computation accumulation process. The GPMC_ECC_CONTROL[8] ECCCLEAR bit must be set to 1 (nonpersistent bit) to clear the accumulator and all ECC result registers.

For each ECC result (each register, j = 1 to 9), the number of bytes or 16-bit words used for ECC computing accumulation can be selected from between two programmable values.

The ECCjRESULTSIZE bits (j = 1 to 9) in the GPMC_ECC_SIZE_CONFIG register select which programmable size value (ECCSIZE0 or ECCSIZE1) must be used for this ECC result (stored in GPMC_ECCj_RESULT register ).

The ECCSIZE0 and ECCSIZE1 fields allow selection of the number of bytes or 16-bit words used for ECC computation accumulation. Any even values from 2 to 512 are allowed.

Flexibility in the number of ECCs computed and the number of bytes or 16-bit words used in the successive ECC computations enables different NAND page error-correction strategies. Usually based on 256 or 512 bytes and on 128 or 256 16-bit word, the number of ECC results required is a function of the NAND device page size. Specific ECC accumulation size can be used when computing the ECC on the NAND spare byte.

For example, with a 2 Kbyte data page 8-bit wide NAND device, eight ECCs accumulated on 256 bytes can be computed and added to one extra ECC computed on the 24 spare bytes area where the eight ECC results used for comparison and correction with the computed data page ECC are stored. The GPMC then provides nine GPMC_ECCj_RESULT registers (j= 1 to 9) to store the results. In this case, ECCSIZE0 is set to 256, and ECCSIZE1 is set to 24; the ECC[1-8]RESULTSIZE bits are cleared to 0, and the ECC9RESULTSIZE bit is set to 1.

#### 11.2.4.12.3.1.2 ECC Enabling

The GPMC_ECC_CONFIG[3-1] ECCCS field selects the allocated chip-select. The GPMC_ECC_CONFIG[0] ECCENABLE bit enables ECC computation on the next detected read or write access to the selected chip-select.

The ECCPOINTER, ECCCLEAR, ECCSIZE, ECCjRESULTSIZE (where j = 1 to 9), ECC16B, and ECCCS fields must not be changed or cleared while an ECC computation is in progress.

The ECC accumulator and ECC result register must not be changed or cleared while an ECC computation is in progress.

Table 11-12 describes the ECC enable settings.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-12. ECC Enable Settings**

| Bit Field | Register | Value | Comments |
|---|---|---|---|
| ECCCS | GPMC_ECC_CONFIG | 0-3h | Selects the chip-select where ECC is computed |
| ECC16B | GPMC_ECC_CONFIG | 0-1 | Selects column number for ECC calculation |
| ECCCLEAR | GPMC_ECC_CONTROL | 0-7h | Clears all ECC result registers |
| ECCPOINTER | GPMC_ECC_CONTROL | 0-7h | A write to this bit field selects the ECC result register where the first ECC computation is stored. Set to 1 by default. |
| ECCSIZE1 | GPMC_ECC_SIZE_CONFIG | 0-FFh | Defines ECCSIZE1 |
| ECCSIZE0 | GPMC_ECC_SIZE_CONFIG | 0-FFh | Defines ECCSIZE0 |
| ECCjRESULTSIZE | GPMC_ECC_SIZE_CONFIG | 0-1 | Selects the size of ECCn result register |
| ECCENABLE | GPMC_ECC_CONFIG | 1 | Enables the ECC computation |

#### *11.2.4.12.3.1.3 ECC Computation*

The ECC algorithm is a multiple parity bit accumulation computed on the odd and even bit streams extracted from the byte or Word 16 streams. The parity accumulation is split into row and column accumulations, as shown in Figure 11-31 and Figure 11-32. The intermediate row and column parities are used to compute the upper level row and column parities. Only the final computation of each parity bit is used for ECC comparison and correction.

P1o = bit7 XOR bit5 XOR bit3 XOR bit1 on each byte of the data stream

P1e = bit6 XOR bit4 XOR bit2 XOR bit0 on each byte of the data stream

P2o = bit7 XOR bit6 XOR bit3 XOR bit2 on each byte of the data stream

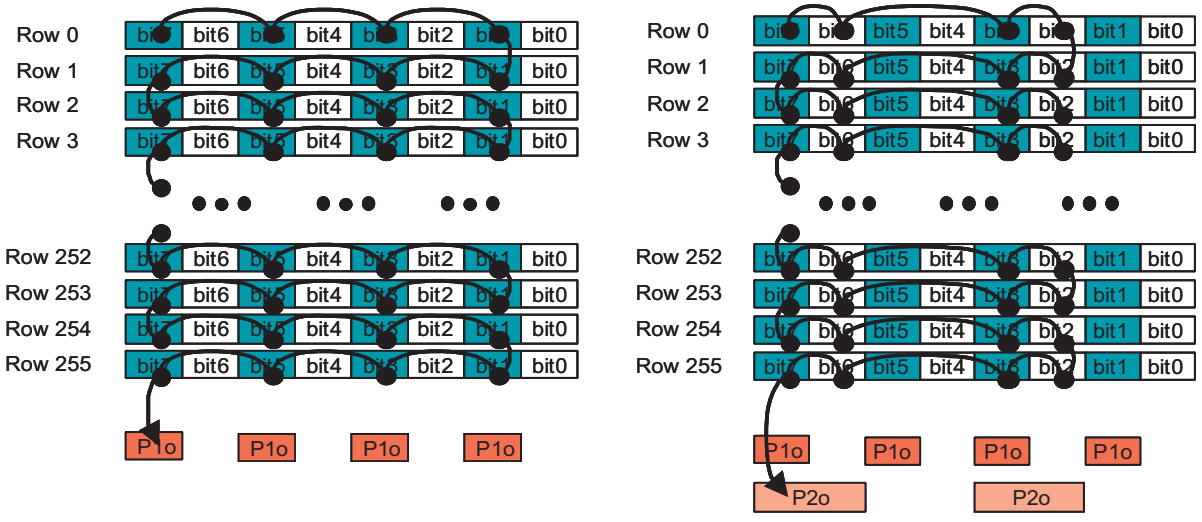
P2e = bit5 XOR bit4 XOR bit1 XOR bit0 on each byte of the data stream

P4o = bit7 XOR bit6 XOR bit5 XOR bit4 on each byte of the data stream

P4e = bit3 XOR bit2 XOR bit1 XOR bit0 on each byte of the data stream

Each column parity bit is XORed with the previous accumulated value.

**Figure 11-31. Hamming Code Accumulation Algorithm (1 of 2)**





Copyright © 2011–2016, Texas Instruments Incorporated

For line parities, the bits of each new data are XORed together, and line parity bits are computed as:

P8e = row0 XOR row2 XOR row4 XOR ... XOR row254

P8o = row1 XOR row3 XOR row5 XOR ... XOR row255

P16e = row0 XOR row1 XOR row4 XOR row5 XOR ... XOR row252 XOR row 253

P16o = row2 XOR row3 XOR row6 XOR row7 XOR ... XOR row254 XOR row 255

Unused parity bits in the result registers are cleared to 0.

**Figure 11-32. Hamming Code Accumulation Algorithm (2 of 2)**

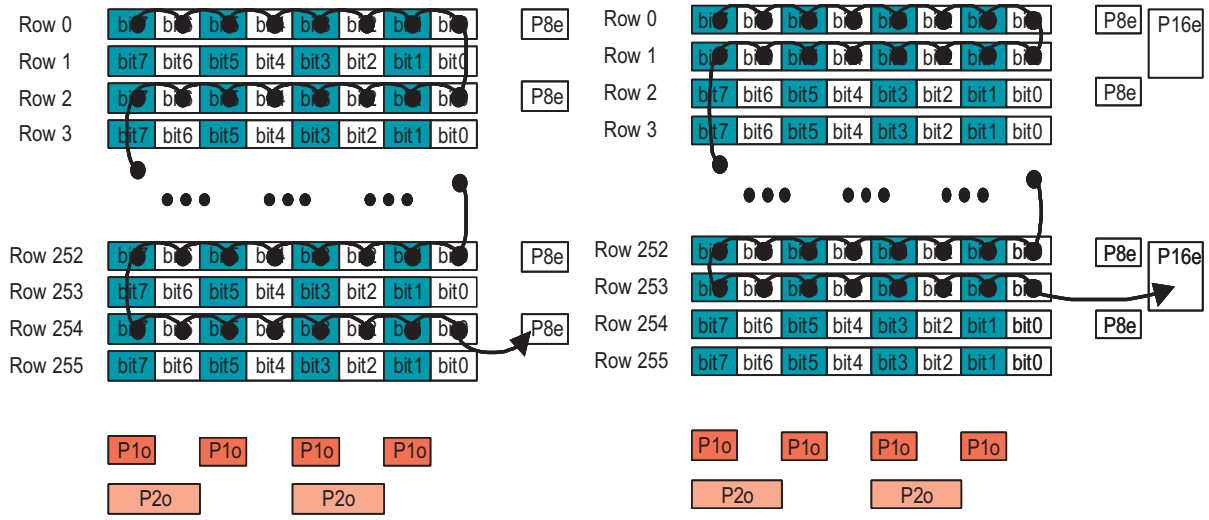
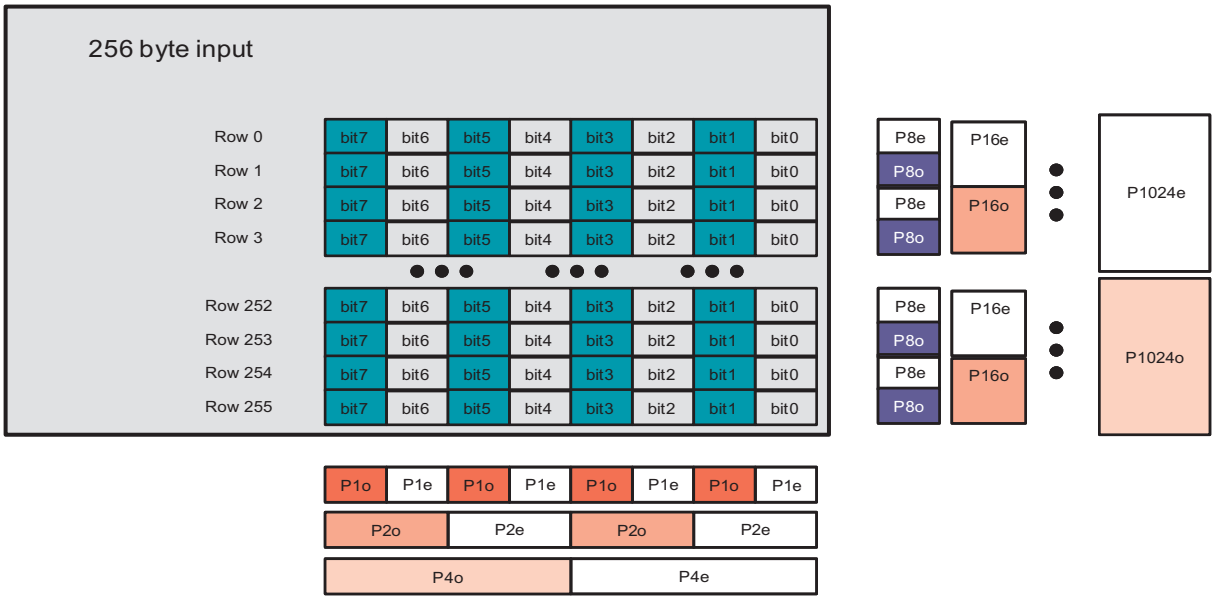



Figure 11-33 shows ECC computation for a 256-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and sixteen row parity bits (P8o-P16o-P32o--P1024o for odd parities, and P8e-P16e-P32e--P1024e for even parities).

**Figure 11-33. ECC Computation for a 256-Byte Data Stream (Read or Write)**



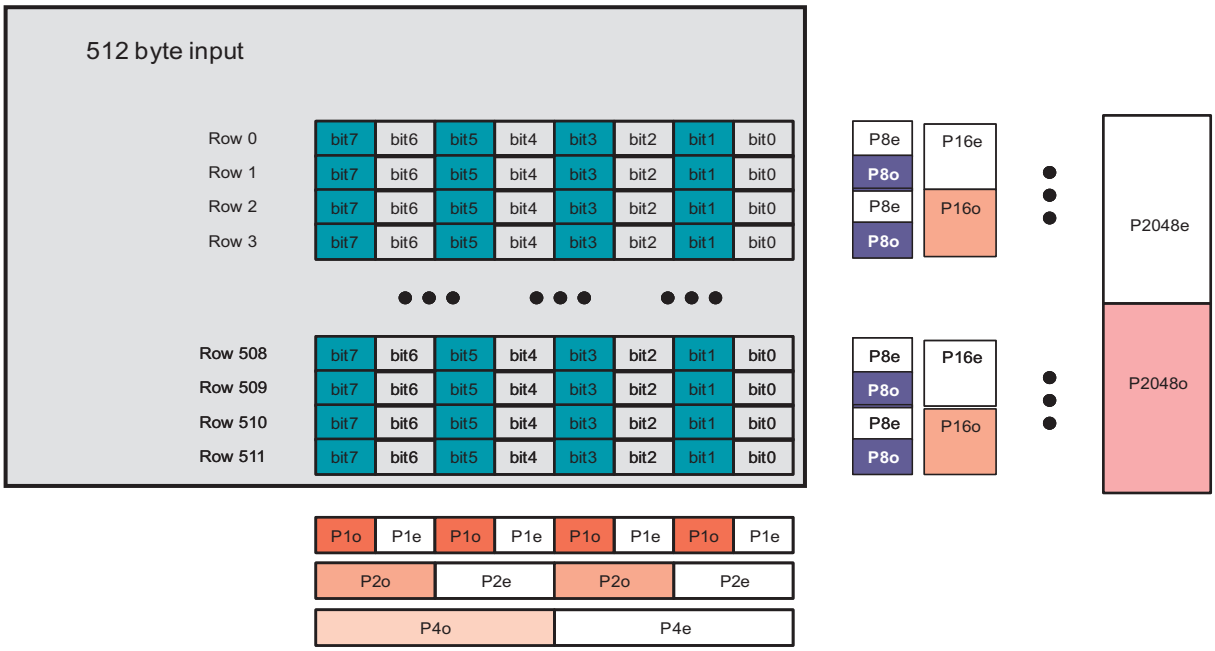



Copyright © 2011–2016, Texas Instruments Incorporated

Figure 11-34 shows ECC computation for a 512-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and eighteen row parity bits (P8o-P16o-P32o--P1024o- - P2048o for odd parities, and P8e-P16e-P32e--P1024e- P2048e for even parities).

For a 2 Kbytes page, four 512 bytes ECC calculations plus one for the spare area are required. Results are stored in the GPMC_ECCj_RESULT registers (j = 1 to 9).

**Figure 11-34. ECC Computation for a 512-Byte Data Stream (Read or Write)**




### 11.2.4.12.3.1.4 ECC Comparison and Correction

To detect an error, the computed ECC result must be XORed with the parity value stored in the spare area of the accessed page.

- If the result of this logical XOR is all 0s, no error is detected and the read data is correct.
- If every second bit in the parity result is a 1, one bit is corrupted and is located at bit address (P2048o, P1024o, P512o, P256o, P128o, P64o, P32o, P16o, P8o, P4o, P2o, P1o). The software must correct the corresponding bit.
- If only one bit in the parity result is 1, it is an ECC error and the read data is correct.

### 11.2.4.12.3.1.5 ECC Calculation Based on 8-Bit Word

The 8-bit based ECC computation is used for 8-bit wide NAND device interfacing.

The 8-bit based ECC computation can be used for 16-bit wide NAND device interfacing to get backward compatibility on the error-handling strategy used with 8-bit wide NAND devices. In this case, the 16-bit wide data read from or written to the NAND device is fragmented into 2 bytes. According to little-endian access, the least significant bit (LSB) of the 16-bit wide data is ordered first in the byte stream used for 8-bit based ECC computation.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.3.1.6 ECC Calculation Based on 16-Bit Word

ECC computation based on a 16-bit word is used for 16-bit wide NAND device interfacing. This ECC computation is not supported when interfacing an 8-bit wide NAND device, and the GPMC_ECC_CONFIG[7] ECC16B bit must be cleared to 0 when interfacing an 8-bit wide NAND device.

The parity computation based on 16-bit words affects the row and column parity mapping. The main difference is that the odd and even parity bits P8o and P8e are computed on rows for an 8-bit based ECC while there are computed on columns for a 16-bit based ECC. Figure 11-35 and Figure 11-36.

**Figure 11-35. 128 Word16 ECC Computation**

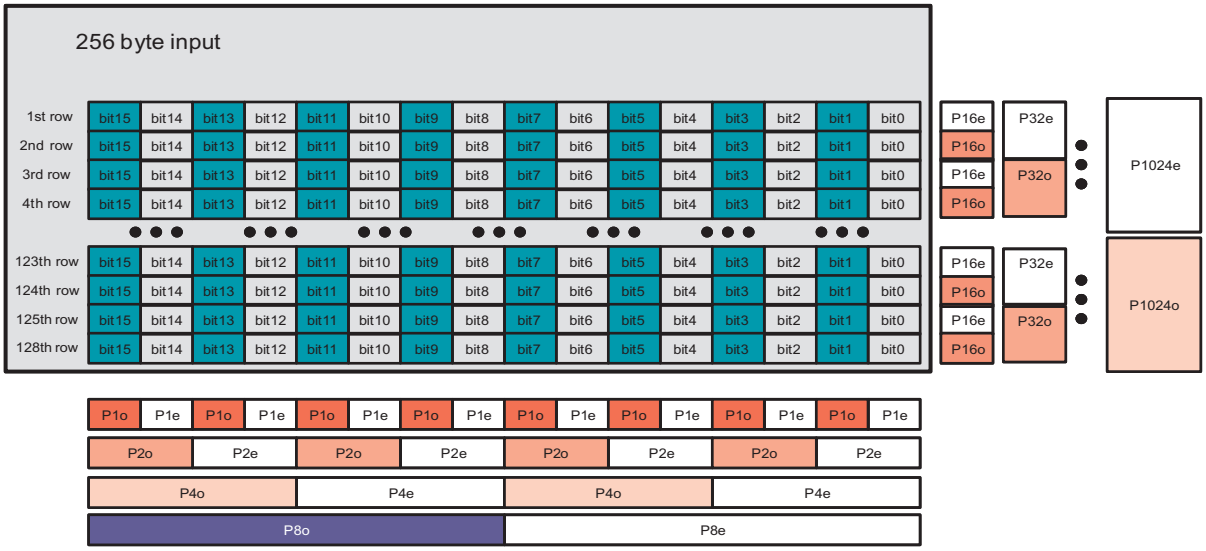
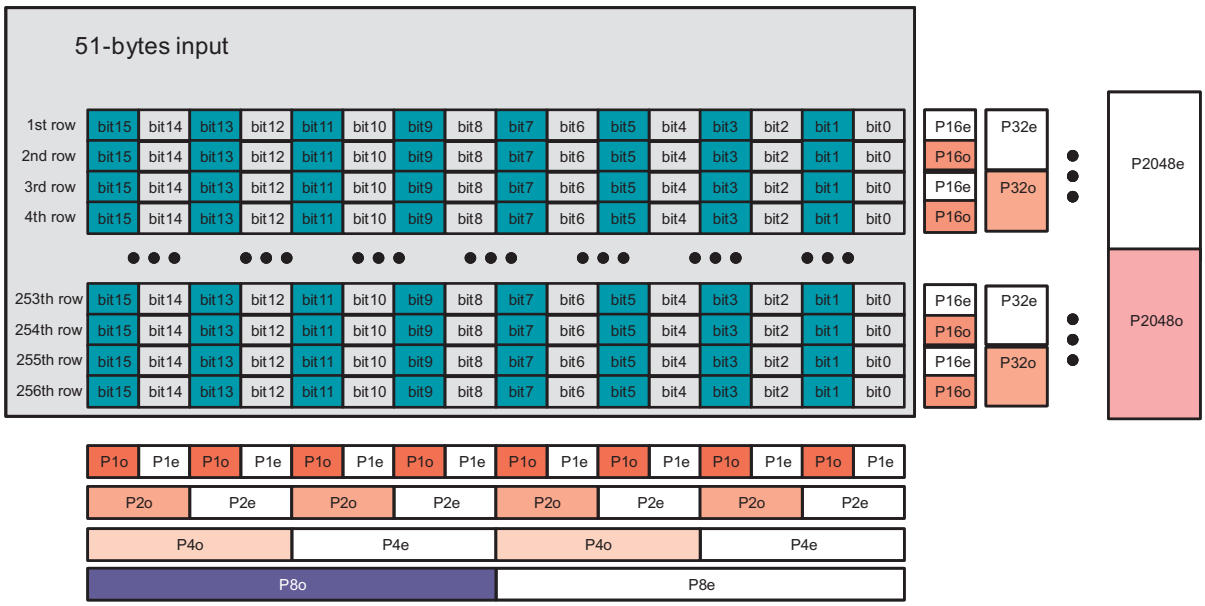



**Figure 11-36. 256 Word16 ECC Computation**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.3.2 BCH Code (Bose-Chaudhurl-Hocquenghem)

All references to Error Code Correction (ECC) in this subsection refer to the 4- to 16-bit error correction BCH code.

##### 11.2.4.12.3.2.1 Requirements

Read and write accesses to a NAND flash take place by whole pages, in a predetermined sequence: first the data byte page itself, then some spare bytes, including the BCH ECC (and other information). The NAND IC can cache a full page, including spares, for read and write accesses.

Typical page write sequence:

- Sequential write to NAND cache of main data + spare data, for a page. ECC is calculated on the fly. Calculated ECC may be inserted on the fly in the spares, or replaced by dummy accesses.
- When the calculated ECC is replaced by dummy accesses, it must be written to the cache in a second, separate phase. The ECC module is disabled during that time.
- NAND writes its cache line (page) to the array

Typical page read sequence:

- Sequential read of a page. ECC is calculated on the fly.
- ECC module buffers status determines the presence of errors.
- Accesses to several memories may be interleaved by the GPMC, but only one of those memories can be a NAND using the BCH engine at a time; in other words, only one BCH calculation (for example, for a single page) can be on-going at any time. Note also that the sequential nature of NAND accesses guarantees that the data is always written / read out in the same order. BCH-relevant accesses are selected by the GPMCs chip-select.
- Each page may hold up to 4 Kbytes of data, spare bytes not included. This means up to 8 x 512-byte BCH messages. Since all the data is written / read out first, followed by the BCH ECC, this means that the BCH engine must be able to hold 8 104-bit remainders or syndromes (or smaller, 52-bit ones) at the same time.

The BCH module has the capacity to store all remainders internally. After the page start, an internal counter is used to detect the 512-byte sector boundaries. On those boundaries, the current remainder is stored and the divider reset for the next calculation. At the end of the page, the BCH module contains all remainders.

- NAND access cycles hold 8 or 16 bits of data each (1 or 2 bytes); Each NAND cycle takes at least 4 cycles of the GPMCs internal clock. This means the NAND flash timing parameters must define a RDCYCLETIME and a WRCYCLETIME of at least 4 clock cycles after optimization when using the BCH calculator.
- The spare area is assumed to be large enough to hold the BCH ECC, that is, to have at least a message of 13 bytes available per 512-byte sector of data. The zone of unused spare area by the ECC may or may not be protected by the same ECC scheme, by extending the BCH message beyond 512 bytes (maximum codeword is 1023-byte long, ECC included, which leaves a lot of space to cover some spares bytes).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.3.2.2 Memory-Mapping of the BCH Codeword

BCH encoding considers a block of data to protect as a polynomial message M(x). In our standard case, 512 bytes of data (that is, 2 bits = 4096 bits) are seen as a polynomial of degree 2 - 1 = 4095, with parameters ranging from M0 to M4095. For 512 bytes of data, 52 bits are required for 4-bit error correction, and 104 bits are required for 8-bit error correction and 207 bits are required for 16-bit error correction. The ECC is a remainder polynomial R(x) of degree 103 (or 51, depending on the selected mode). The complete codeword C(x) is the concatenation of M(x) and R(x) as shown in Table 11-13.

**Table 11-13. Flattened BCH Codeword Mapping (512 Bytes + 104 Bits)**

| | Message M(x) | ECC R(x) | | | | |
|---|---|---|---|---|---|---|
| **Bit number** | M4095 | ... | M0 | R103 | ... | R0 |

If the message is extended by the addition of spare bytes to be protected by the same ECC, the principle is still valid. For example, a 3-byte extension of the message gives a polynomial message M(x) of degree ((512 + 3) × 8) - 1 = 4119, for a total of 3 + 13 = 16 spare bytes of spare, all protected as part of the same codeword.

The message and the ECC bits are manipulated and mapped in the GPMC byte-oriented system. The ECC bits are stored in , , , and (where i = 0 to 5).

- GPMC_BCH_RESULT0_i
- GPMC_BCH_RESULT1_i
- GPMC_BCH_RESULT2_i
- GPMC_BCH_RESULT3_i

#### 11.2.4.12.3.2.3 Memory Mapping of the Data Message

The data message mapping shall follow the following rules:

- Bit endianness within a byte is little-endian, that is, the bytes LS bit is also the lowest-degree polynomial parameter: a byte b7-b0 (with b0 the LS bit) represents a segment of polynomial b7 × x + b6 × x + ... + b0 × x
- The message is mapped in the NAND starting with the highest-order parameters, that is, in the lowest addresses of a NAND page.
- Byte endianness within the NANDs 16-bit words is big endian. This means that the same message mapped in 8- and 16-bit memories has the same content at the same byte address.

The BCH module has no visibility over actual addresses. The most important point is the sequence of data word the BCH sees. However, the NAND page is always scanned incrementally in read and write accesses, and this produces the mapping patterns described in the following.

Table 11-14 and Table 11-15 show the mapping of the same 512-byte vector (typically a BCH message) in the NAND memory space. Note that the byte 'address' is only an offset modulo 512 (200h), since the same page may contain several contiguous 512-byte sectors (BCH blocks). The LSB and MSB are respectively the bits M0 and M(2^12-1) of the codeword mapping given above. In both cases the data vectors are aligned, that is, their boundaries coincide with the RAMs data word boundaries.

**Table 11-14. Aligned Message Byte Mapping in 8-bit NAND**

| Byte Offset | 8-Bit Word |
|---|---|
| 0 | (msb) Byte 511 (1FFh) |
| 1h | Byte 510 (1FEh) |
| ⋮ | ⋮ |
| 1FFh | Byte 0 (0) (LSB) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-15. Aligned Message Byte Mapping in 16-bit NAND**

| Byte Offset | 16-Bit Words MSB | 16-Bit Words LSB |
|---|---|---|
| 0 | Byte 510 (1FEh) | (msb) Byte 511 (1FFh) |
| 2h | Byte 508 (1FCh) | Byte 509 (1FDh) |
| ⋮ | ⋮ | ⋮ |
| 1FEh | Byte 0 (0) | (lsb) Byte 1 (1) |

Table 11-16 and Table 11-17 show the mapping in memory of arbitrarily-sized messages, starting on access (byte or 16-bit word) boundaries for more clarity. Note that message may actually start and stop on arbitrary nibbles. A nibble is a 4-bit entity. The unused nibbles are not discarded, and they can still be used by the BCH module, but as part of the next message section (for example, on another sectors ECC).

**Table 11-16. Aligned Nibble Mapping of Message in 8-bit NAND**

| Byte Offset | 8-Bit Word | |
|---|---|---|
| | 4-Bit Most Significant Nibble | 4-Bit Less Significant Nibble |
| 1 | (MSB) Nibble S-1 | Nibble S-2 |
| 2 | Nibble S-3 | Nibble S-4 |
| ⋮ | ⋮ | ⋮ |
| S/2 - 2 | Nibble 3 | Nibble 2 |
| S/2 - 1 | Nibble 1 | Nibble 0 (LSB) |

**Table 11-17. Misaligned Nibble Mapping of Message in 8-bit NAND**

| Byte Offset | 8-Bit Word | |
|---|---|---|
| | 4-Bit Most Significant Nibble | 4-Bit Less Significant Nibble |
| 1 | (MSB) Nibble S-1 | Nibble S-2 |
| 2 | Nibble S-3 | Nibble S-4 |
| ⋮ | ⋮ | ⋮ |
| (S+1)/2 - 2 | Nibble 2 | Nibble 1 |
| (S+1)/2 - 1 | Nibble 0 (LSB) | |

**Table 11-18. Aligned Nibble Mapping of Message in 16-bit NAND**

| Byte Offset | 16-Bit Word | | | |
|---|---|---|---|---|
| | 4-Bit Most Significant Nibble | | 4-Bit Less Significant Nibble | |
| 0 | Nibble S-3 | Nibble S-4 | (MSB) Nibble S-1 | Nibble S-2 |
| 2 | Nibble S-7 | Nibble S-8 | Nibble S-5 | Nibble S-6 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| S/2 - 4 | Nibble 5 | Nibble 4 | Nibble 7 | Nibble 6 |
| S/2 - 2 | Nibble 1 | Nibble 0 (LSB) | Nibble 3 | Nibble 2 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-19. Misaligned Nibble Mapping of Message in 16-bit NAND (1 Unused Nibble)**

| Byte Offset | 16-Bit Word | | | |
|---|---|---|---|---|
| | 4-Bit Most Significant Nibble | | 4-Bit Less Significant Nibble | |
| 0 | Nibble S-3 | Nibble S-4 | (MSB) Nibble S-1 | Nibble S-2 |
| 2 | Nibble S-7 | Nibble S-8 | Nibble S-5 | Nibble S-6 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| (S+1)/2 - 4 | Nibble 4 | Nibble 3 | Nibble 6 | Nibble 5 |
| (S+1)/2 - 2 | Nibble 0 (LSB) | | Nibble 2 | Nibble 1 |

**Table 11-20. Misaligned Nibble Mapping of Message in 16-bit NAND (2 Unused Nibble)**

| Byte Offset | 16-Bit Word | | | |
|---|---|---|---|---|
| | 4-Bit Most Significant Nibble | | 4-Bit Less Significant Nibble | |
| 0 | Nibble S-3 | Nibble S-4 | (MSB) Nibble S-1 | Nibble S-2 |
| 2 | Nibble S-7 | Nibble S-8 | Nibble S-5 | Nibble S-6 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| (S+2)/2 - 4 | Nibble 3 | Nibble 2 | Nibble 5 | Nibble 4 |
| (S+2)/2 - 2 | | | Nibble 1 | Nibble 0 (LSB) |

**Table 11-21. Misaligned Nibble Mapping of Message in 16-bit NAND (3 Unused Nibble)**

| Byte Offset | 16-Bit Word | | | |
|---|---|---|---|---|
| | 4-Bit Most Significant Nibble | | 4-Bit Less Significant Nibble | |
| 0 | Nibble S-3 | Nibble S-4 | (MSB) Nibble S-1 | Nibble S-2 |
| 2 | Nibble S-7 | Nibble S-8 | Nibble S-5 | Nibble S-6 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| (S+3)/2 - 4 | Nibble 2 | Nibble 1 | Nibble 4 | Nibble 3 |
| (S+3)/2 - 2 | | | Nibble 0 (LSB) | |

Note that many other cases exist than the ones represented above, for example, where the message does not start on a word boundary.

#### *11.2.4.12.3.2.4 Memory Mapping of the ECC*

The ECC (or remainder) is presented by the BCH module as a single 104-bit (or 52-bit), little-endian vector. It is up to the software to fetch those 13 bytes (or 6 bytes) from the modules interface, then store them to the NANDs spare area (page write) or to an intermediate buffer for comparison with the stored ECC (page read). There are no constraints on the ECC mapping inside the spare area: it is a softwarecontrolled operation.

However, it is advised to maintain a coherence in the respective formats of the message or the ECC remainder once they have been read out of the NAND. The error correction algorithm works from the complete codeword (concatenated message and remainder) once an error as been detected. The creation of this codeword should be made as straightforward as possible.


Copyright © 2011–2016, Texas Instruments Incorporated

There are cases where the same NAND access contains both data and the ECC protecting that data. This is the case when the data/ECC boundary (which can be on any nibble) does not coincide with an access boundary. The ECC is calculated on-the-fly following the write. In that case, the write must also contain part of the ECC because it is impossible to insert the ECC on-the-fly. Instead:

- During the initial page write (BCH encoding), the ECC is replaced by dummy bits. The BCH encoder is by definition turned OFF during the ECC section, so the BCH result is unmodified.
- During a second phase, the ECC is written to the correct location, next to the actual data.
- The completed line buffer is then written to the NAND array.

#### 11.2.4.12.3.2.5 Wrapping Modes

For a given wrapping mode, the module automatically goes through a specific number of sections, as data is being fed into the module. For each section, the BCH core can be enabled (in which case the data is fed to the BCH divider) or not (in which case the BCH simply counts to the end of the section). When enabled, the data is added to the ongoing calculation for a given sector number (for example, number 0).

Wrapping modes are described below. To get a better understanding and see the real-life read and write sequences implemented with each mode, see Section 11.2.4.12.3.3.

For each mode:

- A sequence describes the mode in pseudo-language, with for each section the size and the buffer used for ECC processing (if ON). The programmable lengths are size, size0 and size1.
- A checksum condition is given. If the checksum condition is not respected for a given mode, the modules behavior is unpredictable. S is the number of sectors in the page; size0 and size1 are the section sizes programmed for the mode, in nibbles.

Note that wrapping modes 8, 9, 10, and 11 insert a 1-nibble padding where the BCH processing is OFF. This is intended for t = 4 ECC, where ECC is 6 bytes long and the ECC area is expected to include (at least) 1 unused nibble to remain byte-aligned.

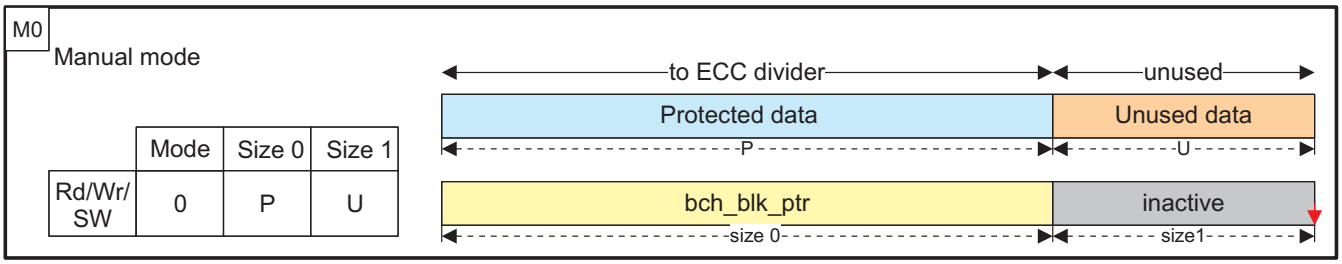
#### 11.2.4.12.3.2.6 Manual Mode (0x0)

This mode is intended for short sequences, added manually to a given buffer through the software data port input. A complete page may be built out of several such sequences.

To process an arbitrary sequence of 4-bit nibbles, accesses to the software data port shall be made, containing the appropriate data. If the sequence end does not coincide with an access boundary (for example, to process 5 nibbles = 20 bits in 16-bit access mode) and those nibbles need to be skipped, a number of unused nibbles shall be programmed in size1 (in the same example: 5 nibbles to process + 3 to discard = 8 nibbles = exactly 2 x 16-bit accesses: we must program size0 = 5, size1 = 3).

Figure 11-37 shows the manual mode sequence and mapping. In this figure, size and size0 are the same parameter.

**Figure 11-37. Manual Mode Sequence and Mapping**





Copyright © 2011–2016, Texas Instruments Incorporated

Section processing sequence:

- One time with buffer
  - size0 nibbles of data, processing ON
  - size1 nibbles of unused data, processing OFF

Checksum: size0 + size1 nibbles must fit in a whole number of accesses.

In the following sections, S is the number of sectors in the page.

#### 11.2.4.12.3.2.7 Mode 0x1

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + size1)

#### 11.2.4.12.3.2.8 Mode 0xA (10)

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - 1 nibble pad spare, processing OFF
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + 1 + size1)

#### 11.2.4.12.3.2.9 Mode 0x2

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + size1)

#### 11.2.4.12.3.2.10 Mode 0x3

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - size1)


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.3.2.11 Mode 0x7

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = size0 + (S - size1)

#### 11.2.4.12.3.2.12 Mode 0x8

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - (1 + size1))

#### 11.2.4.12.3.2.13 Mode 0x4

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)
  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - size1)

#### 11.2.4.12.3.2.14 Mode 0x9

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)
  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - (1 + size1))


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.3.2.15 Mode 0x5

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + size1)

#### 11.2.4.12.3.2.16 Mode 0xB (11)

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + 1 + size1)

#### 11.2.4.12.3.2.17 Mode 0x6

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + size1)


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.3.3 Supported NAND Page Mappings and ECC Schemes

The following rules apply throughout the entire mapping description:

- Main data area (sectors) size is hardcoded to 512 bytes.
- Spare area size is programmable.
- All page sections (of main area data bytes, protected spare bytes, unprotected spare bytes, and ECC) are defined as explained in Section 11.2.4.12.3.2.3.

Each one of the following sections shows a NAND page mapping example (per-sector spare mappings, pooled spare mapping, per-sector spare mapping, with ECC separated at the end of the page).

In the mapping diagrams, sections that belong to the same BCH codeword have the same color (blue or green); unprotected sections are not covered (orange) by the BCH scheme.

Below each mapping diagram, a write (encoding) and read (decoding: syndrome generation) sequence is given, with the number of the active buffers at each point in time (yellow). In the inactive zones (grey), no computing is taking place but the data counter is still active.

In Figure 11-38 to Figure 11-40, tables on the left summarize the mode, size0, size1 parameters to program for respectively write and read processing of a page, with the given mapping, where:

- P is the size of spare byte section Protected by the ECC (in nibbles)
- U is the size of spare byte section Unprotected by the ECC (in nibbles)
- E is the size of the ECC itself (in nibbles)
- S is the number of Sectors per page (2 in the current diagrams)

Each time the processing of a BCH block is complete (ECC calculation for write/encoding, syndrome generation for read/decoding, indicated by red arrows), the update pointer is pulsed. Note that the processing for block 0 can be the first or the last to complete, depending on the NAND page mapping and operation (read or write). All examples show a page size of 1kByte + spares, that is, S = 2 sectors of 512 bytes. The same principles can be extended to larger pages by adding more sectors.

The actual BCH codeword size is used during the error location work to restrict the search range: by definition, errors can only happen in the codeword that was actually written to the NAND, and not in the mathematical codeword of n = 2 - 1 = 8191 bits. That codeword (higher-order bits) is all-zero and implicit during computations.

The actual BCH codeword size depends on the mode, on the programmed sizes and on the sector number (all sizes in nibbles):

- Spares mapped and protected per sector (Figure 11-38: see M1-M2-M3-M9-M10):
  - all sectors: (512) + P + E
- Spares pooled and protected by sector 0 (Figure 11-38: see M5-M6):
  - sector 0 codeword: (512) + P + E
  - other sectors: (512) + E
- Unprotected spares (Figure 11-38: see M4-M7-M8-M11-M12):
  - all codewords (512) + E


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.3.3.1 Per-Sector Spare Mappings

In these schemes (Figure 11-38), each 512-byte sector of the main area has its own dedicated section of the spare area. The spare area of each sector is composed of:

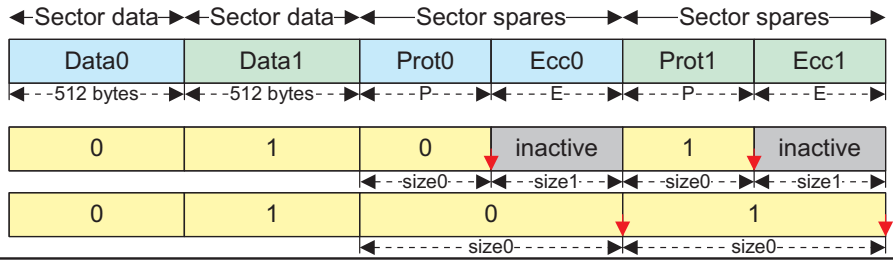
- ECC, which must be located after the data it protects
- other data, which may or may not be protected by the sectors ECC

**Figure 11-38. NAND Page Mapping and ECC: Per-Sector Schemes**

M1

Per-sector spares
Spares covered by sector ECC per sector ECC mapping.

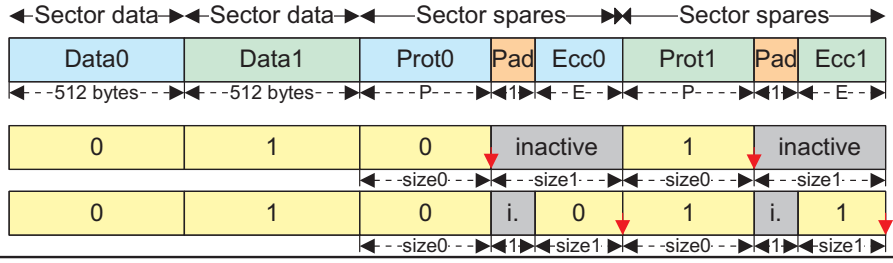
| | Mode | Size0 | Size1 |
|---|---|---|---|
| Write | 1 | P | E |
| Read | 1 | P+E | 0 |




M2

Per-sector spares
Spares covered by sector ECC per sector, left-padded ECC.

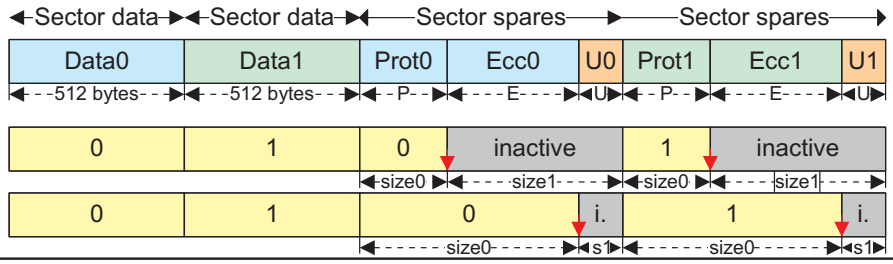
| | Mode | Size0 | Size1 |
|---|---|---|---|
| Write | 1 | P | 1+E |
| Read | 10 | P | E |




M3

Per-sector spares
Spares covered by sector ECC, ECC not right-aligned.

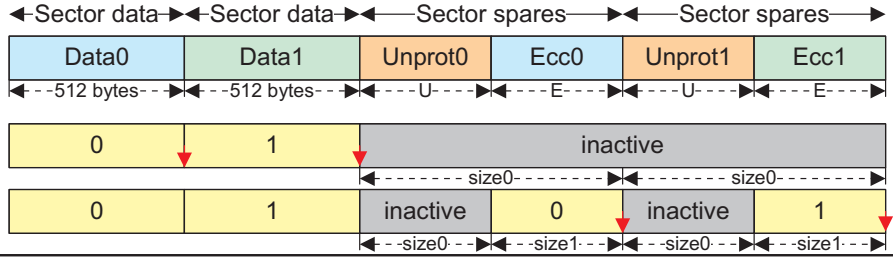
| | Mode | Size0 | Size1 |
|---|---|---|---|
| Write | 1 | P | E+U |
| Read | 1 | P+E | U |




M4

Per-sector spares
Spares not covered by ECC, ECC right-aligned per sector.

| | Mode | Size0 | Size1 |
|---|---|---|---|
| Write | 2 | U+E | 0 |
| Read | 2 | U | E |





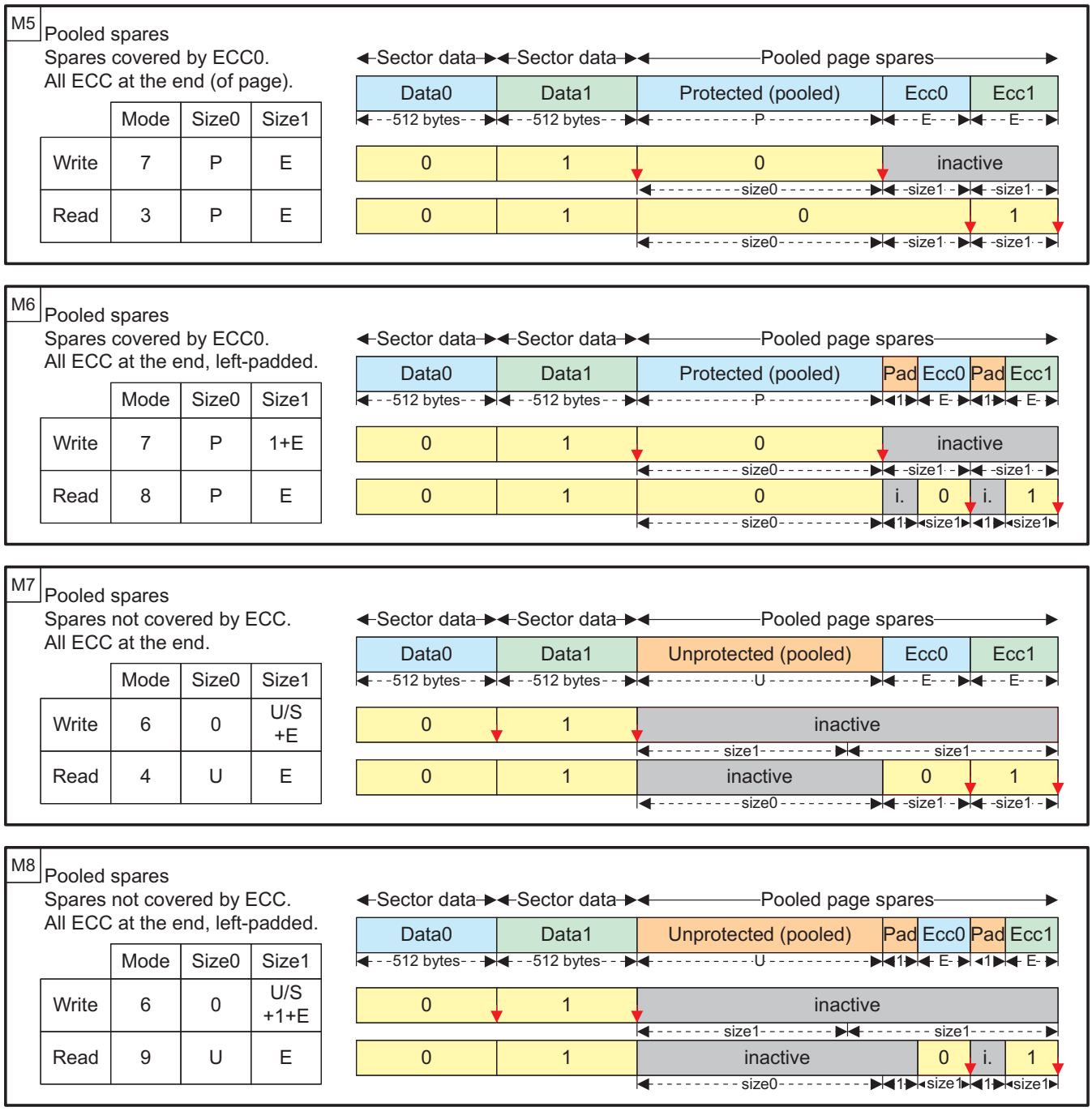

Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.3.3.2 Pooled Spare Mapping

In these schemes (Figure 11-39), the spare area is pooled for the page.

- The ECC of each sector is aligned at the end of the spare area.
- The non-ECC spare data may or may not be covered by the ECC of sector 0

**Figure 11-39. NAND Page Mapping and ECC: Pooled Spare Schemes**





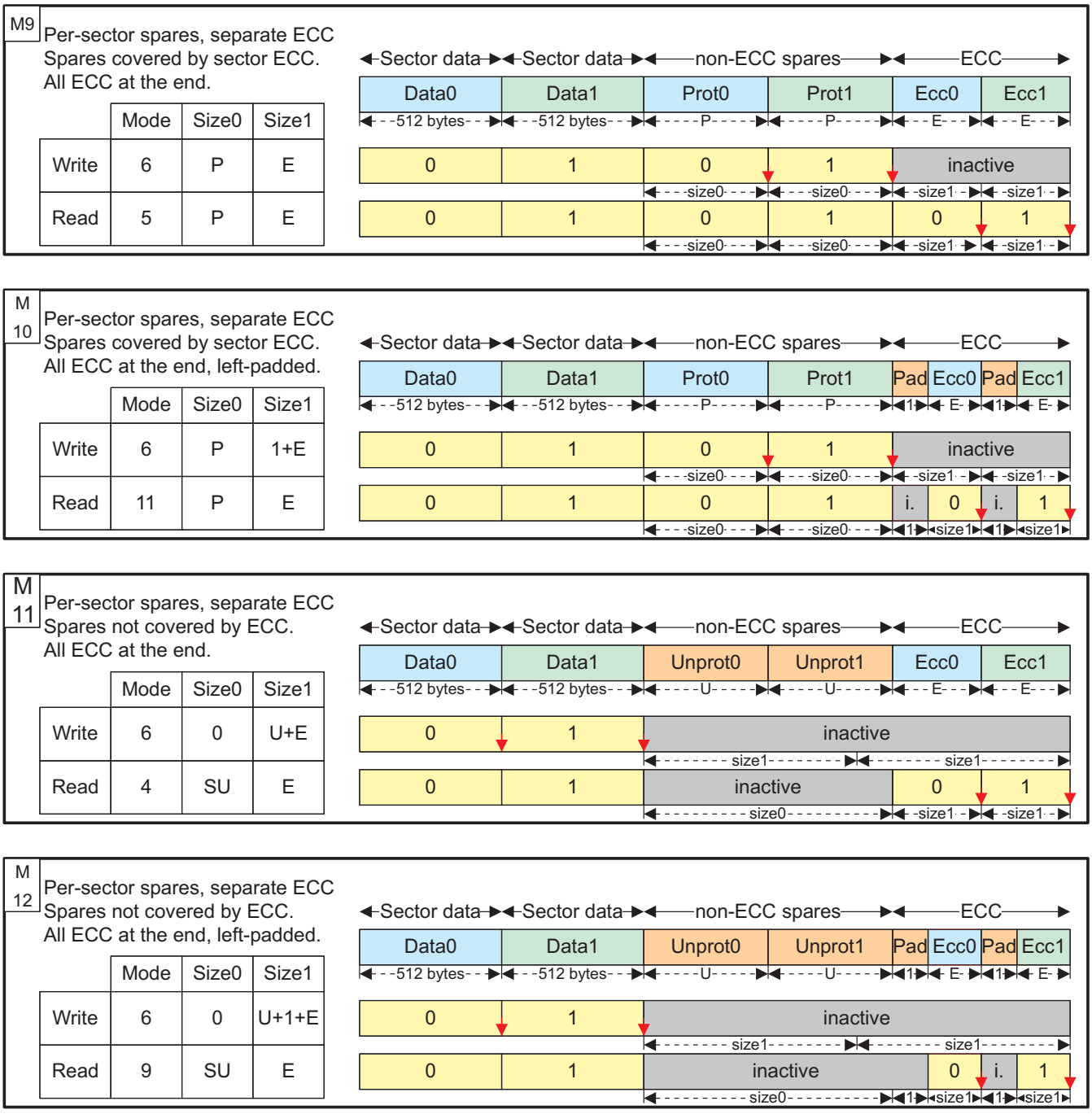
Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.3.3.3 Per-Sector Spare Mapping, With ECC Separated at the End of the Page

In these schemes (Figure 11-40), each 512-byte sector of the main area is associated with two sections of the spare area.

- ECC section, all aligned at the end of the page
- other data section, aligned before the ECCs, each of which may or may not be protected by its sectors ECC

**Figure 11-40. NAND Page Mapping and ECC: Per-Sector Schemes, with Separate ECC**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.4 Prefetch and Write-Posting Engine

NAND device data access cycles are usually much slower than the MPU system frequency; such NAND read or write accesses issued by the processor will impact the overall system performance, especially considering long read or write sequences required for NAND page loading or programming. To minimize this effect on system performance, the GPMC includes a prefetch and write-posting engine, which can be used to read from or write to any chip-select location in a buffered manner.

The prefetch and write-posting engine is a simplified embedded-access requester that presents requests to the access engine on a user-defined chip-select target. The access engine interleaves these requests with any request coming from the L3 interface; as a default the prefetch and write-posting engine has the lowest priority.

The prefetch and write-posting engine is dedicated to data-stream access (as opposed to random data access); thus, it is primarily dedicated to NAND support. The engine does not include an address generator; the request is limited to chip-select target identification. It includes a 64-byte FIFO associated with a DMA request synchronization line, for optimal DMA-based use.

The prefetch and write-posting engine uses an embedded 64 bytes (32 16-bit word) FIFO to prefetch data from the NAND device in read mode (prefetch mode) or to store host data to be programmed into the NAND device in write mode (write-posting mode). The FIFO draining and filling (read and write) can be controlled either by the MPU through interrupt synchronization (an interrupt is triggered whenever a programmable threshold is reached) or the sDMA through DMA request synchronization, with a programmable request byte size in both prefetch or posting mode.

The prefetch and write-posting engine includes a single memory pool. Therefore, only one mode, read or write, can be used at any given time. In other words, the prefetch and write-posting engine is a single-context engine that can be allocated to only one chip-select at a time for a read prefetch or a write-posting process.

The engine does not support atomic command and address phase programming and is limited to linear memory read or write access. In consequence, it is limited to NAND data-stream access. The engine relies on the MPU NAND software driver to control block and page opening with the correct data address pointer initialization, before the engine can read from or write to the NAND memory device.

Once started, the engine data reads and writes sequencing is solely based on FIFO location availability and until the total programmed number of bytes is read or written.

Any host-concurrent accesses to a different chip-select are correctly interleaved with ongoing engine accesses. The engine has the lowest priority access so that host accesses to a different chip-select do not suffer a large latency.

A round-robin arbitration scheme can be enabled to ensure minimum bandwidth to the prefetch and write-posting engine in the case of back-to-back direct memory requests to a different chip-select. If the GPMC_PREFETCH_CONFIG1[23] PFPWENROUNDROBIN bit is enabled, the arbitration grants the prefetch and write posting engine access to the GPMC bus for a number of requests programmed in the GPMC_PREFETCH_CONFIG1[19-16] PFPWWEIGHTEDPRIO field.

The prefetch/write-posting engine read or write request is routed to the access engine with the chip-select destination ID. After the required arbitration phase, the access engine processes the request as a single access with the data access size equal to the device size specified in the corresponding chip-select configuration.

The destination chip-select configuration must be set to the NAND protocol-compatible configuration for which address lines are not used (the address bus is not changed from its current value). Selecting a different chip-select configuration can produce undefined behavior.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.4.1 General Facts About the Engine Configuration

The engine can be configured only if the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit is de-asserted.

The engine must be correctly configured in prefetch or write-posting mode and must be linked to a NAND chip-select before it can be started. The chip-select is linked using the GPMC_PREFETCH_CONFIG1[26-24] ENGINECSSELECTOR field.

In both prefetch and write-posting modes, the engine respectivelly uses byte or 16-bit word access requests for an 8- or 16-bit wide NAND device attached to the linked chip-select. The FIFOTHRESHOLD and TRANSFERCOUNT fields must be programmed accordingly as a number of bytes or a number of 16-bit word.

When the GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE bit is set, the FIFO entry on the L3 interconnect port side is accessible at any address in the associated chip-select memory region. When the ENABLEENGINE bit is set, any host access to this chip-select is rerouted to the FIFO input. Directly accessing the NAND device linked to this chip-select from the host is still possible through these registers (where i = 0 to 5):

- GPMC_NAND_COMMAND_i
- GPMC_NAND_ADDRESS_i
- GPMC_NAND_DATA_i

The FIFO entry on the L3 interconnect port can be accessed with Byte, 16-bit word, or 32-bit word access size, according to little-endian format, even though the FIFO input is 32-bit wide.

The FIFO control is made easier through the use of interrupts or DMA requests associated with the FIFOTHRESHOLD bit field. The GPMC_PREFETCH_STATUS[30-24] FIFOPOINTER field monitors the number of available bytes to be read in prefetch mode or the number of free empty slots which can be written in write-posting mode. The GPMC_PREFETCH_STATUS[13-0] COUNTVALUE field monitors the number of remaining bytes to be read or written by the engine according to the TRANSFERCOUNT value. The FIFOPOINTER and COUNTVALUE bit fields are always expressed as a number of bytes even if a 16-bit wide NAND device is attached to the linked chip-select.

In prefetch mode, when the FIFOPOINTER equals 0, that is, the FIFO is empty, a host read access receives the byte last read from the FIFO as its response. In case of 32-bit word or 16-bit word read accesses, the last byte read from the FIFO is copied the required number of times to fit the requested word size. In write-posting mode, when the FIFOPOINTER equals 0, that is, the FIFO is full, a host write overwrites the last FIFO byte location. There is no underflow or overflow error reporting in the GPMC.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.4.2 Prefetch Mode

The prefetch mode is selected when the GPMC_PREFETCH_CONFIG1[0] ACCESSMODE bit is cleared.

The MPU NAND software driver must issue the block and page opening (READ) command with the correct data address pointer initialization before the engine can be started to read from the NAND memory device. The engine is started by asserting the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit. The STARTENGINE bit automatically clears when the prefetch process completes.

If required, the ECC calculator engine must be initialized (i.e., reset, configured, and enabled) before the prefetch engine is started, so that the ECC is correctly computed on all data read by the prefetch engine.

When the GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE bit is cleared, the prefetch engine starts requesting data as soon as the STARTENGINE bit is set. If using this configuration, the host must monitor the NAND device-ready pin so that it only sets the STARTENGINE bit when the NAND device is in a ready state, meaning data is valid for prefetching.

When the SYNCHROMODE bit is set, the prefetch engine starts requesting data when an active to inactive wait signal transition is detected. The transition detector must be cleared before any transition detection; see Section 11.2.4.12.2.2. The GPMC_PREFETCH_CONFIG1[5-4] WAITPINSELECTOR field selects which gpmc_wait pin edge detector triggers the prefetch engine in this synchronized mode.

If the STARTENGINE bit is set after the NAND address phase (page opening command), the engine is effectively started only after the actual NAND address phase completion. To prevent GPMC stall during this NAND address phase, set the STARTENGINE bit field before NAND address phase completion when in synchronized mode. The prefetch engine will start when an active to inactive wait signal transition is detected. The STARTENGINE bit is automatically cleared on prefetch process completion.

The prefetch engine issues a read request to fill the FIFO with the amount of data specified by GPMC_PREFETCH_CONFIG2[13-0] TRANSFERCOUNT field.

Table 11-22 describes the prefetch mode configuration.

**Table 11-22. Prefetch Mode Configuration**

| Bit Field | Register | Value | Comments |
|---|---|---|---|
| STARTENGINE | GPMC_PREFETCH_CONTROL | 0 | Prefetch engine can be configured only if STARTENGINE is cleared to 0. |
| ENGINECSSELECTOR | GPMC_PREFETCH_CONFIG1 | 0 to 3h | Selects the chip-select associated with a NAND device where the prefetch engine is active. |
| ACCESSMODE | GPMC_PREFETCH_CONFIG1 | 0 | Selects prefetch mode |
| FIFOTHRESHOLD | GPMC_PREFETCH_CONFIG1 | | Selects the maximum number of bytes read or written by the host on DMA or interrupt request |
| TRANSFERCOUNT | GPMC_PREFETCH_CONFIG1 | | Selects the number of bytes to be read or written by the engine to the selected chip-select |
| SYNCHROMODE | GPMC_PREFETCH_CONFIG1 | 0/1 | Selects when the engine starts the access to the chip-select |
| WAITPINSELECT | GPMC_PREFETCH_CONFIG1 | 0 to 1 | Selects wait pin edge detector (if GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE = 1) |
| ENABLEOPTIMIZEDACCESS | GPMC_PREFETCH_CONFIG1 | 0/1 | See Section 11.2.4.12.4.6 |
| CYCLEOPTIMIZATION | GPMC_PREFETCH_CONFIG1 | | Number of clock cycle removed to timing parameters |
| ENABLEENGINE | GPMC_PREFETCH_CONFIG1 | 1 | Engine enabled |
| STARTENGINE | GPMC_PREFETCH_CONFIG1 | 1 | Starts the prefetch engine |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.4.3 FIFO Control in Prefetch Mode

The FIFO can be drained directly by the MPU or by an eDMA channel.

In MPU draining mode, the FIFO status can be monitored through the GPMC_PREFETCH_STATUS[30-24] FIFOPOINTER field or through the GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bit. The FIFOPOINTER indicates the current number of available data to be read; FIFOTHRESHOLDSTATUS set to 1 indicates that at least FIFOTHRESHOLD bytes are available from the FIFO.

An interrupt can be triggered by the GPMC if the GPMC_IRQENABLE[0] FIFOEVENTENABLE bit is set. The FIFO interrupt event is logged, and the GPMC_IRQSTATUS[0] FIFOEVENTSTATUS bit is set. To clear the interrupt, the MPU must read all the available bytes, or at least enough bytes to get below the programmed FIFO threshold, and the FIFOEVENTSTATUS bit must be cleared to enable further interrupt events. The FIFOEVENTSTATUS bit must always be reset prior to asserting the FIFOEVENTENABLE bit to clear any out-of-date logged interrupt event. This interrupt generation must be enabled after enabling the STARTENGINE bit.

Prefetch completion can be monitored through the GPMC_PREFETCH_STATUS[13-0] COUNTVALUE field. COUNTVALUE indicates the number of currently remaining data to be requested according to the TRANSFERCOUNT value. An interrupt can be triggered by the GPMC when the prefetch process is complete (that is, COUNTVALUE equals 0) if the GPMC_IRQENABLE[1] TERMINALCOUNTEVENTENABLE bit is set. At prefetch completion, the TERMINALCOUNT interrupt event is also logged, and the GPMC_IRQSTATUS[1] TERMINALCOUNTSTATUS bit is set. To clear the interrupt, the MPU must clear the TERMINALCOUNTSTATUS bit. The TERMINALCOUNTSTATUS bit must always be cleared prior to asserting the TERMINALCOUNTEVENTENABLE bit to clear any out-of-date logged interrupt event.

**NOTE:** The COUNTVALUE value is only valid when the prefetch engine is active (started), and an interrupt is only triggered when COUNTVALUE reaches 0, that is, when the prefetch engine automatically goes from an active to an inactive state.

The number of bytes to be prefetched (programmed in TRANSFERCOUNT) must be a multiple of the programmed FIFOTHRESHOLD to trigger the correct number of interrupts allowing a deterministic and transparent FIFO control. If this guideline is respected, the number of ISR accesses is always required and the FIFO is always empty after the last interrupt is trigerred. In other cases, the TERMINALCOUNT interrupt must be used to read the remaining bytes in the FIFO (the number of remaining bytes being lower than the FIFOTHRESHOLD value).

In DMA draining mode, the GPMC_PREFETCH_CONFIG1[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes are ready to be read from the FIFO. The DMA channel owning this DMA request must be programmed so that the number of bytes programmed in FIFOTHRESHOLD is read from the FIFO during the DMA request process. The DMA request is kept active until this number of bytes has effectively been read from the FIFO, and no other DMA request can be issued until the ongoing active request is complete.

In prefetch mode, the TERMINALCOUNT event is also a source of DMA requests if the number of bytes to be prefetched is not a multiple of FIFOTHRESHOLD, the remaining bytes in the FIFO can be read by the DMA channel using the last DMA request. This assumes that the number of remaining bytes to be read is known and controlled through the DMA channel programming model.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive to active prefetch (the STARTENGINE bit is set to 1). The associated DMA channel must always be enabled by the MPU after setting the STARTENGINE bit so that the out-of-date active DMA request does not trigger spurious DMA transfers.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.4.4 Write-Posting Mode

The write-posting mode is selected when the GPMC_PREFETCH_CONFIG1[0] ACCESSMODE bit is set.

The MPU NAND software driver must issue the correct address pointer initialization command (page program) before the engine can start writing data into the NAND memory device. The engine starts when the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit is set to 1. The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the MPU NAND software driver must issue the second cycle program command and monitor the status for programming process completion (adding ECC handling, if required).

If used, the ECC calculator engine must be started (configured, reset, and enabled) before the posting engine is started so that the ECC parities are properly calculated on all data written by the prefetch engine to the associated chip-select.

In write-posting mode, the GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE bit must be cleared so that posting starts as soon as the STARTENGINE bit is set and the FIFO is not empty.

If the STARTENGINE bit is set after the NAND address phase (page program command), the STARTENGINE setting is effective only after the actual NAND command completion. To prevent GPMC stall during this NAND command phase, set the STARTENGINE bit field before the NAND address completion and ensure that the associated DMA channel is enabled after the NAND address phase.

The posting engine issues a write request when valid data are available from the FIFO and until the programmed GPMC_PREFETCH_CONFIG2[13-0] TRANSFERCOUNT accesses have been completed.

The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the MPU NAND software driver must issue the second cycle program command and monitor the status for programming process completion. The closing program command phase must only be issued when the full NAND page has been written into the NAND flash write buffer, including the spare area data and the ECC parities, if used.

**Write-Posting Mode Configuration**

| Bit Field | Register | Value | Comments |
|---|---|---|---|
| STARTENGINE | GPMC_PREFETCH_CONTROL | 0 | Write-posting engine can be configured only if STARTENGINE is cleared to 0. |
| ENGINECSSELECTOR | GPMC_PREFETCH_CONFIG1 | 0 to 3h | Selects the chip-select associated with a NAND device where the prefetch engine is active |
| ACCESSMODE | GPMC_PREFETCH_CONFIG1 | 1 | Selects write-posting mode |
| FIFOTHRESHOLD | GPMC_PREFETCH_CONFIG1 | | Selects the maximum number of bytes read or written by the host on DMA or interrupt request |
| TRANSFERCOUNT | GPMC_PREFETCH_CONFIG2 | | Selects the number of bytes to be read or written by the engine from/to the selected chip-select |
| SYNCHROMODE | GPMC_PREFETCH_CONFIG1 | 0 | Engine starts the access to chip-select as soon as STARTENGINE is set. |
| ENABLEOPTIMIZEDACCESS | GPMC_PREFETCH_CONFIG1 | 0/1 | See Section 11.2.4.12.4.6 |
| CYCLEOPTIMIZATION | GPMC_PREFETCH_CONFIG | | |
| ENABLEENGINE | GPMC_PREFETCH_CONFIG1 | 1 | Engine enabled |
| STARTENGINE | GPMC_PREFETCH_CONTROL | 1 | Starts the prefetch engine |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.4.5 FIFO Control in Write-Posting Mode

The FIFO can be filled directly by the MPU or by an sDMA channel.

In MPU filling mode, the FIFO status can be monitored through the FIFOPOINTER or through the GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bit. FIFOPOINTER indicates the current number of available free byte places in the FIFO, and the FIFOTHRESHOLDSTATUS bit, when set, indicates that at least FIFOTHRESHOLD free byte places are available in the FIFO.

An interrupt can be issued by the GPMC if the GPMC_IRQENABLE[0] FIFOEVENTENABLE bit is set. When the interrupt is fired, the GPMC_IRQSTATUS[0] FIFOEVENTSTATUS bit is set. To clear the interrupt, the MPU must write enough bytes to fill the FIFO, or enough bytes to get below the programmed threshold, and the FIFOEVENTSTATUS bit must be cleared to get further interrupt events. The FIFOEVENTSTATUS bit must always be cleared prior to asserting the FIFOEVENTENABLE bit to clear any out-of-date logged interrupt event. This interrupt must be enabled after enabling the STARTENGINE bit

The posting completion can be monitored through the GPMC_PREFETCH_STATUS[13-0] COUNTVALUE field. COUNTVALUE indicates the current number of remaining data to be written based on the TRANSFERCOUNT value. An interrupt is issued by the GPMC when the write-posting process completes (that is, COUNTVALUE equal to 0) if the GPMC_IRQENABLE[1] TERMINALCOUNTEVENTENABLE bit is set. When the interrupt is fired, the GPMC_IRQSTATUS[1] TERMINALCOUNTSTATUS bit is set. To clear the interrupt, the MPU must clear the TERMINALCOUNTSTATUS bit. The TERMINALCOUNTSTATUS bit must always be cleared prior to asserting the TERMINALCOUNTEVENTENABLE bit to clear any out-of-date logged interrupt event.

**NOTE:** The COUNTVALUE value is only valid if the write-posting engine is active and started, and an interrupt is only issued when COUNTVALUE reaches 0, that is, when the posting engine automatically goes from active to inactive.

In DMA filling mode, the DMAMode bit field in the GPMC_PREFETCH_CONFIG1[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes-free places are available in the FIFO. The DMA channel owning this DMA request must be programmed so that a number of bytes equal to the value programmed in the FIFOTHRESHOLD bit field are written into the FIFO during the DMA access. The DMA request remains active until the associated number of bytes has effectively been written into the FIFO, and no other DMA request can be issued until the ongoing active request has been completed.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive to active prefetch (STARTENGINE set to 1). The associated DMA channel must always be enabled by the MPU after setting the STARTENGINE bit so that an out-of-date active DMA request does not trigger spurious DMA transfers.

In write-posting mode, the DMA or the MPU fill the FIFO with no consideration to the associated byte enables. Any byte stored in the FIFO is written into the memory device.


Copyright © 2011–2016, Texas Instruments Incorporated

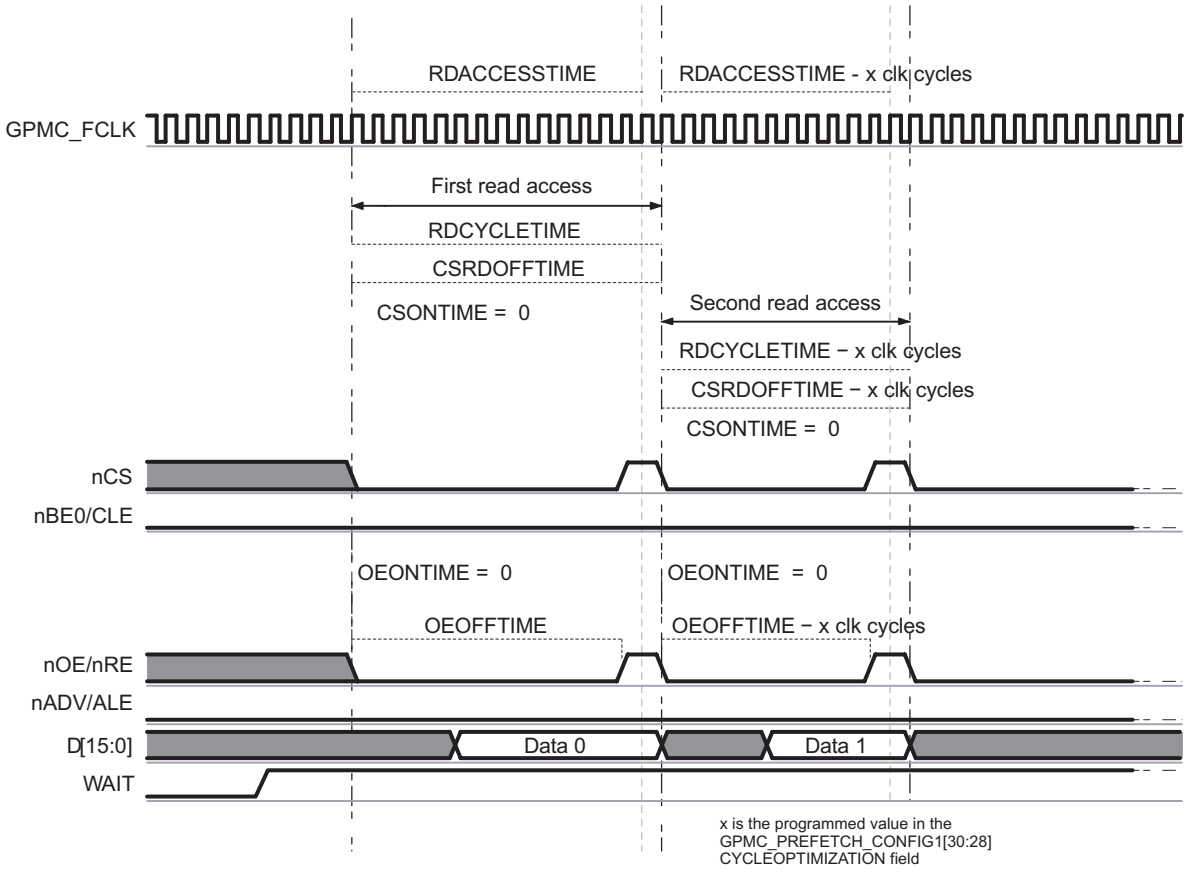
#### 11.2.4.12.4.6 Optimizing NAND Access Using the Prefetch and Write-Posting Engine

Access time to a NAND memory device can be optimized for back-to-back accesses if the associated $\overline{CS}$ signal is not deasserted between accesses. The GPMC access engine can track prefetch engine accesses to optimize the access timing parameter programmed for the allocated chip-select, if no accesses to other chip-selects (that is, interleaved accesses) occur. Similarly, the access engine also eliminates the CYCLE2CYCLEDELAY even if CYCLE2CYCLESAMECSEN is set. This capability is limited to the prefetch and write-posting engine accesses, and MPU accesses to a NAND memory device (through the defined chip-select memory region or through the GPMC_NAND_DATA_i(where , i = 0 to 5) are never optimized.

The GPMC_PREFETCH_CONFIG1[27] ENABLEOPTIMIZEDACCESS bit must be set to enable optimized accesses. To optimize access time, the GPMC_PREFETCH_CONFIG1[30-28] CYCLEOPTIMIZATION field defines the number of GPMC_FCLK cycles to be suppressed from the RDCYCLETIME, WRCYCLETIME, RDACCESSTIME, WRACCESSTIME, CSOFFTIME, ADVOFFTIME, OEOFFTIME, and WEOFFTIME timing parameters.

Figure 11-41, in the case of back-to-back accesses to the NAND flash through the prefetch engine, CYCLE2CYCLESAMECSEN is forced to 0 when using optimized accesses. The first access uses the regular timing settings for this chip-select. All accesses after this one use settings reduced by x clock cycles, x being defined by the GPMC_PREFETCH_CONFIG1[30-28] CYCLEOPTIMIZATION field.

**Figure 11-41. NAND Read Cycle Optimization Timing Description**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.2.4.12.4.7 Interleaved Accesses Between Prefetch and Write-Posting Engine and Other Chip-Selects

Any on-going read or write access from the prefetch and write-posting engine is completed before an access to any other chip-select can be initiated. As a default, the arbiter uses a fixed-priority algorithm, and the prefetch and write-posting engine has the lowest priority. The maximum latency added to access starting time in this case equals the RDCYCLETIME or WRCYCLETIME (optimized or not) plus the requested BUSTURNAROUND delay for bus turnaround completion programmed for the chip-select to which the NAND device is connected to.

Alternatively, a round-robin arbitration can be used to prioritize accesses to the external bus. This arbitration scheme is enabled by setting the GPMC_PREFETCH_CONFIG1[23] PFPWENROUNDROBIN bit. When a request to another chip-select is received while the prefetch and write-posting engine is active, priority is given to the new request. The request processed thereafter is the prefetch and write-posting engine request, even if another interconnect request is passed in the mean time. The engine keeps control of the bus for an additional number of requests programmed in the GPMC_PREFETCH_CONFIG1[19-16] PFPWWEIGHTEDPRIO bit field. Control is then passed to the direct interconnect request.

As an example, the round-robin arbitration scheme is selected with PFPWWEIGHTEDPRIO set to 2h. Considering the prefetch and write-posting engine and the interconnect interface are always requesting access to the external interface, the GPMC grants priority to the direct interconnect access for one request. The GPMC then grants priority to the engine for three requests, and finnaly back to the direct interconnect access, until the arbiter is reset when one of the two initiators stops initiating requests.

## 11.3 Basic Programming Model

### 11.3.1 GPMC High-Level Programming Model Overview

The high-level programming model introduces a top-down approach for users that need to configure the GPMC module. Figure 11-42 shows a programming model top-level diagram for the GPMC. Each block of the diagram is described in one of the following subsections through a set of registers to configure. Table 11-23 and Table 11-24 list each step in the model.

**Table 11-23. GPMC Configuration in NOR Mode**

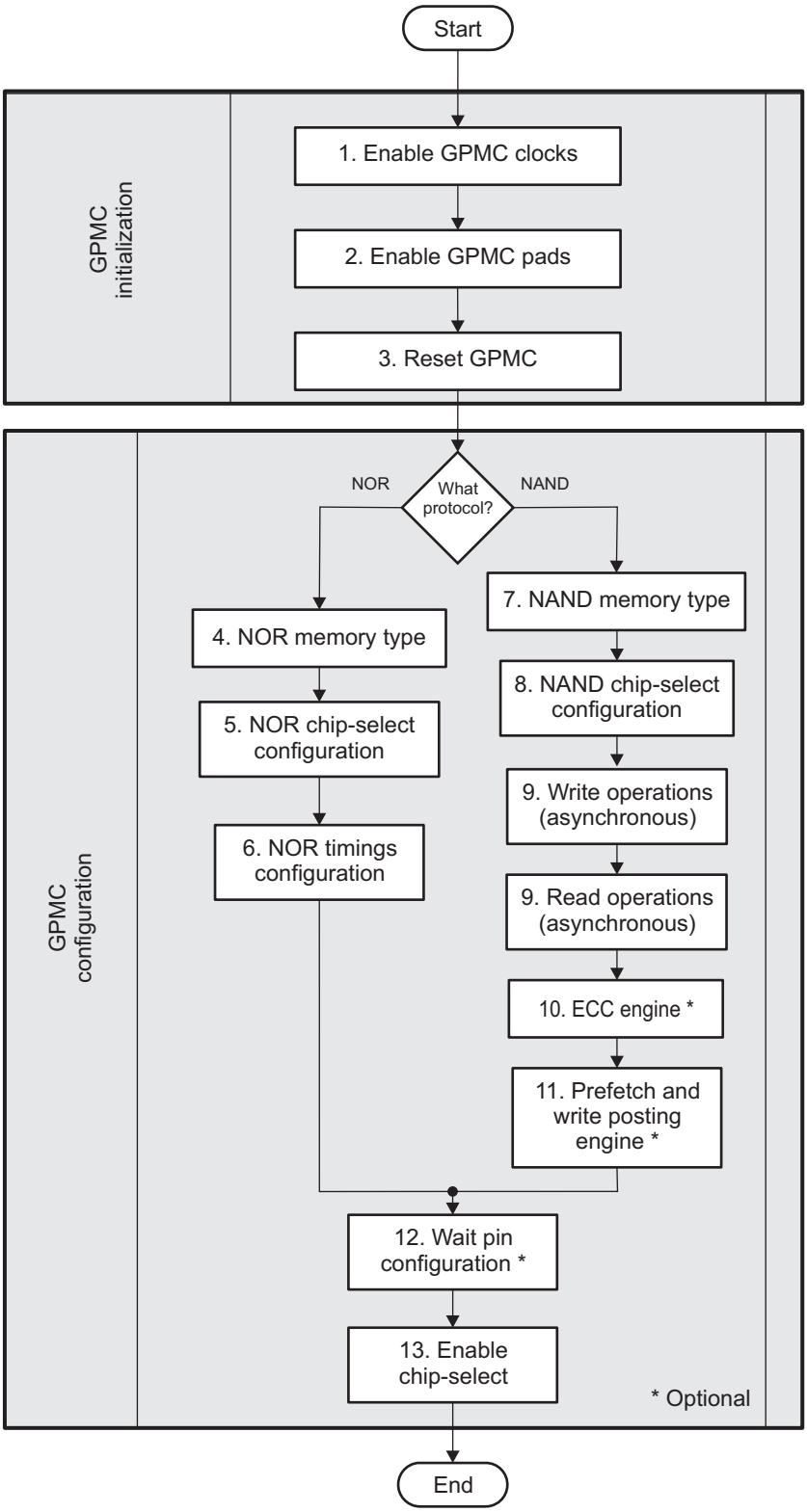
| Step | Description |
|---|---|
| NOR Memory Type | See Table 11-26 |
| NOR Chip-Select Configuration | See Table 11-27 |
| NOR Timings Configuration | See Table 11-28 |
| Wait Pin Configuration | See Table 11-29 |
| Enable Chip-Select | See Table 11-30 |

**Table 11-24. GPMC Configuration in NAND Mode**

| Step | Description |
|---|---|
| NAND Memory Type | See Table 11-31 |
| NAND Chip-Select Configuration | See Table 11-32 |
| Write Operations (Asynchronous) | See Table 11-33 |
| Read Operations (Asynchronous) | See Table 11-33 |
| ECC Engine | See Table 11-34 |
| Prefetch and Write-Posting Engine | See Table 11-35 |
| Wait Pin Configuration | See Table 11-36 |
| Enable Chip-Select | See Table 11-37 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 11-42. Programming Model Top-Level Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

### 11.3.2 GPMC Initialization

Table 11-25 describes the settings required to reset the GPMC.

**Table 11-25. Reset GPMC**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Start a software reset | GPMC_SYSCONFIG[1] SOFTRESET | 1 |
| Wait until | GPMC_SYSSTATUS[0] RESETDONE | 1 |

### 11.3.3 GPMC Configuration in NOR Mode

This section gives a generic configuration for parameters related to the NOR memory connected to the GPMC. Table 11-26 through Table 11-30 list the steps to configure the GPMC in NOR mode.

**NOTE:** In the tables of this section, 'x' in Value column stands for 'depends on configuration'.

**Table 11-26. NOR Memory Type**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Set the NOR protocol | GPMC_CONFIG1_i[11-10] DEVICETYPE | 0 |
| Set a device size | GPMC_CONFIG1_i[13-12] DEVICESIZE | x |
| Select an address and data multiplexing protocol | GPMC_CONFIG1_i[9] MUXADDDATA | x |
| Set the attached device page length | GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH | x |
| Set the wrapping burst capabilities | GPMC_CONFIG1_i[31] WRAPBURST | x |
| Select a timing signals latencies factor | GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY | x |
| Select an output clock frequency | GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER | x |
| Choose an output clock activation time | GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME | x |
| Set a single or multiple access for read operations | GPMC_CONFIG1_i[30] READMULTIPLE | x |
| Set a synchronous or asynchronous mode for read operations | GPMC_CONFIG1_i[29] READTYPE | x |
| Set a single or multiple access for write operations | GPMC_CONFIG1_i[28] WRITEMULTIPLE | x |
| Set a synchronous or asynchronous mode for write operations | GPMC_CONFIG1_i[27] WRITETYPE | x |

**Table 11-27. NOR Chip-Select Configuration**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Select the chip-select base address | GPMC_CONFIG7_i[5-0] BASEADDRESS | x |
| Select the chip-select mask address | GPMC_CONFIG7_i[11-8] MASKADDRESS | x |

**Table 11-28. NOR Timings Configuration**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Configure adequate timing parameters in various memory modes | See Section 11.3.6 | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-29. WAIT Pin Configuration**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Enable or disable wait pin monitoring for read operations | GPMC_CONFIG1_i[22] WAITREADMONITORING | x |
| Enable or disable wait pin monitoring for write operations | GPMC_CONFIG1_i[21] WAITWRITEMONITORING | x |
| Select a wait pin monitoring time | GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME | x |
| Choose the input wait pin for the chip-select | GPMC_CONFIG1_i[17-16] WAITPINSELECT | x |

**Table 11-30. Enable Chip-Select**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| When all parameters are configured, enable the chip-select | GPMC_CONFIG7_i[6] CSVALID | x |

### *11.3.4 GPMC Configuration in NAND Mode*

This section gives a generic configuration for parameters related to NAND memory connected to the GPMC.

**Table 11-31. NAND Memory Type**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Set the NAND protocol | GPMC_CONFIG1_i[11-10] DEVICETYPE | 2h |
| Set a device size | GPMC_CONFIG1_i[13-12] DEVICESIZE | x |
| Set the address and data multiplexing protocol to non-multiplexed attached device | GPMC_CONFIG1_i[9] MUXADDDATA | 0 |
| Select a timing signals latencies factor | GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY | x |
| Set a synchronous or asynchronous mode and a single or multiple access for read and write operations | See Section 11.3.5 | x |

**Table 11-32. NAND Chip-Select Configuration**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Select the chip-select base address | GPMC_CONFIG7_i[5-0] BASEADDRESS | x |
| Select the chip-select minimum granularity (16M bytes) | GPMC_CONFIG7_i[11-8] MASKADDRESS | x |

**Table 11-33. Asynchronous Read and Write Operations**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Configure adequate timing parameters in asynchronous modes | See Section 11.3.6 | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-34. ECC Engine**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Select the ECC result register where the first ECC computation is stored (Only applies to Hamming) | GPMC_ECC_CONTROL[3-0] ECCPOINTER | x |
| Clear all ECC result registers | GPMC_ECC_CONTROL[8] ECCCLEAR | Write 1 to clear |
| Define ECCSIZE0 and ECCSIZE1 | GPMC_ECC_SIZE_CONFIG[19-12] ECCSIZE0 and GPMC_ECC_SIZE_CONFIG[29-22] ECCSIZE1 | x |
| Select the size of each of the 9 result registers (size specified by ECCSIZE0 or ECCSIZE1) | GPMC_ECC_SIZE_CONFIG[j-1] ECCjRESULTSIZE where j = 1 to 9 | x |
| Select the chip-select where ECC is computed | GPMC_ECC_SIZE_CONFIG[3-1] ECCCS | x |
| Select the Hamming code or BCH code ECC algorithm in use | GPMC_ECC_SIZE_CONFIG[16] ECCALGORITHM | x |
| Select word size for ECC calculation | GPMC_ECC_SIZE_CONFIG[7] ECC16B | x |
| If the BCH code is used,<br>Set an error correction capability and<br>Select a number of sectors to process | GPMC_ECC_SIZE_CONFIG[13-12] ECCBCHTSEL and GPMC_ECC_SIZE_CONFIG[6-4] ECCTOPSECTOR | x |
| Enable the ECC computation | GPMC_ECC_SIZE_CONFIG[0] ECCENABLE | 1 |

**Table 11-35. Prefetch and Write-Posting Engine**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Disable the engine before configuration | GPMC_PREFETCH_CONTROL[0] STARTENGINE | 0 |
| Select the chip-select associated with a NAND device where the prefetch engine is active | GPMC_PREFETCH_CONFIG1[26-24] ENGINECSSELECTOR | x |
| Select access direction through prefetch engine, read or write. | GPMC_PREFETCH_CONFIG1[0] ACCESSMODE | x |
| Select the threshold used to issue a DMA request | GPMC_PREFETCH_CONFIG1[14-8] FIFOTHRESHOLD | x |
| Select either DMA synchronized mode or SW manual mode. | GPMC_PREFETCH_CONFIG1[2] DMAMODE | x |
| Select if the engine immediately starts accessing the memory upon STARTENGINE assertion or if hardware synchronization based on a WAIT signal is used. | GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE | x |
| Select which wait pin edge detector should start the engine in synchronized mode | GPMC_PREFETCH_CONFIG1[5-4] WAITPINSELECTOR | x |
| Enter a number of clock cycles removed to timing parameters (For all back-to-back accesses to the NAND flash but not the first one) | GPMC_PREFETCH_CONFIG1[30-28] CYCLEOPTIMIZATION | x |
| Enable the prefetch postwite engine | GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE | 1 |
| Select the number of bytes to be read or written by the engine to the selected chip-select | GPMC_PREFETCH_CONFIG2[13-0] TRANSFERCOUNT | x |
| Start the prefetch engine | GPMC_PREFETCH_CONTROL[0] STARTENGINE | 1 |

**Table 11-36. WAIT Pin Configuration**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| Selects when the engine starts the access to CS | GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE | x |
| Select which wait pin edge detector should start the engine in synchronized mode | GPMC_PREFETCH_CONFIG1[5-4] WAITPINSELECTOR | x |

**Table 11-37. Enable Chip-Select**

| Sub-process Name | Register / Bitfield | Value |
|---|---|---|
| When all parameters are configured, enable the chip-select | GPMC_CONFIG7_i[6] CSVALID | x |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.3.5 Set Memory Access

This section details the bit field to configure to set the GPMC in various memory modes.

**Table 11-38. Mode Parameters Check List Table**

| Register | Bit | Bit Field Name | Asynchronous | | | | Synchronous | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Single Read Access | Single Write Access | Multiple Read (Page) Access | Multiple Write (Page) Access | Single Read Access | Single Write Access | Multiple Read (Burst) Access | Multiple Write (Burst) Access |
| GPMC_CONFIG1_i | 30 | READMULTIPLE | 0 | - | 1 | N/S | 0 | - | 1 | - |
| GPMC_CONFIG1_i | 29 | READTYPE | 0 | - | 0 | N/S | 1 | - | 1 | - |
| GPMC_CONFIG1_i | 28 | WRITEMULTIPLE | - | 0 | | N/S | - | 0 | - | 1 |
| GPMC_CONFIG1_i | 27 | WRITETYPE | - | 0 | | N/S | - | 1 | - | 1 |

**Table 11-39. Access Type Parameters Check List Table**

| Register | Bit | Bit Field Name | Access Type | | |
|---|---|---|---|---|---|
| | | | Non-Mux | Address/Data Mux | AAD Mux |
| GPMC_CONFIG1_i | 9-8 | MUXADDDATA | 0 | 2h | 1 |

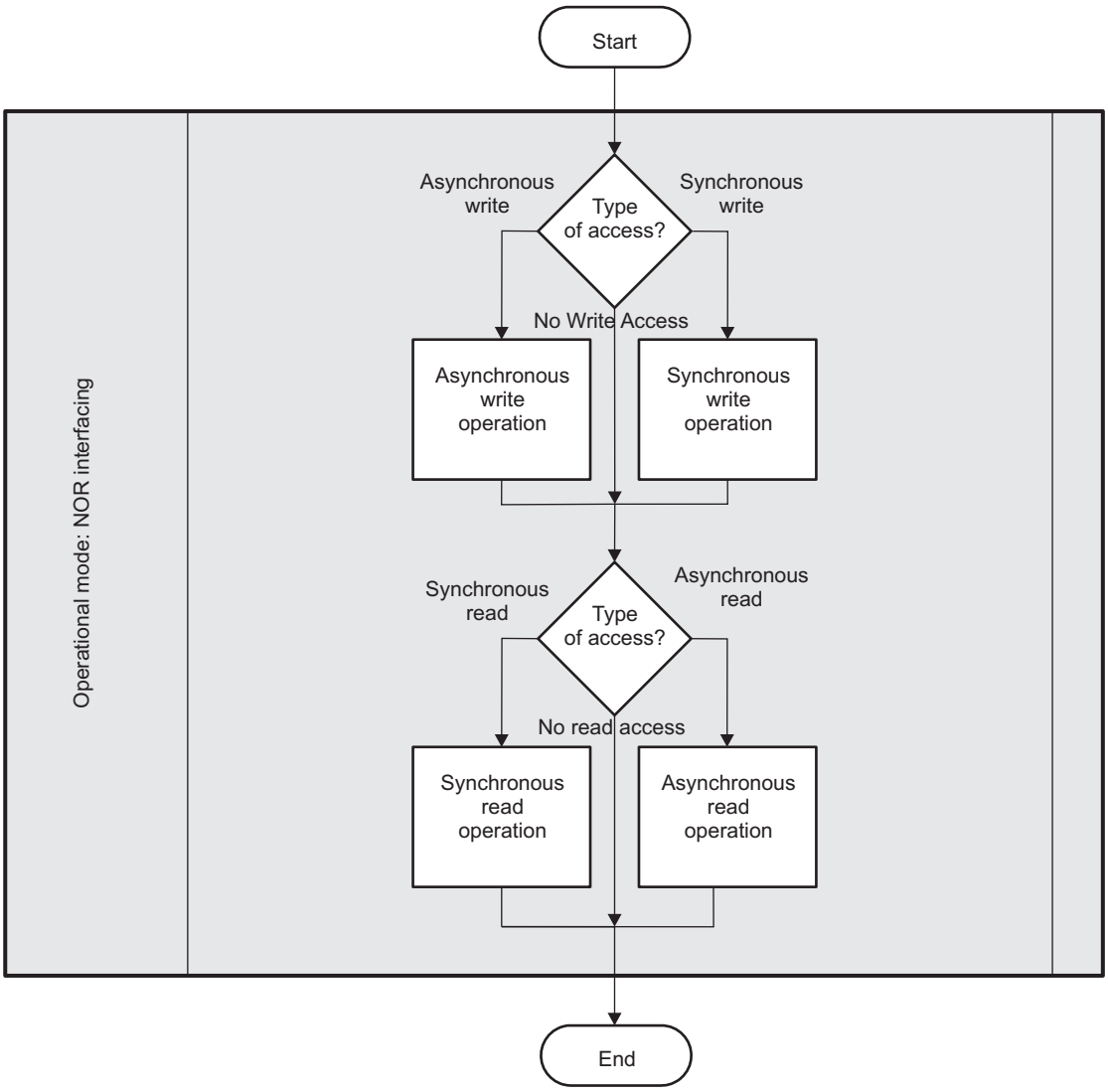

Copyright © 2011–2016, Texas Instruments Incorporated

### 11.3.6 GPMC Timing Parameters

Figure 11-43 shows a programming model diagram for the NOR interfacing timing parameters.

Table 11-40 lists bit fields to configure adequate timing parameter in various memory modes.

**Figure 11-43. NOR Interfacing Timing Parameters Diagram**






Copyright © 2011–2016, Texas Instruments Incorporated

## Table 11-40. Timing Parameters

| Register | Bit | Bit Field Name | Asynchronous | | | Synchronous | | | | Access Type | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Single Read Access | Single Write Access | Multiple Read (Page) access | Single Read Access | Single Write Access | Multiple Read (Burst) Access | Multiple Write (Burst) Access | Non-multiplexed | Address /Data-multiplexed | AAD-multiplexed |
| GPMC_CONFIG1_i | 9-8 | MUXADDDATA | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG1_i | 29 | READTYPE | y | | y | y | | y | | y | y | y |
| GPMC_CONFIG1_i | 30 | READMULTIPLE | y | | y | y | | y | | y | y | y |
| GPMC_CONFIG1_i | 27 | WRITETYPE | | y | | | y | | y | y | y | y |
| GPMC_CONFIG1_i | 28 | WRITEMULTIPLE | | y | | | y | | y | y | y | y |
| GPMC_CONFIG1_i | 31 | WRAPBURST | | | | | | y | y | y | y | y |
| GPMC_CONFIG1_i | 26-25 | CLKACTIVATIONTIME | | | | y | y | y | y | y | y | y |
| GPMC_CONFIG1_i | 19-18 | WAITMONITORINGTIME | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG1_i | 4 | TIMEPARAGRANULARITY | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG2_i | 20-16 | CSWROFFTIME | | y | | | y | | y | y | y | y |
| GPMC_CONFIG2_i | 12-8 | CSRDOFFTIME | y | | y | y | | y | | y | y | y |
| GPMC_CONFIG2_i | 7 | CSEXTRADELAY | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG2_i | 3-0 | CSONTIME | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG3_i | 30-28 | ADVAADMUXWROFFTIME | | y | | | y | | y | | | y |
| GPMC_CONFIG3_i | 26-24 | ADVAADMUXRDOFFTIME | y | | y | y | | y | | | | y |
| GPMC_CONFIG3_i | 6-4 | ADVAADMUXONTIME | y | y | y | y | y | y | y | | | y |
| GPMC_CONFIG3_i | 20-16 | ADVWROFFTIME | | y | | | y | | y | y | y | y |
| GPMC_CONFIG3_i | 12-8 | ADVRDOFFTIME | y | | y | y | | y | | y | y | y |
| GPMC_CONFIG3_i | 7 | ADVEXTRADELAY | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG3_i | 3-0 | ADVONTIME | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG4_i | 15-13 | OEAADMUXOFFTIME | y | y | y | y | y | y | y | | | y |
| GPMC_CONFIG4_i | 6-4 | OEAADMUXONTIME | y | y | y | y | y | y | y | | | y |
| GPMC_CONFIG4_i | 28-24 | WEOFFTIME | | y | | | y | | y | y | y | y |
| GPMC_CONFIG4_i | 23 | WEEXTRADELAY | | y | | | y | | y | y | y | y |
| GPMC_CONFIG4_i | 19-16 | WEONTIME | | y | | | y | | y | y | y | y |
| GPMC_CONFIG4_i | 12-8 | OEOFFTIME | y | | y | y | | y | | y | y | y |
| GPMC_CONFIG4_i | 7 | OEEXTRADELAY | y | | y | y | | y | | y | y | y |
| GPMC_CONFIG4_i | 3-0 | OEONTIME | y | | y | y | | y | | y | y | y |
| GPMC_CONFIG5_i | 27-24 | PAGEBURSTACCESSTIME | | | y | | | y | y | y | y | y |
| GPMC_CONFIG5_i | 20-16 | RDACCESSTIME | y | | y | y | | y | | y | y | y |
| GPMC_CONFIG5_i | 12-8 | WRCYCLETIME | | y | | | y | | y | y | y | y |
| GPMC_CONFIG5_i | 4-0 | RDCYCLETIME | y | | y | y | | y | | y | y | y |
| GPMC_CONFIG6_i | 28-24 | WRACCESSTIME | | y | | | y | | y | y | y | y |
| GPMC_CONFIG6_i | 19-16 | WRDATAONADMUXBUS | | y | | | y | | y | | y | y |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-40. Timing Parameters (continued)**

| Register | Bit | Bit Field Name | Asynchronous | | | Synchronous | | | | Access Type | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Single Read Access | Single Write Access | Multiple Read (Page) access | Single Read Access | Single Write Access | Multiple Read (Burst) Access | Multiple Write (Burst) Access | Non-multiplexed | Address /Data-multiplexed | AAD-multiplexed |
| GPMC_CONFIG6_i | 11-8 | CYCLE2CYCLEDELAY | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG6_i | 7 | CYCLE2CYCLESAMECSEN | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG6_i | 6 | CYCLE2CYCLEDIFFCSEN | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG6_i | 3-0 | BUSTURNAROUND | y | y | y | y | y | y | y | y | y | y |
| GPMC_CONFIG7_i | 6 | CSVALID | y | y | y | y | y | y | y | y | y | y |



Copyright © 2011–2016, Texas Instruments Incorporated

### 11.3.6.1 GPMC Timing Parameters Formulas

This section intends to help the user to calculate the GPMC timing bit fields values. Formulas are not listed exhaustively.

The section details:

- NAND Flash Interface Timing Parameters Formulas
- Synchronous NOR Flash Timing Parameters Formulas
- Asynchronous NOR Flash Timing Parameters Formulas

#### *11.3.6.1.1 NAND Flash Interface Timing Parameters Formulas*

This section lists formulas to use in order to calculate NAND timing parameters. This is the case when GPMC_CONFIG1_i[11-10] DEVICETYPE = 2h. Table 11-41 details NAND timing parameters.

**Table 11-41. NAND Formulas Description Table**

| Configuration Parameter | Unit | Description |
|---|---|---|
| A | ns | Pulse duration - $\overline{\text{GPMC_WE}}$ valid time |
| B | ns | Delay time - $\overline{\text{GPMC_CS}}$ valid to $\overline{\text{GPMC_WE}}$ valid |
| C | ns | Delay time - GPMC_$\overline{\text{BE0}}$_CLE/GPMC_$\overline{\text{ADV}}$_ALE high to $\overline{\text{GPMC_WE}}$ valid |
| D | ns | Delay time - GPMC_AD[15:0] valid to $\overline{\text{GPMC_WE}}$ valid |
| E | ns | Delay time - $\overline{\text{GPMC_WE}}$ invalid to GPMC_AD[15:0] invalid |
| F | ns | Delay time - $\overline{\text{GPMC_WE}}$ invalid to GPMC_$\overline{\text{BE0}}$_CLE/GPMC_$\overline{\text{ADV}}$_ALE invalid |
| G | ns | Delay time - $\overline{\text{GPMC_WE}}$ invalid to $\overline{\text{GPMC_CS}}$ invalid |
| H | ns | Cycle time - Write cycle time |
| I | ns | Delay time - $\overline{\text{GPMC_CS}}$ valid to $\overline{\text{GPMC_OE}}$ valid |
| J | ns | Setup time - GPMC_AD[15:0] valid to $\overline{\text{GPMC_OE}}$ invalid |
| K | ns | Pulse duration - $\overline{\text{GPMC_OE}}$ valid time |
| L | ns | Cycle time - Read cycle time |
| M | ns | Delay time - $\overline{\text{GPMC_OE}}$ invalid to $\overline{\text{GPMC_CS}}$ invalid |

The configuration parameters are calculated through the following formulas.

A = (WEOffTime - WEOnTime) × (TimeParaGranularity + 1) × GPMC_FCLK period

B = ((WEOnTime - CSOnTime) × (TimeParaGranularity + 1) + 0.5 × (WEExtraDelay - CSExtraDelay)) × GPMC_FCLK period

C = ((WEOnTime - ADVOnTime) × (TimeParaGranularity + 1) + 0.5 × (WEExtraDelay - ADVExtraDelay)) × GPMC_FCLK period

D = (WEOnTime × (TimeParaGranularity + 1) + 0.5 × WEExtraDelay ) × GPMC_FCLK period

E = (WrCycleTime - WEOffTime × (TimeParaGranularity + 1) - 0.5 × WEExtraDelay ) × GPMC_FCLK period

F = (ADVWrOffTime - WEOffTime × (TimeParaGranularity + 1) + 0.5 × (ADVExtraDelay - WEExtraDelay ) × GPMC_FCLK period

G = (CSWrOffTime - WEOffTime × (TimeParaGranularity + 1) + 0.5 × (CSExtraDelay - WEExtraDelay ) × GPMC_FCLK period

H = WrCycleTime × (1 + TimeParaGranularity) × GPMC_FCLK period

I = ((OEOnTime - CSOnTime) × (TimeParaGranularity + 1) + 0.5 × (OEExtraDelay - CSExtraDelay)) × GPMC_FCLK period

J = ((RdAccessTime - OEOffTime) × (TimeParaGranularity + 1) - 0.5 × OEExtraDelay)) × GPMC_FCLK period


Copyright © 2011–2016, Texas Instruments Incorporated

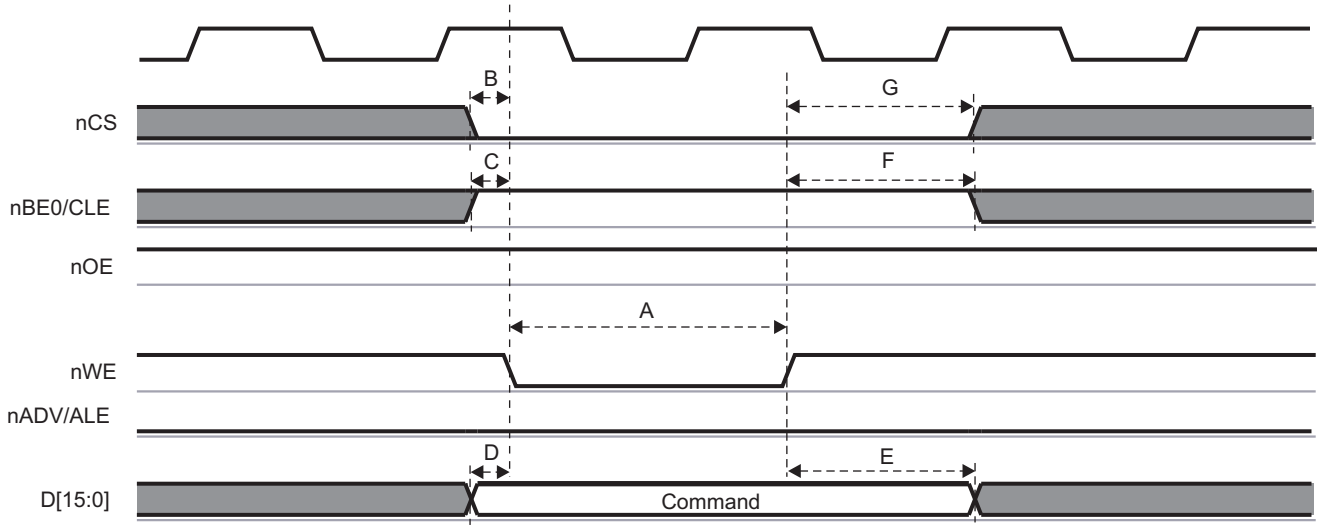
K = (OEOffTime - OEOnTime) × (1 + TimeParaGranularity) × GPMC_FCLK period

L = RdCycleTime × (1 + TimeParaGranularity) × GPMC_FCLK period

M = (CSRdOffTime - OEOffTime × (TimeParaGranularity + 1) + 0.5 × (CSExtraDelay - OEExtraDelay ) × GPMC_FCLK period

Figure 11-44 shows a command latch cycle timing simplified example where formulas are associated with signal waves.

**Figure 11-44. NAND Command Latch Cycle Timing Simplified Example**




### 11.3.6.1.2 Synchronous NOR Flash Timing Parameters Formulas

This section lists all formulas to use in order to calculate synchronous NOR timing parameters. This is the case when GPMC_CONFIG1_i[11-10] DEVICETYPE = 0 and when READTYPE or WRITETYPE are set to synchronous mode.

**Table 11-42. Synchronous NOR Formulas Description Table**

| Configuration Parameter | Unit | Description |
|---|---|---|
| A | ns | Pulse duration - $\overline{\text{GPMC_CS}}$ low |
| B | ns | Delay time - address bus valid to GPMC_CLK first edge |
| | | Delay time - GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ valid to GPMC_CLK first edge |
| C | ns | Pulse duration - GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ low |
| D | ns | Delay time - GPMC_CLK rising edge to GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ invalid |
| | | Delay time - GPMC_CLK rising edge to GPMC_$\overline{\text{ADV}}$_ALE invalid |
| E | ns | Delay time - GPMC_CLK rising edge to $\overline{\text{GPMC_CS}}$ invalid |
| | | Delay time - GPMC_CLK rising edge to $\overline{\text{GPMC_OE}}$ invalid |
| F | ns | Delay time - GPMC_CLK rising edge to $\overline{\text{GPMC_CS}}$ transition |
| G | ns | Delay time - GPMC_CLK rising edge to GPMC_$\overline{\text{ADV}}$_ALE transition |
| H | ns | Delay time - GPMC_CLK rising edge to $\overline{\text{GPMC_OE}}$ transition |
| I | ns | Delay time - GPMC_CLK rising edge to $\overline{\text{GPMC_WE}}$ transition |
| J | ns | Delay time - GPMC_CLK rising edge to GPMC_AD data bus transition |
| | | Delay time - GPMC_CLK rising edge to GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ transition |
| K | ns | Pulse duration - GPMC_$\overline{\text{ADV}}$_ALE low |
| L | ns | Delay time - GPMC_WAIT invalid to first data latching GPMC_CLK edge |


Copyright © 2011–2016, Texas Instruments Incorporated

The configuration parameters are calculated through the following formulas.

For single read accesses:

A = (CSRDOFFTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

C = RDCYCLETIME × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

D = (RDCYCLETIME - RDACCESSTIME) × GPMC_FCLK period

E = (CSRDOFFTIME - RDACCESSTIME) × GPMC_FCLK period

For burst read accesses (where n is the page burst access number):

A = (CSRDOFFTIME - CSONTIME + (n - 1) × PAGEBURSTACCESSTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

C = (RDCYCLETIME + (n - 1) × PAGEBURSTACCESSTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

D = (RDCYCLETIME - (RDACCESSTIME + (n - 1) × PAGEBURSTACCESSTIME)) × GPMC_FCLK period

E = (CSRDOFFTIME - (RDACCESSTIME + (n - 1) × PAGEBURSTACCESSTIME)) × GPMC_FCLK period

For burst write accesses (where n is the page burst access number):

A = (CSWROFFTIME - CSONTIME + (n - 1) × PAGEBURSTACCESSTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

C = (WRCYCLETIME + (n - 1) × PAGEBURSTACCESSTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

D = (WRCYCLETIME - (WRACCESSTIME + (n - 1) × PAGEBURSTACCESSTIME)) × GPMC_FCLK period

E = (CSWROFFTIME - (WRACCESSTIME + (n - 1) × PAGEBURSTACCESSTIME)) × GPMC_FCLK period

For all accesses:

For $\overline{CS}$ falling edge ($\overline{CS}$ activated):

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0

  F = 0.5 × CSEXTRADELAY × GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 1

  F = 0.5 × CSEXTRADELAY × GPMC_FCLK period, when (CLKACTIVATIONTIME and CSONTIME are odd) or (CLKACTIVATIONTIME and CSONTIME are even)

  F = (1 + 0.5 × CSEXTRADELAY) × GPMC_FCLK period, otherwise
- Case where GPMCFCLKDIVIDER = 2

  F = 0.5 × CSEXTRADELAY × GPMC_FCLK period, when (CSONTIME - CLKACTIVATIONTIME) is a multiple of 3

  F = (1 + 0.5 × CSEXTRADELAY) × GPMC_FCLK period, when (CSONTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  F = (2 + 0.5 × CSEXTRADELAY) × GPMC_FCLK period, when (CSONTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{CS}$ rising edge ($\overline{CS}$ de-activated) in reading mode:

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0

  F = 0.5 × CSEXTRADELAY × GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 1

  F = 0.5 × CSEXTRADELAY × GPMC_FCLK period, when (CLKACTIVATIONTIME and


Copyright © 2011–2016, Texas Instruments Incorporated

CSRDOFFTIME are odd) or (CLKACTIVATIONTIME and CSRDOFFTIME are even)
F = (1 + 0.5 × CSEXTRADELAY) × GPMC_FCLK period, otherwise


Copyright © 2011–2016, Texas Instruments Incorporated

- Case where GPMCFCLKDIVIDER = 2

  F = 0.5 × CSEXTRADELAY × GPMC_FCLK period, when (CSRDOFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  F = (1 + 0.5 × CSEXTRADELAY) × GPMC_FCLK period, when (CSRDOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  F = (2 + 0.5 × CSEXTRADELAY) × GPMC_FCLK period, when (CSRDOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{CS}$ rising edge ($\overline{CS}$ de-activated) in writing mode:

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0

  F = 0.5 × CSEXTRADELAY × GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 1

  F = 0.5 × CSEXTRADELAY × GPMC_FCLK period, when (CLKACTIVATIONTIME and CSWROFFTIME are odd) or (CLKACTIVATIONTIME and CSWROFFTIME are even)

  F = (1 + 0.5 × CSEXTRADELAY) × GPMC_FCLK period, otherwise
- Case where GPMCFCLKDIVIDER = 2

  F = 0.5 × CSEXTRADELAY × GPMC_FCLK period, when (CSWROFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  F = (1 + 0.5 × CSEXTRADELAY) × GPMC_FCLK period, when (CSWROFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  F = (2 + 0.5 × CSEXTRADELAY) × GPMC_FCLK period, when (CSWROFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{ADV}$ falling edge ($\overline{ADV}$ activated):

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0

  G = 0.5 × ADVEXTRADELAY × GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 1

  G = 0.5 × ADVEXTRADELAY × GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVONTIME are odd) or (CLKACTIVATIONTIME and ADVONTIME are even)

  G = (1 + 0.5 × ADVEXTRADELAY) × GPMC_FCLK period, otherwise
- Case where GPMCFCLKDIVIDER = 2

  G = 0.5 × ADVEXTRADELAY × GPMC_FCLK period, when (ADVONTIME - CLKACTIVATIONTIME) is a multiple of 3

  G = (1 + 0.5 × ADVEXTRADELAY) × GPMC_FCLK period, when (ADVONTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  G = (2 + 0.5 × ADVEXTRADELAY) × GPMC_FCLK period, when (ADVONTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{ADV}$ rising edge ($\overline{ADV}$ de-activated) in reading mode:

- Case where [1-0] GPMCFCLKDIVIDER = 0

  G = 0.5 × ADVEXTRADELAY × GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 1

  G = 0.5 × ADVEXTRADELAY × GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVRDOFFTIME are odd) or (CLKACTIVATIONTIME and ADVRDOFFTIME are even)

  G = (1 + 0.5 × ADVEXTRADELAY) × GPMC_FCLK period, otherwise


Copyright © 2011–2016, Texas Instruments Incorporated

- Case where GPMCFCLKDIVIDER = 2

  G = 0.5 × ADVEXTRADELAY × GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  G = (1 + 0.5 × ADVEXTRADELAY) × GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  G = (2 + 0.5 × ADVEXTRADELAY) × GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{\text{ADV}}$ rising edge ($\overline{\text{ADV}}$ de-activated) in writing mode:

- Case where [1-0] GPMCFCLKDIVIDER = 0

  G = 0.5 × ADVEXTRADELAY × GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 1

  G = 0.5 × ADVEXTRADELAY × GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVWROFFTIME are odd) or (CLKACTIVATIONTIME and ADVWROFFTIME are even)

  G = (1 + 0.5 × ADVEXTRADELAY) × GPMC_FCLK period, otherwise
- Case where GPMCFCLKDIVIDER = 2

  G = 0.5 × ADVEXTRADELAY × GPMC_FCLK period, when (ADVWROFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  G = (1 + 0.5 × ADVEXTRADELAY) × GPMC_FCLK period, when (ADVWROFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  G = (2 + 0.5 × ADVEXTRADELAY) × GPMC_FCLK period, when (ADVWROFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{\text{OE}}$ falling edge ($\overline{\text{OE}}$ activated):

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0

  H = 0.5 × OEEXTRADELAY × GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 1

  H = 0.5 × OEEXTRADELAY × GPMC_FCLK period, when (CLKACTIVATIONTIME and OEONTIME are odd) or (CLKACTIVATIONTIME and OEONTIME are even)

  H = (1 + 0.5 × OEEXTRADELAY) × GPMC_FCLK period, otherwise
- Case where GPMCFCLKDIVIDER = 2

  H = 0.5 × OEEXTRADELAY × GPMC_FCLK period, when (OEONTIME - CLKACTIVATIONTIME) is a multiple of 3

  H = (1 + 0.5 × OEEXTRADELAY) × GPMC_FCLK period, when (OEONTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  H = (2 + 0.5 × OEEXTRADELAY) × GPMC_FCLK period, when (OEONTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{\text{OE}}$ rising edge ($\overline{\text{OE}}$ de-activated):

- Case where [1-0] GPMCFCLKDIVIDER = 0

  H = 0.5 × OEEXTRADELAY × GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 1

  H = 0.5 × OEEXTRADELAY × GPMC_FCLK period, when (CLKACTIVATIONTIME and OEOFFTIME are odd) or (CLKACTIVATIONTIME and OEOFFTIME are even)

  H = (1 + 0.5 × OEEXTRADELAY) × GPMC_FCLK period, otherwise


Copyright © 2011–2016, Texas Instruments Incorporated

- Case where GPMCFCLKDIVIDER = 2

  H = 0.5 × OEEXTRADELAY × GPMC_FCLK period, when (OEOFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  H = (1 + 0.5 × OEEXTRADELAY) × GPMC_FCLK period, when (OEOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  H = (2 + 0.5 × OEEXTRADELAY) × GPMC_FCLK period, when (OEOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{WE}$ falling edge ($\overline{WE}$ activated):

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0

  I = 0.5 × WEEXTRADELAY × GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 1

  I = 0.5 × WEEXTRADELAY × GPMC_FCLK period, when (CLKACTIVATIONTIME and WEONTIME are odd) or (CLKACTIVATIONTIME and WEONTIME are even)

  I = (1 + 0.5 × WEEXTRADELAY) × GPMC_FCLK period, otherwise
- Case where GPMCFCLKDIVIDER = 2

  I = 0.5 × WEEXTRADELAY × GPMC_FCLK period, when (WEONTIME - CLKACTIVATIONTIME) is a multiple of 3

  I = (1 + 0.5 × WEEXTRADELAY) × GPMC_FCLK period, when (WEONTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  I = (2 + 0.5 × WEEXTRADELAY) × GPMC_FCLK period, when (WEONTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{WE}$ rising edge ($\overline{WE}$ de-activated):

- Case where [1-0] GPMCFCLKDIVIDER = 0

  I = 0.5 × WEEXTRADELAY × GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 1

  I = 0.5 × WEEXTRADELAY × GPMC_FCLK period, when (CLKACTIVATIONTIME and WEOFFTIME are odd) or (CLKACTIVATIONTIME and WEOFFTIME are even)

  I = (1 + 0.5 × WEEXTRADELAY) × GPMC_FCLK period otherwise
- Case where GPMCFCLKDIVIDER = 2

  I = 0.5 × WEEXTRADELAY × GPMC_FCLK period, when (WEOFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  I = (1 + 0.5 × WEEXTRADELAY) × GPMC_FCLK period, when (WEOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  I = (2 + 0.5 × WEEXTRADELAY) × GPMC_FCLK period, when (WEOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For GPMC_$\overline{ADV}$ low pulse duration:

- Read operation

  K = (ADVRDOFFTIME - ADVONTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period
- Write operation

  K = (ADVWROFFTIME - ADVONTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

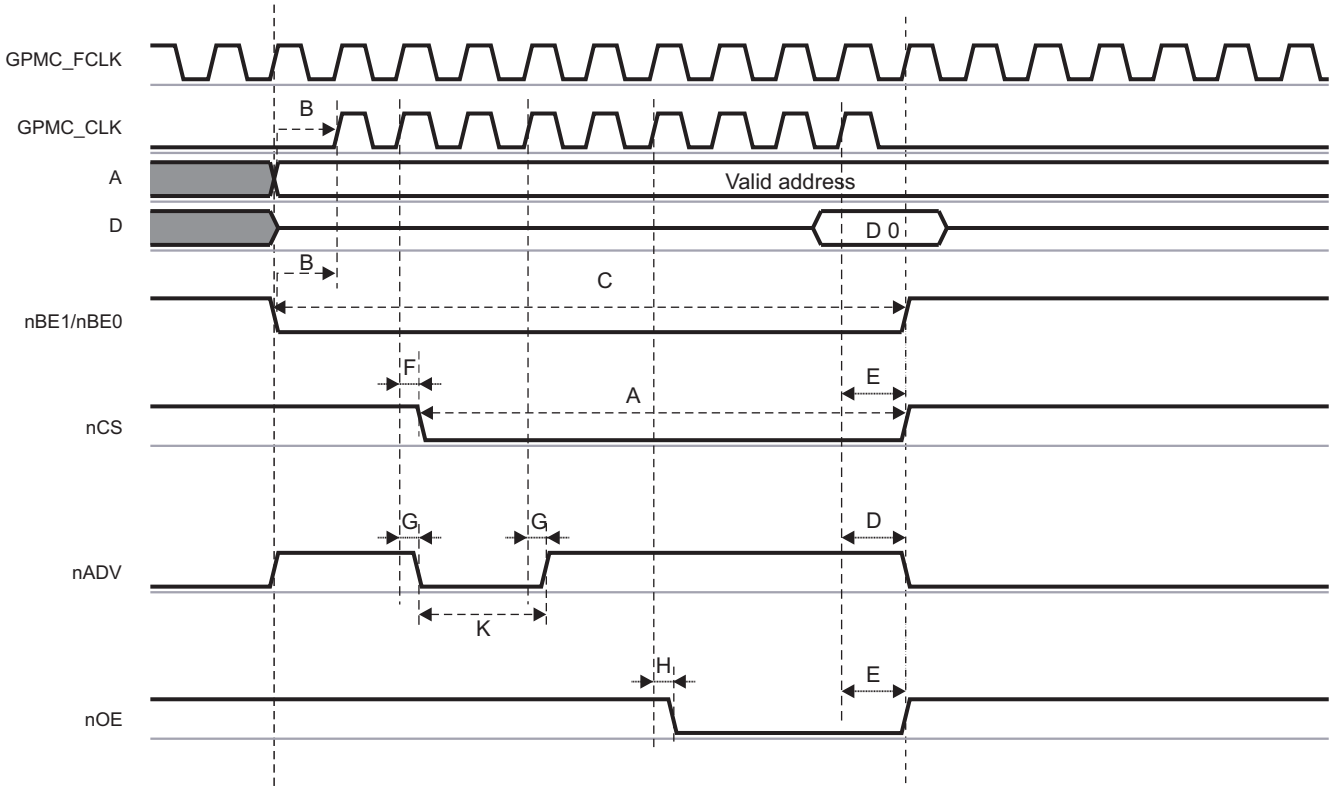
For GPMC_WAIT invalid to first data latching GPMC_CLK edge:

L = WAITMONITORINGTIME × (GPMCFCLKDIVIDER + 1) × GPMC_FCLK period + GPMC_CLK period


Copyright © 2011–2016, Texas Instruments Incorporated

Figure 11-45 shows a synchronous NOR single read simplified example where formulas are associated with signal waves.

**Figure 11-45. Synchronous NOR Single Read Simplified Example**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.3.6.1.3 Asynchronous NOR Flash Timing Parameters Formulas

This section lists all formulas to use in order to calculate asynchronous NOR timing parameters. This is the case when [11-10] DEVICETYPE = 0x0 and when READTYPE or WRITETYPE are set to asynchronous mode.

**Table 11-43. Asynchronous NOR Formulas Description Table**

| Configuration Parameter | Unit | Description |
|---|---|---|
| A | ns | Pulse duration - GPMC_CS low |
| B | ns | Delay time - GPMC_CS valid to GPMC_ADV_ALE invalid |
| C | ns | Delay time - GPMC_CS valid to GPMC_OE invalid (single read) |
| D | ns | Pulse duration - address bus valid - 2nd, 3rd and 4th accesses |
| E | ns | Delay time - GPMC_CS valid to GPMC_WE valid |
| F | ns | Delay time - GPMC_CS valid to GPMC_WE invalid |
| G | ns | Address invalid duration between 2 successive R/W accesses |
| H | ns | Setup time - read data valid before GPMC_OE high |
| I | ns | Delay time - GPMC_CS valid to GPMC_OE invalid (burst read) |
| J | ns | Delay time - address bus valid to GPMC_CS valid |
| | | Delay time - data bus valid to GPMC_CS valid |
| | | Delay time - GPMC_BE0_CLE/GPMC_BE1 valid to GPMC_CS valid |
| K | ns | Delay time - GPMC_CS valid to GPMC_ADV_ALE valid |
| L | ns | Delay time - GPMC_CS valid to GPMC_OE valid |
| M | ns | Delay time - GPMC_CS valid to first data latching edge |
| N | ns | Pulse duration - GPMC_BE0_CLE/GPMC_BE1 valid time |
| O | ns | Delay time - GPMC_CS valid to GPMC_ADV_ALE valid |

The configuration parameters are calculated through the following formulas. Note that these formulas are not exhaustive.

GPMC_CS low pulse:

For single read: A = (CSRDOFFTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

For burst read: A = (CSRDOFFTIME - CSONTIME + (N - 1) × PAGEBURSTACCESSTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period, where N = page burst access number

For single write: A = (CSWROFFTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

For burst write: A = (CSWROFFTIME - CSONTIME + (N - 1) × PAGEBURSTACCESSTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period, where N = page burst access number

GPMC_CS valid to GPMC_ADV_ALE invalid delay:

For reading: B = ((ADVRDOFFTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) + 0.5 × (ADVEXTRADELAY - CSEXTRADELAY)) × GPMC_FCLK period

For writing: B = ((ADVWROFFTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) + 0.5 × (ADVEXTRADELAY - CSEXTRADELAY)) × GPMC_FCLK period

C = ((OEOFFTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) + 0.5 × (OEEXTRADELAY - CSEXTRADELAY)) × GPMC_FCLK period

D = PAGEBURSTACCESSTIME × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

E = ((WEONTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) + 0.5 × (WEEXTRADELAY - CSEXTRADELAY)) × GPMC_FCLK period

F = ((WEOFFTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) + 0.5 × (WEEXTRADELAY - CSEXTRADELAY)) × GPMC_FCLK period

G = CYCLE2CYCLEDELAY × GPMC_FCLK period


Copyright © 2011–2016, Texas Instruments Incorporated

H = ((OEOFFTIME - RDACCESSTIME) × (TIMEPARAGRANULARITY + 11) + 0.5 × OEEXTRADELAY) × GPMC_FCLK period

I = ((OEOFFTIME + (N - 1) × PAGEBURSTACCESSTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) + 0.5 × (OEEXTRADELAY - CSEXTRADELAY)) × GPMC_FCLK period, where N = page burst access number

J = (CSONTIME × (TIMEPARAGRANULARITY + 1) + 0.5 × CSEXTRADELAY) × GPMC_FCLK period

K = ((ADVONTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) + 0.5 × (ADVEXTRADELAY - CSEXTRADELAY)) × GPMC_FCLK period

L = ((OEONTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) + 0.5 × (OEEXTRADELAY - CSEXTRADELAY)) × GPMC_FCLK period

M = ((RDACCESSTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) - 0.5 × CSEXTRADELAY) × GPMC_FCLK period

GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ pulse:

For single read: N = RDCYCLETIME × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period

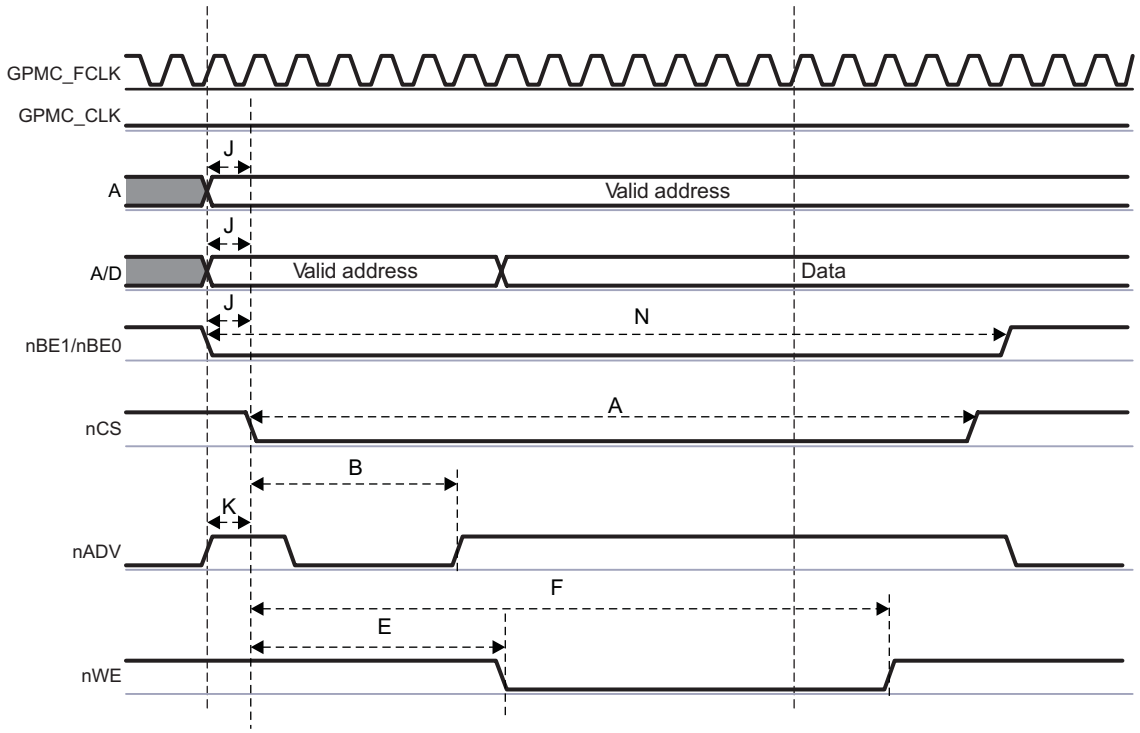
For burst read: N = (RDCYCLETIME + (N - 1) × PAGEBURSTACCESSTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period, where N = page burst access number

For burst write: N = (WRCYCLETIME + (N - 1) × PAGEBURSTACCESSTIME) × (TIMEPARAGRANULARITY + 1) × GPMC_FCLK period, where N = page burst access number

O = ((WRCYCLETIME + (N - 1) × PAGEBURSTACCESSTIME - CSONTIME) × (TIMEPARAGRANULARITY + 1) + 0.5 × (ADVEXTRADELAY - CSEXTRADELAY)) × GPMC_FCLK period

Figure 11-46 shows an asynchronous NOR single write simplified example where formulas are associated with signal waves.

**Figure 11-46. Asynchronous NOR Single Write Simplified Example**




Write multiple access is not supported in asynchronous mode. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.


Copyright © 2011–2016, Texas Instruments Incorporated

## 11.4 Use Cases And Tips

### 11.4.1 How to Set GPMC Timing Parameters for Typical Accesses

#### 11.4.1.1 External Memory Attached to the GPMC Module

As discussed in the introduction to this chapter, the GPMC module supports the following external memory types:

- Asynchronous or synchronous, 8-bit or 16-bit-width memory or device
- 16-bit address/data-multiplexed or not multiplexed NOR flash device
- 8- or 16-bit NAND flash device

The following examples show how to calculate GPMC timing parameters by showing a typical parameter setup for the access to be performed.

The example is based on a 512-Mb multiplexed NOR flash memory with the following characteristics:

- Type: NOR flash (address/data-multiplexed mode)
- Size: 512M bits
- Data Bus: 16 bits wide
- Speed: 100 MHz clock frequency
- Read access time: 80 ns

#### 11.4.1.2 Typical GPMC Setup

Table 11-44 lists some of the I/Os of the GPMC module.

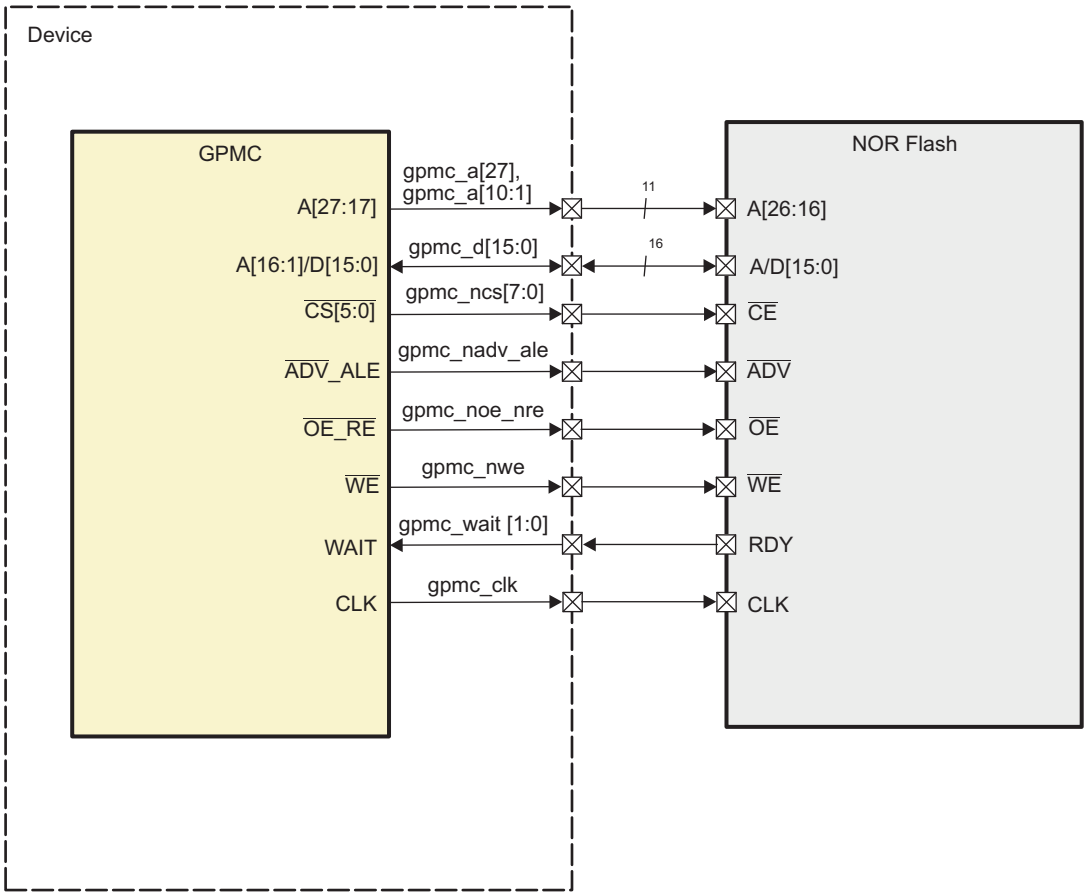
**Table 11-44. GPMC Signals**

| Signal Name | I/O | Description |
|---|---|---|
| GPMC_FCLK | Internal | Functional and interface clock. Acts as the time reference. |
| GPMC_CLK | O | External clock provided to the external device for synchronous operations |
| GPMC_A[27:17] | O | Address |
| GPMC_D[15: 0] | I/O | Data-multiplexed with addresses A[16:1] on memory side |
| $\overline{\text{GPMC_CSx}}$ | O | Chip-select (where x = 0, or 1) |
| GPMC_$\overline{\text{ADV}}$_ALE | O | Address valid enable |
| $\overline{\text{GPMC_OE_RE}}$ | O | Output enable (read access only) |
| $\overline{\text{GPMC_WE}}$ | O | Write enable (write access only) |
| GPMC_WAIT | I | Ready signal from memory device. Indicates when valid burst data is ready to be read |


Copyright © 2011–2016, Texas Instruments Incorporated

Figure 11-47 shows the typical connection between the GPMC module and an attached NOR Flash memory.

**Figure 11-47. GPMC Connection to an External NOR Flash Memory**




The following sections demonstrate how to calculate GPMC parameters for three access types:

- Synchronous burst read
- Asynchronous read
- Asynchronous single write


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.4.1.3 GPMC Configuration for Synchronous Burst Read Access

The clock runs at 100 MHz (f = 100 MHz; T = 10 ns).

Table 11-45 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 11-46 shows how to calculate timings for the GPMC using the memory parameters.

Figure 11-48 shows the synchronous burst read access.

**Table 11-45. Useful Timing Parameters on the Memory Side**

| AC Read Characteristics on the Memory Side | Description | Duration (ns) |
|---|---|---|
| tCES | $\overline{CS}$ setup time to clock | 0 |
| tACS | Address setup time to clock | 3 |
| tIACC | Synchronous access time | 80 |
| tBACC | Burst access time valid clock to output delay | 5,2 |
| tCEZ | Chip-select to High-Impedance | 7 |
| tOEZ | Output enable to High-Impedance | 7 |
| tAVC | $\overline{ADV}$ setup time | 6 |
| tAVD | $\overline{AVD}$ pulse | 6 |
| tACH | Address hold time from clock | 3 |

The following terms, which describe the timing interface between the controller and its attached device, are used to calculate the timing parameters on the GPMC side:

- Read Access time (GPMC side): Time required to activate the clock + read access time requested on the memory side + data setup time required for optimal capture of a burst of data
- Data setup time (GPMC side): Ensures a good capture of a burst of data (as opposed to taking a burst of data out). One word of data is processed in one clock cycle (T = 10 ns). The read access time between 2 bursts of data is tBACC = 5,2 ns. Therefore, data setup time is a clock period - tBACC = 4,8 ns of data setup.
- Access completion (GPMC side): (Different from page burst access time) Time required between the last burst access and access completion: $\overline{CS}$/$\overline{OE}$ hold time ($\overline{CS}$ and $\overline{OE}$ must be released at the end of an access. These signals are held to allow the access to complete).
- Read cycle time (GPMC side): Read Access time + access completion
- Write cycle time for burst access: Not supported for NOR flash memory

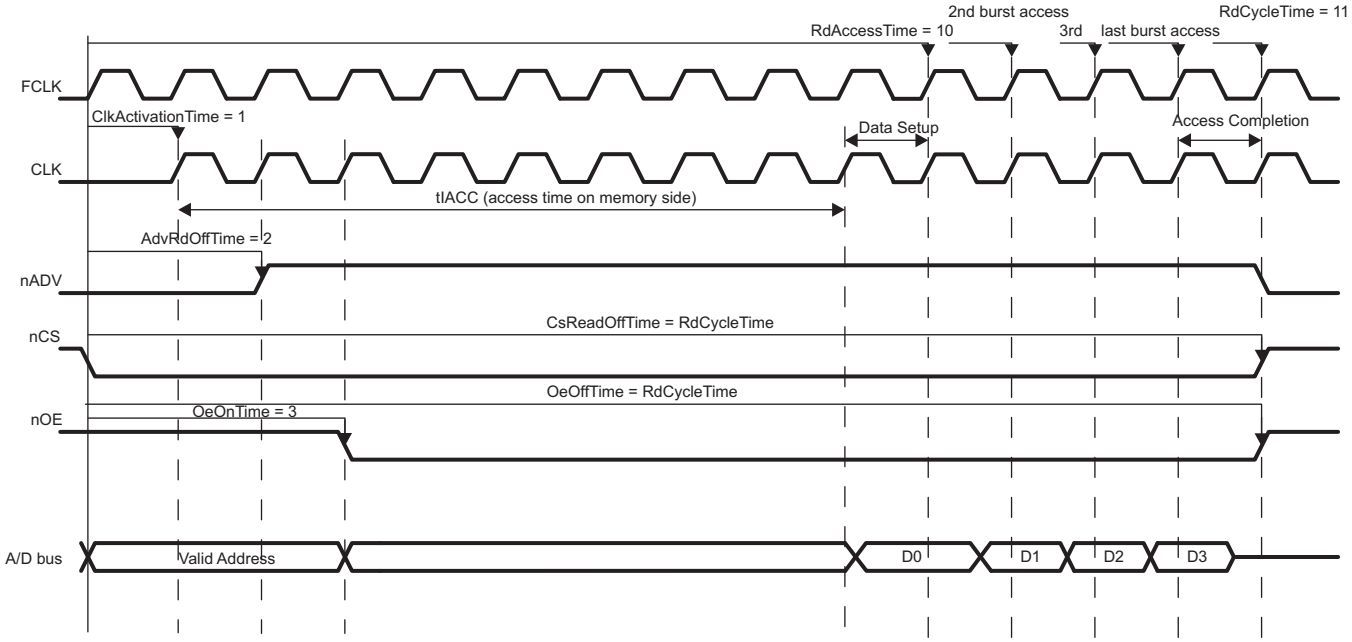


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-46. Calculating GPMC Timing Parameters**

| Parameter Name on GPMC Side | Formula | Duration (ns) | Number of Clock Cycles (F = 100 MHz) | GPMC Register Configurations |
|---|---|---|---|---|
| GPMC FCLK Divider | - | - | - | GPMCFCLKDIVIDER = 0 |
| ClkActivationTime | min ( tCES, tACS) | 3 | 1 | CLKACTIVATIONTIME = 1 |
| RdAccessTime | roundmax (ClkActivationTime + tIACC + DataSetupTime) | 94,8: (10 + 80 + 4,8) | 10 : roundmax (94,8 / 10) | RDACCESSTIME = Ah |
| PageBurstAccessTime | roundmax (tBACC) | roundmax (5,2) | 1 | PAGEBURSTACCESSTIME = 1 |
| RdCycleTime | RdAccessTime + max ( tCEZ, tOEZ) | 101,8: (94,8 + 7) | 11 | RDCYCLETIME = Bh |
| CsOnTime | tCES | 0 | 0 | CSONTIME = 0 |
| CsReadOffTime | RdCycleTime | - | 11 | CSRDOFFTIME = Bh |
| AdvOnTime | tAVC | 0 | 0 | ADVONTIME = 0 |
| AdvRdOffTime | tAVD + tAVC | 12 | 2 | ADVRDOFFTIME = 2h |
| OeOnTime | (ClkActivationTime + tACH) < OeOnTime < (ClkActivationTime + tIACC) | - | 3, for instance | OEONTIME = 3h |
| OeOffTime | RdCycleTime | - | 11 | OEOFFTIME = Bh |

**Figure 11-48. Synchronous Burst Read Access (Timing Parameters in Clock Cycles)**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.4.1.4 GPMC Configuration for Asynchronous Read Access

The clock runs at 100 MHz ( f = 100 MHz; T = 10 ns).

Table 11-47 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 11-48 shows how to calculate timings for the GPMC using the memory parameters.

Figure 11-49 shows the asynchronous read access.

**Table 11-47. AC Characteristics for Asynchronous Read Access**

| AC Read Characteristics on the Memory Side | Description | Duration (ns) |
|---|---|---|
| tCE | Read Access time from $\overline{CS}$ low | 80 |
| tAAVDS | Address setup time to rising edge of $\overline{ADV}$ | 3 |
| tAVDP | $\overline{ADV}$ low time | 6 |
| tCAS | $\overline{CS}$ setup time to $\overline{ADV}$ | 0 |
| tOE | Output enable to output valid | 6 |
| tOEZ | Output enable to High-Impedance | 7 |

Use the following formula to calculate the RdCycleTime parameter for this typical access:

RdCycleTime = RdAccessTime + AccessCompletion = RdAccessTime + 1 clock cycle + tOEZ

- First, on the memory side, the external memory makes the data available to the output bus. This is the memory-side read access time defined in Table 11-48: the number of clock cycles between the address capture ($\overline{ADV}$ rising edge) and the data valid on the output bus.

The GPMC requires some hold time to allow the data to be captured correctly and the access to be finished.

- To read the data correctly, the GPMC must be configured to meet the data setup time requirement of the memory; the GPMC module captures the data on the next rising edge. This is access time on the GPMC side.
- There must also be a data hold time for correctly reading the data (checking that there is no $\overline{OE}$/$\overline{CS}$ deassertion while reading the data). This data hold time is 1 clock cycle (that is, RdAccessTime + 1).
- To complete the access, $\overline{OE}$/$\overline{CS}$ signals are driven to high-impedance. RdAccessTime + 1 cycle + tOEZ is the read cycle time.
- Addresses can now be relatched and a new read cycle begins.

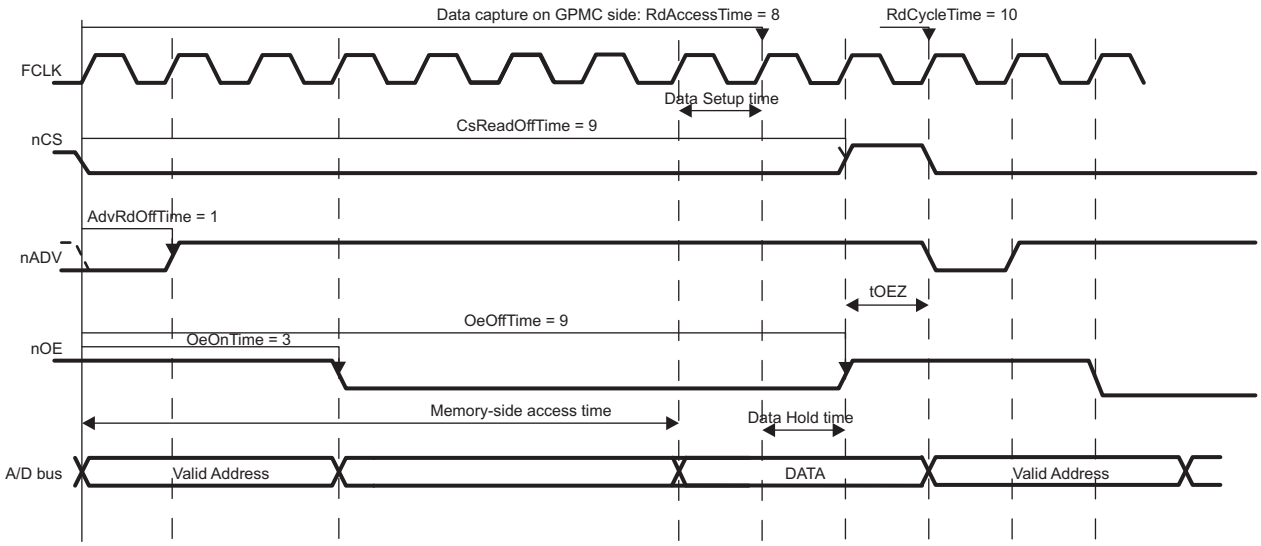


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-48. GPMC Timing Parameters for Asynchronous Read Access**

| Parameter Name on GPMC side | Formula | Duration (ns) | Number of Clock Cycles (F = 100 MHz) | GPMC Register Configurations |
|---|---|---|---|---|
| ClkActivationTime | n/a (asynchronous mode) | | | |
| RdAccessTime | round max (tCE) | 80 | 8 | RDACCESSTIME = 8h |
| PageBurstAccessTime | n/a (single access) | | | |
| RdCycleTime | RdAccessTime + 1cycle + tOEZ | 97 | 10 | RDCYCLETIME = Ah |
| CsOnTime | tCAS | 0 | 0 | CSONTIME = 0 |
| CsReadOffTime | RdAccessTime + 1 cycle | 90 | 9 | CSRDOFFTIME = 9h |
| AdvOnTime | tAAVDS | 3 | 1 | ADVONTIME = 1 |
| AdvRdOffTime | tAAVDS + tAVDP | 9 | 1 | ADVRDOFFTIME = 1 |
| OeOnTime | OeOnTime ≥ AdvRdOffTime (multiplexed mode) | - | 3, for instance | OEONTIME = 3h |
| OeOffTime | RdAccessTime + 1cycle | 90 | 9 | OEOFFTIME = 9h |

**Figure 11-49. Asynchronous Single Read Access (Timing Parameters in Clock Cycles)**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.4.1.5 GPMC Configuration for Asynchronous Single Write Access

The clock runs at 100 MHz: (f = 100 MHz; T = 10 ns).

Table 11-49 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 11-50 shows how to calculate timings for the GPMC using the memory parameters.

Figure 11-50 shows the synchronous burst write access.

**Table 11-49. AC Characteristics for Asynchronous Single Write (Memory Side)**

| AC Characteristics on the Memory Side | Description | Duration (ns) |
|---|---|---|
| tWC | Write cycle time | 60 |
| tAVDP | $\overline{ADV}$ low time | 6 |
| tWPH | Write pulse width high | 20 |
| tCS | $\overline{CS}$ setup time to $\overline{WE}$ | 3 |
| tCAS | $\overline{CS}$ setup time to $\overline{ADV}$ | 0 |
| tAVSC | $\overline{ADV}$ setup time | 3 |

For asynchronous single write access, write cycle time is WrCycleTime = WeOffTime + AccessCompletion = WeOffTime + 1. For the AccesCompletion, the GPMC requires 1 cycle of data hold time ($\overline{CS}$ de-assertion).

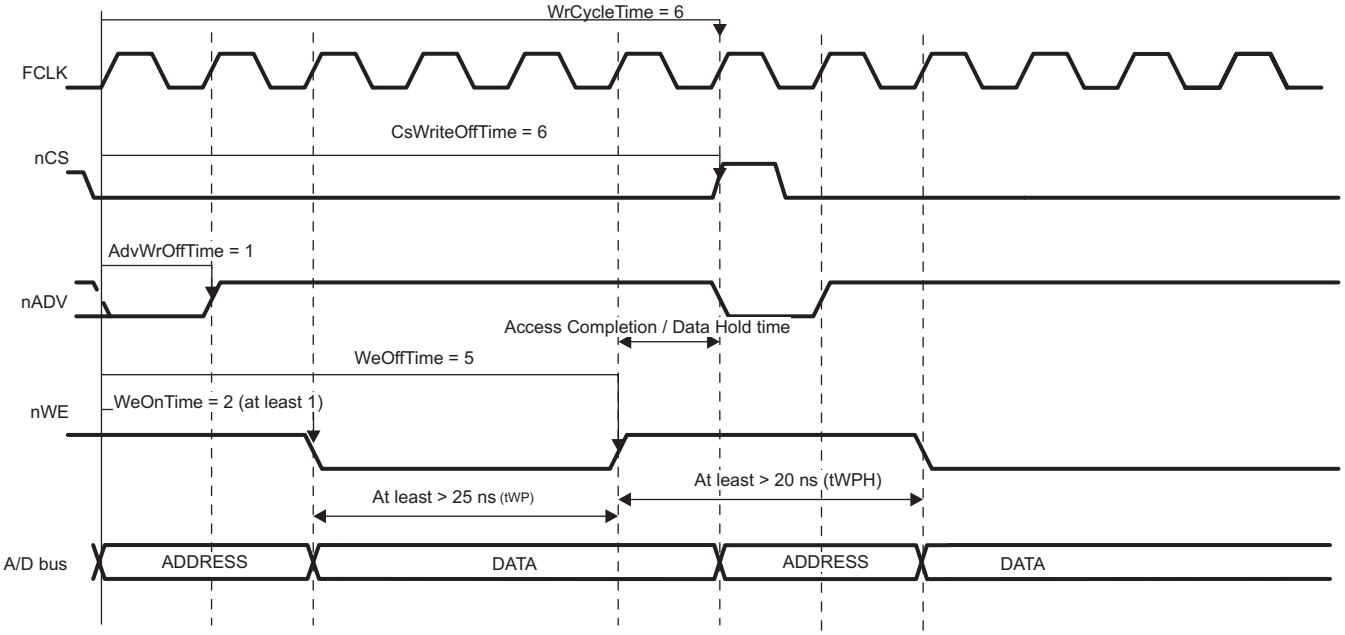


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-50. GPMC Timing Parameters for Asynchronous Single Write**

| Parameter Name on GPMC side | Formula | Duration (ns) | Number of Clock Cycles (F = 100 MHz) | GPMC Register Configurations |
|---|---|---|---|---|
| ClkActivationTime | n/a (asynchronous mode) | | | |
| WrAccessTime | Applicable only to WAITMONITORING (the value is the same as for read access) | | | |
| PageBurstAccessTime | n/a (single access) | | | |
| WrCycleTime | WeOffTime + AccessCompletion | 58 | 6 | WRCYCLETIME = 6h |
| CsOnTime | tCAS | 0 | 0 | CSONTIME = 0 |
| CsWrOffTime | WeOffTime + 1 cycle | 58 | 6 | CSWROFFTIME = 6h |
| AdvOnTime | tAVSC | 3 | 1 | ADVONTIME = 1 |
| AdvWrOffTime | tAVSC + tAVDP | 9 | 1 | ADVWROFFTIME = 1 |
| WeOnTime | tCS | 3 | 1 | WEONTIME = 1 |
| WeOffTime | tCS + tWP + tWPH | 48 | 5 | WEOFFTIME = 5h |

**Figure 11-50. Asynchronous Single Write Access (Timing Parameters in Clock Cycles)**






Copyright © 2011–2016, Texas Instruments Incorporated

### 11.4.2 How to Choose a Suitable Memory to Use With the GPMC

This section is intended to help the user select a suitable memory device to interface with the GPMC controller.

#### 11.4.2.1 Supported Memories or Devices

NAND flash and NOR flash architectures are the two flash technologies. The GPMC supports various types of external memory or device, basically any one that supports NAND or NOR protocols:

- 8- and 16-bit width asynchronous or synchronous memory or device (8-bit: non burst device only)
- 16-bit address and data multiplexed NOR flash devices (pSRAM, , ...)
- 8- and 16-bit NAND flash device

#### 11.4.2.2 Memory Pin Multiplexing

This section highlights the interfacing differences of the GPMC supported memories (see Table 11-51).

**Table 11-51. Supported Memory Interfaces**

| Function | 16-bit Address/Data Non-Muxed Flash | 16-bit Address/Data Muxed pSRAM or NOR Flash | OneNAND | 16-bit NAND | 8-bit NAND |
|---|---|---|---|---|---|
| GPMC_A[27] | A26 | A27 | | | |
| GPMC_A[26] | A25 | Not Used | | | |
| GPMC_A[25] | A24 | Not Used | | | |
| GPMC_A[24] | A23 | Not Used | | | |
| GPMC_A[23] | A22 | Not Used | | | |
| GPMC_A[22] | A21 | Not Used | | | |
| GPMC_A[21] | A20 | Not Used | | | |
| GPMC_A[20] | A19 | Not Used | | | |
| GPMC_A[19] | A18 | Not Used | | | |
| GPMC_A[18] | A17 | Not Used | | | |
| GPMC_A[17] | A16 | Not Used | | | |
| GPMC_A[16] | A15 | Not Used | | | |
| GPMC_A[15] | A14 | Not Used | | | |
| GPMC_A[14] | A13 | Not Used | | | |
| GPMC_A[13] | A12 | Not Used | | | |
| GPMC_A[12] | A11 | Not Used | | | |
| GPMC_A[11] | A10 | Not Used | | | |
| GPMC_A[10] | A9 | A26 | | | |
| GPMC_A[9] | A8 | A25 | | | |
| GPMC_A[8] | A7 | A24 | | | |
| GPMC_A[7] | A6 | A23 | | | |
| GPMC_A[6] | A5 | A22 | | | |
| GPMC_A[5] | A4 | A21 | | | |
| GPMC_A[4] | A3 | A20 | | | |
| GPMC_A[3] | A2 | A19 | | | |
| GPMC_A[2] | A1 | A18 | | | |
| GPMC_A[1] | A0 | A17 | | | |
| GPMC_A[0] | Not Used | Not Used | | | |
| GPMC_D[15] | D15 | D15 or A16 | D15 or A16 | IO15 | |
| GPMC_D[14] | D14 | D14 or A15 | D14 or A15 | IO14 | |
| GPMC_D[13] | D13 | D13 or A14 | D13 or A14 | IO13 | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-51. Supported Memory Interfaces (continued)**

| Function | 16-bit Address/Data Non-Muxed Flash | 16-bit Address/Data Muxed pSRAM or NOR Flash | OneNAND | 16-bit NAND | 8-bit NAND |
|---|---|---|---|---|---|
| GPMC_D[12] | D12 | D12 or A13 | D12 or A13 | IO12 | |
| GPMC_D[11] | D11 | D11 or A12 | D11 or A12 | IO11 | |
| GPMC_D[10] | D10 | D10 or A11 | D10 or A11 | IO10 | |
| GPMC_D[9] | D9 | D9 or A10 | D9 or A10 | IO9 | |
| GPMC_D[8] | D8 | D8 or A9 | D8 or A9 | IO8 | |
| GPMC_D[7] | D7 | D7 or A8 | D7 or A8 | IO7 | IO7 |
| GPMC_D[6] | D6 | D6 or A7 | D6 or A7 | IO6 | IO6 |
| GPMC_D[5] | D5 | D5 or A6 | D5 or A6 | IO5 | IO5 |
| GPMC_D[4] | D4 | D4 or A5 | D4 or A5 | IO4 | IO4 |
| GPMC_D[3] | D3 | D3 or A4 | D3 or A4 | IO3 | IO3 |
| GPMC_D[2] | D2 | D2 or A3 | D2 or A3 | IO2 | IO2 |
| GPMC_D[1] | D1 | D1 or A2 | D1 or A2 | IO1 | IO1 |
| GPMC_D[0] | D0 | D0 or A1 | D0 or A1 | IO0 | IO0 |
| GPMC_CLK | CLK | CLK | CLK | | |
| GPMC_$\overline{CS0}$ | $\overline{CS0}$ (Chip Select) | $\overline{CS0}$ (Chip Select) | $\overline{CS0}$ (Chip Select) | $\overline{CE0}$ (Chip Enable) | $\overline{CE0}$ (Chip Enable) |
| GPMC_$\overline{CS1}$ | $\overline{CS1}$ | $\overline{CS1}$ | $\overline{CS1}$ | $\overline{CE1}$ | $\overline{CE1}$ |
| GPMC_$\overline{CS2}$ | $\overline{CS2}$ | $\overline{CS2}$ | $\overline{CS2}$ | $\overline{CE2}$ | $\overline{CE2}$ |
| GPMC_$\overline{CS3}$ | $\overline{CS3}$ | $\overline{CS3}$ | $\overline{CS3}$ | $\overline{CE3}$ | $\overline{CE3}$ |
| GPMC_$\overline{CS4}$ | $\overline{CS4}$ | $\overline{CS4}$ | $\overline{CS4}$ | $\overline{CE4}$ | $\overline{CE4}$ |
| GPMC_$\overline{CS5}$ | $\overline{CS5}$ | $\overline{CS5}$ | $\overline{CS5}$ | $\overline{CE5}$ | $\overline{CE5}$ |
| GPMC_$\overline{ADV}$_ALE | $\overline{ADV}$ (Address Value) | $\overline{ADV}$ (Address Value) | $\overline{ADV}$ (Address Value) | ALE (address latch enable) | ALE (address latch enable) |
| GPMC_$\overline{OE}$_$\overline{RE}$ | $\overline{OE}$ (Output Enable) | $\overline{OE}$ (Output Enable) | $\overline{OE}$ (Output Enable) | $\overline{RE}$ (read enable) | $\overline{RE}$ (read enable) |
| GPMC_$\overline{WE}$ | $\overline{WE}$ (Write Enable) | $\overline{WE}$ (Write Enable) | $\overline{WE}$ (Write Enable) | $\overline{WE}$ (write enable) | $\overline{WE}$ (write enable) |
| GPMC_$\overline{BE0}$_CLE | $\overline{BE0}$ (Byte Enable) | $\overline{BE0}$ (Byte Enable) | $\overline{BE0}$ (Byte Enable) | CLE (command latch enable) | CLE (command latch enable) |
| GPMC_$\overline{BE1}$ | $\overline{BE1}$ | $\overline{BE1}$ | $\overline{BE1}$ | | |
| GPMC_WAIT | WAIT0 | WAIT0 | WAIT0 | R/$\overline{B0}$ (ready/busy) | R/$\overline{B0}$ (ready/busy) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.4.2.3 NAND Interface Protocol

NAND flash architecture, introduced in 1989, is a flash technology. NAND is a page-oriented memory device, that is, read and write accesses are done by pages. NAND achieves great density by sharing common areas of the storage transistor, which creates strings of serially connected transistors (in NOR devices, each transistor stands alone). Thanks to its high density NAND is best suited to devices requiring high capacity data storage, such as pictures, music, or data files. NAND non-volatility, makes of it a good storage solution for many applications where mobility, low power, and speed are key factors. Low pin count and simple interface are other advantages of NAND.

Table 11-52 summarizes the NAND interface signals level applied to external device or memories.

**Table 11-52. NAND Interface Bus Operations Summary**

| Bus Operation | CLE | ALE | $\overline{CE}$ | $\overline{WE}$ | $\overline{RE}$ |
|---|---|---|---|---|---|
| Read (cmd input) | H | L | L | RE | H |
| Read (add input) | L | H | L | RE | H |
| Write (cmd input) | H | L | L | RE | H |
| Write (add input) | L | H | L | RE | H |
| Data input | L | L | L | RE | H |
| Data output | L | L | L | H | FE |
| Busy (during read) | x | x | H | H | H |
| Busy (during program) | x | x | x | x | x |
| Busy (during erase) | x | x | x | x | x |
| Write protect | x | x | x | x | x |
| Stand-by | x | x | H | x | x |

### 11.4.2.4 NOR Interface Protocol

NOR flash architecture, introduced in 1988, is a flash technology. Unlike NAND, which is a sequential access device, NOR is directly addressable; i.e., it is designed to be a random access device. NOR is best suited to devices used to store and run code or firmware, usually in small capacities. While NOR has fast read capabilities it has slow write and erase functions compared to NAND architecture.

Table 11-53 summarizes the NOR interface signals level applied to external device or memories.

**Table 11-53. NOR Interface Bus Operations Summary**

| Bus Operation | CLK | $\overline{ADV}$ | $\overline{CS}$ | $\overline{OE}$ | $\overline{WE}$ | WAIT | DQ[15:0] |
|---|---|---|---|---|---|---|---|
| Read (asynchronous) | x | L | L | L | H | Asserted | Output |
| Read (synchronous) | Running | L | L | L | H | Driven | Output |
| Read (burst suspend) | Halted | x | L | H | H | Active | Output |
| Write | x | L | L | H | L | Asserted | Input |
| Output disable | x | x | L | H | H | Asserted | High-Z |
| Standby | x | x | H | x | x | High-Z | High-Z |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 11.4.2.5 Other Technologies

Other supported device type interact with the GPMC through the NOR interface protocol.

OneNAND Flash is a high-density and low-power memory device. It is based on single- or multi-level-cell NAND core with SRAM and logic, and interfaces as a synchronous NOR Flash, plus has synchronous write capability. It reads faster than conventional NAND and writes faster than conventionnal NOR flash. Hence, it is appropriate for both mass storage and code storage.

pSRAM stands for pseudo-static random access memory. pSRAM is a low-power memory device for mobile applications. pSRAM is based on the DRAM cell with internal refresh and address control features, and interfaces as a synchronous NOR Flash, plus has synchronous write capability.

#### 11.4.2.6 Supported Protocols

The GPMC supports the following interface protocols when communicating with external memory or external devices:

- Asynchronous read/write access
- Asynchronous read page access (4-8-16 Word16)
- Synchronous read/write access
- Synchronous read burst access without wrap capability (4-8-16 Word16)
- Synchronous read burst access with wrap capability (4-8-16 Word16)

#### 11.4.2.7 GPMC Features and Settings

This section lists GPMC features and settings:

- Supported device type: up to four NAND or NOR protocol external memories or devices
- Operating Voltage: 3.3V
- Maximum GPMC addressing capability: 256 MBytes divided into six chip-selects
- Maximum supported memory size: 256 MBytes (must be a power-of-2)
- Minimum supported memory size: 16 MBytes (must be a power-of-2). Aliasing occurs when addressing smaller memories.
- Data path to external memory or device: 8- and 16-bit wide
- Burst and page access: burst of 4-8-16 Word16
- Supports bus keeping
- Supports bus turn around


Copyright © 2011–2016, Texas Instruments Incorporated

## 11.5 Registers

Table 11-54 provides a summary of the GPMC registers. All GPMC registers are aligned to 32-bit address boundaries. All register file accesses, except the GPMC_NAND_DATA_i register, are little-endian. If the GPMC_NAND_DATA_i register is accessed, the endianness is access-dependent.

In this section, i corresponds to the chip-select number, i = 0 to 5.

**Table 11-54. GPMC Registers**

| Address Offset | Register Name | Section |
|---|---|---|
| 0h | GPMC_REVISION | Section 11.5.1 |
| 10h | GPMC_SYSCONFIG | Section 11.5.2 |
| 14h | GPMC_SYSSTATUS | Section 11.5.3 |
| 18h | GPMC_IRQSTATUS | Section 11.5.4 |
| 1Ch | GPMC_IRQENABLE | Section 11.5.5 |
| 40h | GPMC_TIMEOUT_CONTROL | Section 11.5.6 |
| 44h | GPMC_ERR_ADDRESS | Section 11.5.7 |
| 48h | GPMC_ERR_TYPE | Section 11.5.8 |
| 50h | GPMC_CONFIG | Section 11.5.9 |
| 54h | GPMC_STATUS | Section 11.5.10 |
| 60h + (30h × i) | GPMC_CONFIG1_i[(1)] | Section 11.5.11 |
| 64h + (30h × i) | GPMC_CONFIG2_i[(1)] | Section 11.5.12 |
| 68h + (30h × i) | GPMC_CONFIG3_i[(1)] | Section 11.5.13 |
| 6Ch + (30h × i) | GPMC_CONFIG4_i[(1)] | Section 11.5.14 |
| 70h + (30h × i) | GPMC_CONFIG5_i[(1)] | Section 11.5.15 |
| 74h + (30h × i) | GPMC_CONFIG6_i[(1)] | Section 11.5.16 |
| 78h + (30h × i) | GPMC_CONFIG7_i[(1)] | Section 11.5.17 |
| 7Ch + (30h × i) | GPMC_NAND_COMMAND_i[(1)] | Section 11.5.18 |
| 80h + (30h × i) | GPMC_NAND_ADDRESS_i[(1)] | Section 11.5.19 |
| 84h + (30h × i) | GPMC_NAND_DATA_i[(1)] | Section 11.5.20 |
| 1E0h | GPMC_PREFETCH_CONFIG1 | Section 11.5.21 |
| 1E4h | GPMC_PREFETCH_CONFIG2 | Section 11.5.22 |
| 1ECh | GPMC_PREFETCH_CONTROL | Section 11.5.23 |
| 1F0h | GPMC_PREFETCH_STATUS | Section 11.5.24 |
| 1F4h | GPMC_ECC_CONFIG | Section 11.5.25 |
| 1F8h | GPMC_ECC_CONTROL | Section 11.5.26 |
| 1FCh | GPMC_ECC_SIZE_CONFIG | Section 11.5.27 |
| 200h + (4h × j) | GPMC_ECCj_RESULT[(2)] | Section 11.5.28 |
| 240h + (10h × i) | GPMC_BCH_RESULT0_i[(1)] | Section 11.5.29 |
| 244h + (10h × i) | GPMC_BCH_RESULT1_i[(1)] | Section 11.5.30 |
| 248h + (10h × i) | GPMC_BCH_RESULT2_i[(1)] | Section 11.5.31 |
| 24Ch + (10h × i) | GPMC_BCH_RESULT3_i[(1)] | Section 11.5.32 |
| 2D0h | GPMC_BCH_SWDATA | Section 11.5.36 |
| 300h + (10h × i) | GPMC_BCH_RESULT4_i[(1)] | Section 11.5.33 |
| 304h + (10h × i) | GPMC_BCH_RESULT5_i[(1)] | Section 11.5.34 |
| 308h + (10h × i) | GPMC_BCH_RESULT6_i[(1)] | Section 11.5.35 |

(1) i = 0 to 5 for GPMC
(2) j = 0 to 8 for GPMC


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.1 GPMC_REVISION

This register contains the IP revision code.

**Figure 11-51. GPMC_REVISION**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | REV | |
| R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-55. GPMC_REVISION Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | REV | 0-FFh | IP revision. Major revision is [7-4]. Minor revision is [3-0].<br>Examples: 10h for revision 1.0, 21h for revision 2.1. |

### 11.5.2 GPMC_SYSCONFIG

This register controls the various parameters of the OCP interface.

**Figure 11-52. GPMC_SYSCONFIG**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | SIDLEMODE | | Rsvd | SOFTRESET | AUTOIDLE |
| R-0 | | R/W | | R-0 | R/W | R/W |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-56. GPMC_SYSCONFIG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Reserved |
| 4-3 | SIDLEMODE | | Idle mode |
| | | 0 | Force-idle. An idle request is acknowledged unconditionally |
| | | 1h | No-idle. An idle request is never acknowledged |
| | | 2h | Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module |
| | | 3h | Reserved |
| 2 | Reserved | 0 | Reserved |
| 1 | SOFTRESET | | Software reset (Set 1 to this bit triggers a module reset. This bit is automatically reset by hardware. During reads, it always returns 0) |
| | | 0 | Normal mode |
| | | 1 | The module is reset |
| 0 | AUTOIDLE | | Internal OCP clock gating strategy |
| | | 0 | Interface clock is free-running |
| | | 1 | Automatic Interface clock gating strategy is applied, based on the Interconnect activity |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.3 GPMC_SYSSTATUS

This register provides status information about the module, excluding the interrupt status information.

**Figure 11-53. GPMC_SYSSTATUS**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | RESETDONE |
| R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-57. GPMC_SYSSTATUS Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | RESETDONE | | Internal reset monitoring |
| | | R0 | Internal module reset in on-going |
| | | R1 | Reset completed |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.4 GPMC_IRQSTATUS

This interrupt status register regroups all the status of the module internal events that can generate an interrupt.

**Figure 11-54. GPMC_IRQSTATUS**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 |
|---|---|---|---|
| Reserved | | WAIT1EDGE DETECTIONSTATUS | WAIT0EDGE DETECTIONSTATUS |
| R-0 | | R/W-0 | R/W-0 |

| 7 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | TERMINAL COUNTSTATUS | FIFOEVENT STATUS |
| R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-58. GPMC_IRQSTATUS Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9 | WAIT1EDGEDETECTIONSTATUS | | Status of the Wait1 Edge Detection interrupt |
| | | R0 | A transition on WAIT1 input pin has not been detected |
| | | W0 | WAIT1EDGEDETECTIONSTATUS bit unchanged |
| | | R1 | A transition on WAIT1 input pin has been detected |
| | | W1 | WAIT1EDGEDETECTIONSTATUS bit is reset |
| 8 | WAIT0EDGEDETECTIONSTATUS | | Status of the Wait0 Edge Detection interrupt |
| | | R0 | A transition on WAIT0 input pin has not been detected |
| | | W0 | WAIT0EDGEDETECTIONSTATUS bit unchanged |
| | | R1 | A transition on WAIT0 input pin has been detected |
| | | W1 | WAIT0EDGEDETECTIONSTATUS bit is reset |
| 7-2 | Reserved | 0 | Reserved |
| 1 | TERMINALCOUNTSTATUS | | Status of the TerminalCountEvent interrupt |
| | | R0 | Indicates that CountValue is greater than 0 |
| | | W0 | TERMINALCOUNTSTATUS bit unchanged |
| | | R1 | Indicates that CountValue is equal to 0 |
| | | W1 | TERMINALCOUNTSTATUS bit is reset |
| 0 | FIFOEVENTSTATUS | | Status of the FIFOEvent interrupt |
| | | R0 | Indicates than less than GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and less than FIFOTHRESHOLD bytes free places are available in write-posting mode. |
| | | W0 | FIFOEVENTSTATUS bit unchanged |
| | | R1 | Indicates than at least GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and at least FIFOTHRESHOLD bytes free places are available in write-posting mode. |
| | | W1 | FIFOEVENTSTATUS bit is reset |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.5 GPMC_IRQENABLE

The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on a event-by-event basis.

**Figure 11-55. GPMC_IRQENABLE**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 |
|---|---|---|---|
| Reserved | | WAIT1EDGE DETECTIONENABLE | WAIT0EDGE DETECTIONENABLE |
| R-0 | | R/W-0 | R/W-0 |

| 7 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | TERMINAL COUNTENABLE | FIFOEVENT ENABLE |
| R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-59. GPMC_IRQENABLE Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9 | WAIT1EDGEDETECTIONENABLE | | Enables the Wait1 Edge Detection interrupt |
| | | 0 | Wait1EdgeDetection interrupt is masked |
| | | 1 | Wait1EdgeDetection event generates an interrupt if occurs |
| 8 | WAIT0EDGEDETECTIONENABLE | | Enables the Wait0 Edge Detection interrupt |
| | | 0 | Wait0EdgeDetection interrupt is masked |
| | | 1 | Wait0EdgeDetection event generates an interrupt if occurs |
| 7-2 | Reserved | 0 | Reserved |
| 1 | TERMINALCOUNTEVENTENABLE | | Enables TerminalCountEvent interrupt issuing in pre-fetch or write posting mode |
| | | 0 | TerminalCountEvent interrupt is masked |
| | | 1 | TerminalCountEvent interrupt is not masked |
| 0 | FIFOEVENTENABLE | | Enables the FIFOEvent interrupt |
| | | 0 | FIFOEvent interrupt is masked |
| | | 1 | FIFOEvent interrupt is not masked |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.6 GPMC_TIMEOUT_CONTROL

The GPMC_TIMEOUT_CONTROL register allows the user to set the start value of the timeout counter

**Figure 11-56. GPMC_TIMEOUT_CONTROL**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 13 | 12 | 4 | 3 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | TIMEOUTSTARTVALUE | | Reserved | | TIMEOUTENABLE |
| R-0 | | R/W-0 | | R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-60. GPMC_TIMEOUT_CONTROL Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-13 | Reserved | 0 | Reserved |
| 12-4 | TIMEOUTSTARTVALUE | 0-1FFh | Start value of the time-out counter (000 corresponds to 0 GPMC.FCLK cycle, 1h corresponds to 1 GPMC.FCLK cycle, and 1FFh corresponds to 511 GPMC.FCLK cycles) |
| 3-1 | Reserved | 0 | Reserved |
| 0 | TIMEOUTENABLE | | Enable bit of the TimeOut feature |
| | | 0 | TimeOut feature is disabled |
| | | 1 | TimeOut feature is enabled |

### 11.5.7 GPMC_ERR_ADDRESS

The GPMC_ERR_ADDRESS register stores the address of the illegal access when an error occurs.

**Figure 11-57. GPMC_ERR_ADDRESS**

| 31 | 30 | 0 |
|---|---|---|
| Rsvd | ILLEGALADD | |
| R-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 11-61. GPMC_ERR_ADDRESS Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved |
| 30-0 | ILLEGALADD | 0-7FFF FFFFh | Address of illegal access: A30 (0 for memory region, 1 for GPMC register region) and A29-A0 (1GByte maximum) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.8 GPMC_ERR_TYPE

The GPMC_ERR_TYPE register stores the type of error when an error occurs.

**Figure 11-58. GPMC_ERR_TYPE**

| 31 | | | | | | 16 |
|---|---|---|---|---|---|---|
| Reserved | | | | | | |
| R-0 | | | | | | |

| 15 | | | | 11 | 10 | | | 8 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | ILLEGALMCMD | | | |
| R-0 | | | | | R/W-0 | | | |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | ERRORNOT SUPPADD | ERRORNOT SUPPMCMD | ERROR TIMEOUT | Reserved | ERRORVALID |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-62. GPMC_ERR_TYPE Field Descriptions**

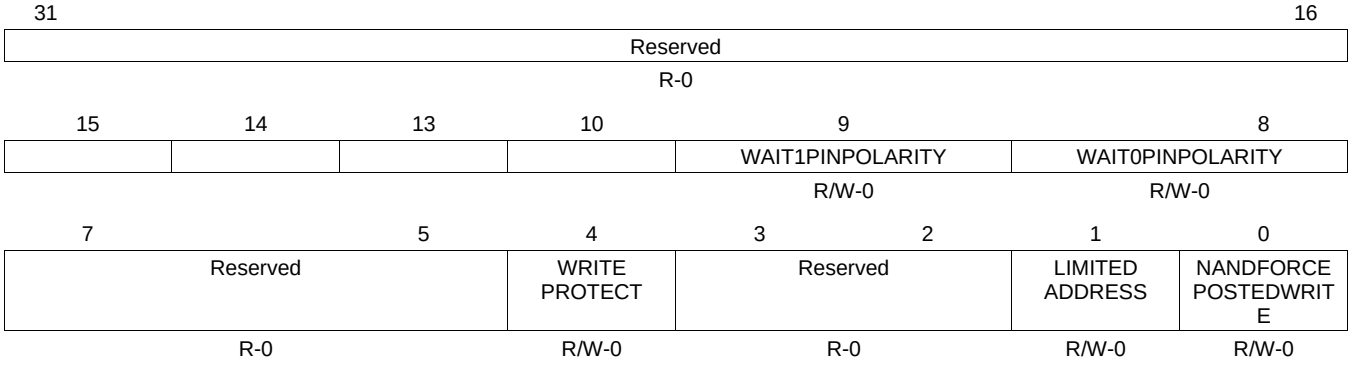
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Reserved | 0 | Reserved |
| 10-8 | ILLEGALMCMD | 0-7h | System Command of the transaction that caused the error |
| 7-5 | Reserved | 0 | Reserved |
| 4 | ERRORNOTSUPPADD | | Not supported Address error |
| | | 0 | No error occurs |
| | | 1 | The error is due to a non supported Address |
| 3 | ERRORNOTSUPPMCMD | | Not supported Command error |
| | | 0 | No error occurs |
| | | 1 | The error is due to a non supported Command |
| 2 | ERRORTIMEOUT | | Time-out error |
| | | 0 | No error occurs |
| | | 1 | The error is due to a time out |
| 1 | Reserved | 0 | Reserved |
| 0 | ERRORVALID | | Error validity status - Must be explicitly cleared with a write 1 transaction |
| | | 0 | All error fields no longer valid |
| | | 1 | Error detected and logged in the other error fields |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.9 GPMC_CONFIG

The configuration register allows global configuration of the GPMC.

**Figure 11-59. GPMC_CONFIG**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 11-63. GPMC_CONFIG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | | Reserved |
| 9 | WAIT1PINPOLARITY | | Selects the polarity of input pin WAIT1 |
| | | 0 | WAIT1 active low |
| | | 1 | WAIT1 active high |
| 8 | WAIT0PINPOLARITY | | Selects the polarity of input pin WAIT0 |
| | | 0 | WAIT0 active low |
| | | 1 | WAIT0 active high |
| 7-5 | Reserved | | Reserved |
| 4 | WRITEPROTECT | | Controls the $\overline{WP}$ output pin level |
| | | 0 | $\overline{WP}$ output pin is low |
| | | 1 | $\overline{WP}$ output pin is high |
| 1 | LIMITEDADDRESS | | Limited Address device support. Enabling this function forces A[27:11] high on the GPMC I/O. Thus only devices with 2 kbytes of addressing space can be used. |
| | | 0 | No effect (normal addressing) |
| | | 1 | GPMC_A[27:11] are driven high and not modified during memory accesses |
| 0 | NANDFORCEPOSTEDWRITE | | Enables the Force Posted Write feature to NAND Cmd/Add/Data location |
| | | 0 | Disables Force Posted Write |
| | | 1 | Enables Force Posted Write |



Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.10 GPMC_STATUS

The status register provides global status bits of the GPMC.

**Figure 11-60. GPMC_STATUS**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 |
|---|---|---|---|
| Reserved | | WAIT1STATUS | WAIT0STATUS |
| R-0 | | R/W-0 | R/W-0 |

| 7 | 1 | 0 |
|---|---|---|
| Reserved | | EMPTYWRITEBUFFERSTATUS |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-64. GPMC_STATUS Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9 | WAIT1STATUS | | Is a copy of input pin WAIT1. (Reset value is WAIT1 input pin sampled at IC reset) |
| | | 0 | WAIT1 asserted (inactive state) |
| | | 1 | WAIT1 de-asserted |
| 8 | WAIT0STATUS | | Is a copy of input pin WAIT0. (Reset value is WAIT0 input pin sampled at IC reset) |
| | | 0 | WAIT0 asserted (inactive state) |
| | | 1 | WAIT0 de-asserted |
| 7-1 | Reserved | 0 | Reserved |
| 0 | EMPTYWRITEBUFFERSTATUS | | Stores the empty status of the write buffer |
| | | 0 | Write Buffer is not empty |
| | | 1 | Write Buffer is empty |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.11 GPMC_CONFIG1_i

The configuration 1 register sets signal control parameters per chip select.

**Figure 11-61. GPMC_CONFIG1_i**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| WRAPBURST | READMULTIPLE | READTYPE | WRITEMULTIPLE | WRITETYPE | CLKACTIVATIONTIME | | ATTACHEDDEVICE PAGELENGTH |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | | R/W-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ATTACHEDDEVICE PAGELENGTH | WAITREADMONITOR ING | WAITWRITEMONITO RING | Reserved | WAITMONITORINGTIME | | WAITPINSELECT | |
| R/W-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | | R/W-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | DEVICESIZE | | DEVICETYPE | | MUXADDDATA | |
| R-0 | | R/W-0 | | R/W-0 | | R/W-0 | |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | TIMEPARA GRANULARITY | Reserved | | GPMCFCLKDIVIDER | |
| R-0 | | R/W-0 | R-0 | | R/W-0 | |

LEGEND: R = Read only; W1C = Write 1 to clear bit; *-n* = value after reset

**Table 11-65. GPMC_CONFIG1_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | WRAPBURST | | Enables the wrapping burst capability. Must be set if the attached device is configured in wrapping burst |
| | | 0 | Synchronous wrapping burst not supported |
| | | 1 | Synchronous wrapping burst supported |
| 30 | READMULTIPLE | | Selects the read single or multiple access |
| | | 0 | single access |
| | | 1 | multiple access (burst if synchonous, page if asynchronous) |
| 29 | READTYPE | | Selects the read mode operation |
| | | 0 | Read Asynchronous |
| | | 1 | Read Synchronous |
| 28 | WRITEMULTIPLE | | Selects the write single or multiple access |
| | | 0 | Single access |
| | | 1 | Multiple access (burst if synchronous, considered as single if asynchronous) |
| 27 | WRITETYPE | | Selects the write mode operation |
| | | 0 | Write Asynchronous |
| | | 1 | Write Synchronous |
| 26-25 | CLKACTIVATIONTIME | | Output GPMC.CLK activation time |
| | | 0 | First rising edge of GPMC_CLK at start access time |
| | | 1h | First rising edge of GPMC_CLK one GPMC_FCLK cycle after start access time |
| | | 2h | First rising edge of GPMC_CLK two GPMC_FCLK cycles after start access time |
| | | 3h | Reserved |
| 24-23 | ATTACHEDDEVICEPAGELENGTH | | Specifies the attached device page (burst) length (1 Word = Interface size) |
| | | 0 | 4 Words |
| | | 1h | 8 Words |
| | | 2h | 16 Words |
| | | 3h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-65. GPMC_CONFIG1_i Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 22 | WAITREADMONITORING | | Selects the Wait monitoring configuration for Read accesses (Reset value is BOOTWAITEN input pin sampled at IC reset) |
| | | 0 | WAIT pin is not monitored for read accesses |
| | | 1 | WAIT pin is monitored for read accesses |
| 21 | WAITWRITEMONITORING | | Selects the Wait monitoring configuration for Write accesses |
| | | 0 | WAIT pin is not monitored for write accesses |
| | | 1 | WAIT pin is monitored for write accesses |
| 20 | Reserved | 0 | Reserved |
| 19-18 | WAITMONITORINGTIME | | Selects input pin Wait monitoring time |
| | | 0 | WAIT pin is monitored with valid data |
| | | 1h | WAIT pin is monitored one GPMC_CLK cycle before valid data |
| | | 2h | WAIt pin is monitored two GPMC_CLK cycle before valid data |
| | | 3h | Reserved |
| 17-16 | WAITPINSELECT | | Selects the input WAIT pin for this chip select (Reset value is BOOTWAITSELECT input pin sampled at IC reset for CS0 and 0 for CS1-5) |
| | | 0 | WAIT input pin is WAIT0 |
| | | 1h | WAIT input pin is WAIT1 |
| | | 2h | Reserved |
| | | 3h | Reserved |
| 15-14 | Reserved | 0 | Reserved |
| 13-12 | DEVICESIZE | | Selects the device size attached (Reset value is BOOTDEVICESIZE input pin sampled at IC reset for $\overline{CS}$[0] and 01 for $\overline{CS}$[1-5]) |
| | | 0 | 8 bit |
| | | 1h | 16 bit |
| | | 2h | Reserved |
| | | 3h | Reserved |
| 11-10 | DEVICETYPE | | Selects the attached device type |
| | | 0 | NOR Flash like, asynchronous and synchronous devices |
| | | 1h | Reserved |
| | | 2h | NAND Flash like devices, stream mode |
| | | 3h | Reserved |
| 9-8 | MUXADDDATA | | Enables the Address and data multiplexed protocol (Reset value is CS0MUXDEVICE input pin sampled at IC reset for $\overline{CS}$[0] and 0 for $\overline{CS}$[1-5]) |
| | | 0 | Non-multiplexed attached device |
| | | 1h | AAD-multiplexed protocol device |
| | | 2h | Address and data multiplexed attached device |
| | | 3h | Reserved |
| 7-5 | Reserved | 0 | Reserved |
| 4 | TIMEPARAGRANULARITY | | Signals timing latencies scalar factor (Rd/WRCycleTime, RdAccessTime, PageBurstAccessTime, CSOnTime, CSRd/WrOffTime, ADVOnTime, ADVRd/WrOffTime, OEOnTime, OEOffTime, WEOnTime, WEOffTime, Cycle2CycleDelay, BusTurnAround, TimeOutStartValue) |
| | | 0 | ×1 latencies |
| | | 1 | ×2 latencies |
| 3-2 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-65. GPMC_CONFIG1_i Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1-0 | GPMCFCLKDIVIDER | | Divides the GPMC.FCLK clock |
| | | 0 | GPMC_CLK frequency = GPMC_FCLK frequency |
| | | 1h | GPMC_CLK frequency = GPMC_FCLK frequency/2 |
| | | 2h | GPMC_CLK frequency = GPMC_FCLK frequency/4 |
| | | 3h | GPMC_CLK frequency = GPMC_FCLK frequency/8 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.12 GPMC_CONFIG2_i

Chip-select signal timing parameter configuration.

**Figure 11-62. GPMC_CONFIG2_i**

| 31 | 21 | 20 | 16 |
|---|---|---|---|
| Reserved | | CSWROFFTIME | |
| R-0 | | R/W-0 | |

| 15 | 13 | 12 | 8 | 7 | 6 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | CSRDOFFTIME | | CSEXTRADELAY | Reserved | | CSONTIME | |
| R-0 | | R/W-0 | | R/W-0 | R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-66. GPMC_CONFIG2_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-21 | Reserved | 0 | Reserved |
| 20-16 | CSWROFFTIME | | CS# de-assertion time from start cycle time for write accesses |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 31 GPMC_FCLK cycles |
| 15-13 | Reserved | 0 | Reserved |
| 12-8 | CSRDOFFTIME | | CS# de-assertion time from start cycle time for read accesses |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 31 GPMC_FCLK cycles |
| 7 | CSEXTRADELAY | | CS# Add Extra Half GPMC.FCLK cycle |
| | | 0 | CS i Timing control signal is not delayed |
| | | 1 | CS i Timing control signal is delayed of half GPMC_FCLK clock cycle |
| 6-4 | Reserved | 0 | Reserved |
| 3-0 | CSONTIME | | CS# assertion time from start cycle time |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 15 GPMC_FCLK cycles |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.13 GPMC_CONFIG3_i

ADV# signal timing parameter configuration.

**Figure 11-63. GPMC_CONFIG3_i**

| 31 | 30 | 28 | 27 | 26 | 24 |
|---|---|---|---|---|---|
| Reserved | ADVAADMUXWROFFTIME | | Reserved | ADVAADMUXRDOFFTIME | |
| R-0 | R/W-0 | | R-0 | R/W-0 | |

| 23 | 21 | 20 | 16 |
|---|---|---|---|
| Reserved | | ADVWROFFTIME | |
| R/W-0 | | R/W-0 | |

| 15 | 13 | 12 | 8 |
|---|---|---|---|
| Reserved | | ADVRDOFFTIME | |
| R-0 | | R/W-0 | |

| 7 | 6 | 4 | 3 | 0 |
|---|---|---|---|---|
| ADVEXTRA DELAY | ADVAADMUXONTIME | | ADVONTIME | |
| R/W-0 | R/W-0 | | R/W-0 | |

LEGEND: R = Read only; W1C = Write 1 to clear bit; -*n* = value after reset

**Table 11-67. GPMC_CONFIG3_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved |
| 30-28 | ADVAADMUXWROFFTIME | | ADV# de-assertion for first address phase when using the AAD-Mux protocol |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 7h | 7 GPMC_FCLK cycles |
| 27 | Reserved | 0 | Reserved |
| 26-24 | ADVAADMUXRDOFFTIME | | ADV# assertion for first address phase when using the AAD-Mux protocol |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 7h | 7 GPMC_FCLK cycles |
| 23-21 | Reserved | 0 | Reserved |
| 20-16 | ADVWROFFTIME | | ADV# de-assertion time from start cycle time for write accesses |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 31 GPMC_FCLK cycles |
| 15-13 | Reserved | 0 | Reserved |
| 12-8 | ADVRDOFFTIME | | ADV# de-assertion time from start cycle time for read accesses |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 31 GPMC_FCLK cycles |
| 7 | ADVEXTRADELAY | | ADV# Add Extra Half GPMC.FCLK cycle |
| | | 0 | $\overline{\text{ADV}}$ Timing control signal is not delayed |
| | | 1 | $\overline{\text{ADV}}$ Timing control signal is delayed of half GPMC_FCLK clock cycle |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-67. GPMC_CONFIG3_i Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 6-4 | ADVAADMUXONTIME | | ADV# assertion for first address phase when using the AAD-Multiplexed protocol |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 7h | 7 GPMC_FCLK cycles |
| 3-0 | ADVONTIME | | ADV# assertion time from start cycle time |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | Fh | 15 GPMC_FCLK cycles |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.14 GPMC_CONFIG4_i

WE# and OE# signals timing parameter configuration.

**Figure 11-64. GPMC_CONFIG4_i**

| 31 | 29 | 28 | 24 |
|---|---|---|---|
| Reserved | | WEOFFTIME | |
| R-0 | | R/W-0 | |

| 23 | 22 | 20 | 19 | 16 |
|---|---|---|---|---|
| WEEXTRADELAY | Reserved | | WEONTIME | |
| R/W-0 | R-0 | | R/W-0 | |

| 15 | 13 | 12 | 8 |
|---|---|---|---|
| OEAADMUXOFFTIME | | OEOFFTIME | |
| R/W-0 | | R/W-0 | |

| 7 | 6 | 4 | 3 | 0 |
|---|---|---|---|---|
| OEEXTRA DELAY | OEAADMUXONTIME | | OEONTIME | |
| R/W-0 | R/W-0 | | R/W-0 | |

LEGEND: R = Read only; W1C = Write 1 to clear bit; -*n* = value after reset

**Table 11-68. GPMC_CONFIG4_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-29 | Reserved | 0 | Reserved |
| 28-24 | WEOFFTIME | | WE# de-assertion time from start cycle time |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 31 GPMC_FCLK cycles |
| 23 | WEEXTRADELAY | | WE# Add Extra Half GPMC.FCLK cycle |
| | | 0 | $\overline{WE}$ Timing control signal is not delayed |
| | | 1 | $\overline{WE}$ Timing control signal is delayed of half GPMC_FCLK clock cycle |
| 22-20 | Reserved | 0 | Reserved |
| 19-16 | WEONTIME | | WE# assertion time from start cycle time |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | Fh | 15 GPMC_FCLK cycles |
| 15-13 | OEAADMUXOFFTIME | | OE# de-assertion time for the first address phase in an AAD-Multiplexed access |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 7h | 7 GPMC_FCLK cycles |
| 12-8 | OEOFFTIME | | OE# de-assertion time from start cycle time |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 31 GPMC_FCLK cycles |
| 7 | OEEXTRADELAY | | OE# Add Extra Half GPMC.FCLK cycle |
| | | 0 | $\overline{OE}$ Timing control signal is not delayed |
| | | 1 | $\overline{OE}$ Timing control signal is delayed of half GPMC_FCLK clock cycle |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-68. GPMC_CONFIG4_i Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 6-4 | OEAADMUXONTIME | | OE# assertion time for the first address phase in an AAD-Multiplexed access |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 7h | 7 GPMC_FCLK cycles |
| 3-0 | OEONTIME | | OE# assertion time from start cycle time |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | Fh | 15 GPMC_FCLK cycles |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.15 GPMC_CONFIG5_i

RdAccessTime and CycleTime timing parameters configuration.

**Figure 11-65. GPMC_CONFIG5_i**

| 31 28 | 27 24 | 23 21 | 20 16 |
|---|---|---|---|
| Reserved | PAGEBURSTACCESSTIME | Reserved | RDACCESSTIME |
| R-0 | R/W-0 | R-0 | R/W-0 |

| 15 13 | 12 8 | 7 5 | 4 0 |
|---|---|---|---|
| Reserved | WRCYCLETIME | Reserved | RDCYCLETIME |
| R-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-69. GPMC_CONFIG5_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | Reserved | 0 | Reserved |
| 27-24 | PAGEBURSTACCESSTIME | | Delay between successive words in a multiple access |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | Fh | 15 GPMC_FCLK cycles |
| 23-21 | Reserved | 0 | Reserved |
| 20-16 | RDACCESSTIME | | Delay between start cycle time and first data valid |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 31 GPMC_FCLK cycles |
| 15-13 | Reserved | 0 | Reserved |
| 12-8 | WRCYCLETIME | | Total write cycle time |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 31 GPMC_FCLK cycles |
| 7-5 | Reserved | 0 | Reserved |
| 4-0 | RDCYCLETIME | | Total read cycle time |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 31 GPMC_FCLK cycles |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.16 GPMC_CONFIG6_i

WrAccessTime, WrDataOnADmuxBus, Cycle2Cycle, and BusTurnAround parameters configuration

**Figure 11-66. GPMC_CONFIG6_i**

| 31 29 | 28 24 | 23 20 | 19 16 |
|---|---|---|---|
| Reserved | WRACCESSTIME | Reserved | WRDATAONADMUXBUS |
| R-0 | R/W-0 | R-0 | R/W-0 |

| 15 12 | 11 8 |
|---|---|
| Reserved | CYCLE2CYCLEDELAY |
| R-0 | R/W-0 |

| 7 | 6 | 5 4 | 3 0 |
|---|---|---|---|
| CYCLE2CYCLE SAMECSEN | CYCLE2CYCLE DIFFCSEN | Reserved | BUSTURNAROUND |
| R/W-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-70. GPMC_CONFIG6_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-29 | Reserved | 0 | Reserved |
| 28-24 | WRACCESSTIME | | Delay from StartAccessTime to the GPMC.FCLK rising edge corresponding the the GPMC.CLK rising edge used by the attached memory for the first data capture |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 1Fh | 31 GPMC_FCLK cycles |
| 23-20 | Reserved | 0 | Reserved |
| 19-16 | WRDATAONADMUXBUS | 0-Fh | Specifies on which GPMC.FCLK rising edge the first data of the synchronous burst write is driven in the add/data multiplexed bus |
| 15-12 | Reserved | 0 | Reserved |
| 11-8 | CYCLE2CYCLEDELAY | | Chip select high pulse delay between two successive accesses |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | Fh | 15 GPMC_FCLK cycles |
| 7 | CYCLE2CYCLESAMECSEN | | Add Cycle2CycleDelay between two successive accesses to the same chip-select (any access type) |
| | | 0 | No delay between the two accesses |
| | | 1 | Add CYCLE2CYCLEDELAY |
| 6 | CYCLE2CYCLEDIFFCSEN | | Add Cycle2CycleDelay between two successive accesses to a different chip-select (any access type) |
| | | 0 | No delay between the two accesses |
| | | 1 | Add CYCLE2CYCLEDELAY |
| 5-4 | Reserved | 0 | Reserved |
| 3-0 | BUSTURNAROUND | | Bus turn around latency between two successive accesses to the same chip-select (read to write) or to a different chip-select (read to read and read to write) |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | Fh | 15 GPMC_FCLK cycles |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.17 GPMC_CONFIG7_i

Chip-select address mapping configuration.

**Figure 11-67. GPMC_CONFIG7_i**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 12 | 11 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | MASKADDRESS | | Rsvd | CSVALID | BASEADDRESS | |
| R-0 | | R/W-0 | | R-0 | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-71. GPMC_CONFIG7_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | Reserved | 0 | Reserved |
| 11-8 | MASKADDRESS | | Chip-select mask address. Values not listed must be avoided as they create holes in the chip-select address space. |
| | | 0 | Chip-select size of 256 Mbytes |
| | | 8h | Chip-select size of 128 Mbytes |
| | | Ch | Chip-select size of 64 Mbytes |
| | | Eh | Chip-select size of 32 Mbytes |
| | | Fh | Chip-select size of 16 Mbytes |
| 7 | Reserved | 0 | Reserved |
| 6 | CSVALID | | Chip-select enable (reset value is 1 for $\overline{CS}$[0] and 0 for $\overline{CS}$[1-5]). |
| | | 0 | $\overline{CS}$ disabled |
| | | 1 | $\overline{CS}$ enabled |
| 5-0 | BASEADDRESS | 0-3Fh | Chip-select base address. |
| | | | CSi base address where i = 0 to 3 (16 Mbytes minimum granularity). Bits [5-0] corresponds to A29, A28, A27, A26, A25, and A24. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.18 GPMC_NAND_COMMAND_i

This register is not a true register, just an address location.

**Figure 11-68. GPMC_NAND_COMMAND_i**

| 31 | 0 |
|---|---|
| GPMC_NAND_COMMAND_i | |
| W-0 | |

LEGEND: W = Write only; -n = value after reset

**Table 11-72. GPMC_NAND_COMMAND_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | GPMC_NAND_COMMAND_i | 0-FFFF FFFFh | Writing data at the GPMC_NAND_COMMAND_i location places the data as the NAND command value on the bus, using a regular asynchronous write access. |

### 11.5.19 GPMC_NAND_ADDRESS_i

This register is not a true register, just an address location.

**Figure 11-69. GPMC_NAND_ADDRESS_i**

| 31 | 0 |
|---|---|
| GPMC_NAND_ADDRESS_i | |
| W-0 | |

LEGEND: W = Write only; -n = value after reset

**Table 11-73. GPMC_NAND_ADDRESS_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | GPMC_NAND_ADDRESS_i | 0-FFFF FFFFh | Writing data at the GPMC_NAND_ADDRESS_i location places the data as the NAND partial address value on the bus, using a regular asynchronous write access. |

### 11.5.20 GPMC_NAND_DATA_i

This register is not a true register, just an address location.

**Figure 11-70. GPMC_NAND_DATA_i**

| 31 | 0 |
|---|---|
| GPMC_NAND_DATA_i | |
| R/W | |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -n = value after reset

**Table 11-74. GPMC_NAND_DATA_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | GPMC_NAND_DATA_i | 0-FFFF FFFFh | Reading data from the GPMC_NAND_DATA_i location or from any location in the associated chip-select memory region activates an asynchronous read access. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.21 GPMC_PREFETCH_CONFIG1

**Figure 11-71. GPMC_PREFETCH_CONFIG1**

| 31 | 30 | 28 | 27 | 26 | 24 |
|---|---|---|---|---|---|
| Reserved | CYCLEOPTIMIZATION | | ENABLEOPTIMIZED ACCESS | ENGINECSSELECTOR | |
| R-0 | R/W-0 | | R/W-0 | R/W-0 | |

| 23 | 22 | 20 | 19 | 16 |
|---|---|---|---|---|
| PFPWENROUNDROBIN | Reserved | | PFPWWEIGHTEDPRIO | |
| R/W-0 | R-0 | | R/W-0 | |

| 15 | 14 | 8 |
|---|---|---|
| Reserved | FIFOTHRESHOLD | |
| R-0 | R/W-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ENABLEENGINE | Reserved | WAITPINSELECTOR | | SYNCHROMODE | DMAMODE | Reserved | ACCESSMODE |
| R/W-0 | R-0 | R/W-0 | | R/W-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 11-75. GPMC_PREFETCH_CONFIG1 Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved |
| 30-28 | CYCLEOPTIMIZATION | | Define the number of GPMC.FCLK cycles to be substracted from RdCycleTime, WrCycleTime, RdAccessTime, CSRdOffTime, CSWrOffTime, ADVRdOffTime, ADVWrOffTime, OEOffTime, WEOffTime |
| | | 0 | 0 GPMC_FCLK cycle |
| | | 1h | 1 GPMC_FCLK cycle |
| | | ⋮ | ⋮ |
| | | 7h | 7 GPMC_FCLK cycles |
| 27 | ENABLEOPTIMIZEDACCESS | | Enables access cycle optimization |
| | | 0 | Access cycle optimization is disabled |
| | | 1 | Access cycle optimization is enabled |
| 26-24 | ENGINECSSELECTOR | | Selects the $\overline{CS}$ where Prefetch Postwrite engine is active |
| | | 0 | $\overline{CS0}$ |
| | | 1h | $\overline{CS1}$ |
| | | 2h | $\overline{CS2}$ |
| | | 3h | $\overline{CS3}$ |
| | | 4h | $\overline{CS4}$ |
| | | 5h | $\overline{CS5}$ |
| | | 6h | $\overline{CS6}$ |
| | | 7h | $\overline{CS7}$ |
| 23 | PFPWENROUNDROBIN | | Enables the PFPW RoundRobin arbitration |
| | | 0 | Prefetch Postwrite engine round robin arbitration is disabled |
| | | 1 | Prefetch Postwrite engine round robin arbitration is enabled |
| 22-20 | Reserved | 0 | Reserved |
| 19-16 | PFPWWEIGHTEDPRIO | | When an arbitration occurs between a direct memory access and a PFPW engine access, the direct memory access is always serviced. If the PFPWEnRoundRobin is enabled, |
| | | 0 | The next access is granted to the PFPW engine |
| | | 1h | The two next accesses are granted to the PFPW engine |
| | | ⋮ | ⋮ |
| | | Fh | The 16 next accesses are granted to the PFPW engine |
| 15 | Reserved | 0 | Reserved |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-75. GPMC_PREFETCH_CONFIG1 Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 14-8 | FIFOTHRESHOLD | | Selects the maximum number of bytes read from the FIFO or written to the FIFO by the host on a DMA or interrupt request |
| | | 0 | 0 byte |
| | | 1h | 1 byte |
| | | ⋮ | ⋮ |
| | | 40h | 64 bytes |
| 7 | ENABLEENGINE | | Enables the Prefetch Postwite engine |
| | | 0 | Prefetch Postwrite engine is disabled |
| | | 1 | Prefetch Postwrite engine is enabled |
| 6 | Reserved | 0 | Reserved |
| 5-4 | WAITPINSELECTOR | | Select which WAIT pin edge detector should start the engine in synchronized mode |
| | | 0 | Selects Wait0EdgeDetection |
| | | 1h | Selects Wait1EdgeDetection |
| | | 2h | Reserved |
| | | 3h | Reserved |
| 3 | SYNCHROMODE | | Selects when the engine starts the access to CS |
| | | 0 | Engine starts the access to CS as soon as STARTENGINE is set |
| | | 1 | Engine starts the access to CS as soon as STARTENGINE is set AND wait to non wait edge detection on the selected wait pin |
| 2 | DMAMODE | | Selects interrupt synchronization or DMA request synchronization |
| | | 0 | Interrupt synchronization is enabled. Only interrupt line will be activated on FIFO threshold crossing. |
| | | 1 | DMA request synchronization is enabled. A DMA request protocol is used. |
| 1 | Reserved | 0 | Reserved |
| 0 | ACCESSMODE | | Selects pre-fetch read or write posting accesses |
| | | 0 | Prefetch read mode |
| | | 1 | Write-posting mode |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.22 GPMC_PREFETCH_CONFIG2

**Figure 11-72. GPMC_PREFETCH_CONFIG2**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 0 |
|---|---|---|---|
| Reserved | | TRANSFERCOUNT | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-76. GPMC_PREFETCH_CONFIG2 Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-14 | Reserved | 0 | Reserved |
| 13-0 | TRANSFERCOUNT | | Selects the number of bytes to be read or written by the engine to the selected $\overline{CS}$ |
| | | 0 | 0 byte |
| | | 1h | 1 byte |
| | | ⋮ | ⋮ |
| | | 2000h | 8 Kbytes |

### 11.5.23 GPMC_PREFETCH_CONTROL

**Figure 11-73. GPMC_PREFETCH_CONTROL**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | STARTENGINE |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-77. GPMC_PREFETCH_CONTROL Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | STARTENGINE | | Resets the FIFO pointer and starts the engine |
| | | R0 | Engine is stopped |
| | | W0 | Stops the engine |
| | | R1 | Engine is running |
| | | W1 | Resets the FIFO pointer to 0 in prefetch mode and 40h in postwrite mode and starts the engine |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.24 GPMC_PREFETCH_STATUS

**Figure 11-74. GPMC_PREFETCH_STATUS**

| 31 | 30 24 | 23 17 | 16 |
|---|---|---|---|
| Rsvd | FIFOPOINTER | Reserved | FIFOTHRESHOLDSTATUS |
| R-0 | R-0 | R-0 | R-0 |

| 15 | 14 13 | 0 |
|---|---|---|
| Reserved | COUNTVALUE | |
| R-0 | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-78. GPMC_PREFETCH_STATUS Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved |
| 30-24 | FIFOPOINTER | 0 | 0 byte available to be read or 0 free empty place to be written |
| | | ⋮ | ⋮ |
| | | 40h | 64 bytes available to be read or 64 empty places to be written |
| 23-17 | Reserved | 0 | Reserved |
| 16 | FIFOTHRESHOLDSTATUS | | Set when FIFOPointer exceeds FIFOThreshold value |
| | | 0 | FIFOPointer smaller or equal to FIFOThreshold. Writing to this bit has no effect |
| | | 1 | FIFOPointer greater than FIFOThreshold. Writing to this bit has no effect |
| 15-14 | Reserved | 0 | Reserved |
| 13-0 | COUNTVALUE | | Number of remaining bytes to be read or to be written by the engine according to the TransferCount value |
| | | 0 | 0 byte remaining to be read or to be written |
| | | 1h | 1 byte remaining to be read or to be writte |
| | | ⋮ | ⋮ |
| | | 2000h | 8 Kbytes remaining to be read or to be written |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.25 GPMC_ECC_CONFIG

**Figure 11-75. GPMC_ECC_CONFIG**

| 31 | | | | | | | | | | | | | | | | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | ECCALGORITHM |
| R-0 | | | | | | | | | | | | | | | | | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 8 |
|---|---|---|---|---|---|
| Reserved | | ECCBCHTSEL | | ECCWRAPMODE | |
| R-0 | | R/W-0 | | R/W-0 | |

| 7 | 6 | 4 | 3 | 1 | 0 |
|---|---|---|---|---|---|
| ECC16B | ECCTOPSECTOR | | ECCCS | | ECCENABLE |
| R/W-0 | R/W-0 | | R/W-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-79. GPMC_ECC_CONFIG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved |
| 16 | ECCALGORITHM | | ECC algorithm used |
| | | 0 | Hamming code |
| | | 1 | BCH code |
| 15-14 | Reserved | 0 | Reserved |
| 13-12 | ECCBCHTSEL | | Error correction capability used for BCH |
| | | 0 | Up to 4 bits error correction (t = 4) |
| | | 1h | Up to 8 bits error correction (t = 8) |
| | | 2h | Up to 16 bits error correction (t = 16) |
| | | 3h | Reserved |
| 11-8 | ECCWRAPMODE | 0-Fh | Spare area organization definition for the BCH algorithm. See the BCH syndrome/parity calculator module functional specification for more details |
| 7 | ECC16B | | Selects an ECC calculated on 16 columns |
| | | 0 | ECC calculated on 8 columns |
| | | 1 | ECC calculated on 16 columns |
| 6-4 | ECCTOPSECTOR | | Number of sectors to process with the BCH algorithm |
| | | 0 | 1 sector (512bytes page) |
| | | 1h | 2 sectors |
| | | 3h | 4 sectors (2kB page) |
| | | ⋮ | ⋮ |
| | | 7h | 8 sectors (4kB page) |
| 3-1 | ECCCS | | Selects the Chip-select where ECC is computed |
| | | 0 | Chip-select 0 |
| | | 1h | Chip-select 1 |
| | | 2h | Chip-select 2 |
| | | 3h | Chip-select 3 |
| | | 4h | Chip-select 4 |
| | | 5h | Chip-select 5 |
| | | 6h-7h | Reserved |
| 0 | ECCENABLE | | Enables the ECC feature |
| | | 0 | ECC disabled |
| | | 1 | ECC enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.26 GPMC_ECC_CONTROL

**Figure 11-76. GPMC_ECC_CONTROL**

| 31 | | | | | | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |
| 15 | | | | | | | 9 | 8 | 7 | | | 4 | 3 | | 0 |
| Reserved | | | | | | | | ECCCLEAR | Reserved | | | | ECCPOINTER | | |
| R-0 | | | | | | | | R/W-0 | R-0 | | | | R/W-0 | | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-80. GPMC_ECC_CONTROL Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved |
| 8 | ECCCLEAR | | Clear all ECC result registers |
| | | R0 | All reads return 0 |
| | | W0 | Ignored |
| | | W1 | Clears all ECC result registers |
| 7-4 | Reserved | 0 | Reserved |
| 3-0 | ECCPOINTER | 0-Fh | Selects ECC result register (Reads to this field give the dynamic position of the ECC pointer - Writes to this field select the ECC result register where the first ECC computation will be stored). Writing values not listed disables the ECC engine (ECCEnable bit of GPMC_ECC_CONFIG cleared to 0). |
| | | 0 | Writing 0 disables the ECC engine (ECCENABLE bit of GPMC_ECC_CONFIG cleared to 0) |
| | | 1h | ECC result register 1 selected |
| | | 2h | ECC result register 2 selected |
| | | 3h | ECC result register 3 selected |
| | | 4h | ECC result register 4 selected |
| | | 5h | ECC result register 5 selected |
| | | 6h | ECC result register 6 selected |
| | | 7h | ECC result register 7 selected |
| | | 8h | ECC result register 8 selected |
| | | 9h | ECC result register 9 selected |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.27 GPMC_ECC_SIZE_CONFIG

**Figure 11-77. GPMC_ECC_SIZE_CONFIG**

| 31 | 30 | 29 | 22 | 21 | 20 | 19 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | ECCSIZE1 | | Reserved | | ECCSIZE0 | |
| R-0 | | R/W-0 | | R-0 | | R/W-0 | |

| 15 | 12 | 11 | 9 | 8 |
|---|---|---|---|---|
| ECCSIZE0 | | Reserved | | ECC9RESULTSIZE |
| R/W-0 | | R-0 | | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ECC8RESULTSIZE | ECC7RESULTSIZE | ECC6RESULTSIZE | ECC5RESULTSIZE | ECC4RESULTSIZE | ECC3RESULTSIZE | ECC2RESULTSIZE | ECC1RESULTSIZE |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-81. GPMC_ECC_SIZE_CONFIG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Reserved |
| 29-22 | ECCSIZE1 | | Defines ECC size 1 |
| | | 0 | 2 Bytes |
| | | 1h | 4 Bytes |
| | | 2h | 6 Bytes |
| | | 3h | 8 Bytes |
| | | ⋮ | ⋮ |
| | | FFh | 512 Bytes |
| 21-20 | Reserved | 0 | Reserved |
| 19-12 | ECCSIZE0 | | Defines ECC size 0 |
| | | 0 | 2 Bytes |
| | | 1h | 4 Bytes |
| | | 2h | 6 Bytes |
| | | 3h | 8 Bytes |
| | | ⋮ | ⋮ |
| | | FFh | 512 Bytes |
| 11-9 | Reserved | 0 | Reserved |
| 8 | ECC9RESULTSIZE | | Selects ECC size for ECC 9 result register |
| | | 0 | ECCSIZE0 selected |
| | | 1 | ECCSIZE1 selected |
| 7 | ECC8RESULTSIZE | | Selects ECC size for ECC 8 result register |
| | | 0 | ECCSIZE0 selected |
| | | 1 | ECCSIZE1 selected |
| 6 | ECC7RESULTSIZE | | Selects ECC size for ECC 7 result register |
| | | 0 | ECCSIZE0 selected |
| | | 1 | ECCSIZE1 selected |
| 5 | ECC6RESULTSIZE | | Selects ECC size for ECC 6 result register |
| | | 0 | ECCSIZE0 selected |
| | | 1 | ECCSIZE1 selected |
| 4 | ECC5RESULTSIZE | | Selects ECC size for ECC 5 result register |
| | | 0 | ECCSIZE0 selected |
| | | 1 | ECCSIZE1 selected |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 11-81. GPMC_ECC_SIZE_CONFIG Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 3 | ECC4RESULTSIZE | | Selects ECC size for ECC 4 result register |
| | | 0 | ECCSIZE0 selected |
| | | 1 | ECCSIZE1 selected |
| 2 | ECC3RESULTSIZE | | Selects ECC size for ECC 3 result register |
| | | 0 | ECCSIZE0 selected |
| | | 1 | ECCSIZE1 selected |
| 1 | ECC2RESULTSIZE | | Selects ECC size for ECC 2 result register |
| | | 0 | ECCSIZE0 selected |
| | | 1 | ECCSIZE1 selected |
| 0 | ECC1RESULTSIZE | | Selects ECC size for ECC 1 result register |
| | | 0 | ECCSIZE0 selected |
| | | 1 | ECCSIZE1 selected |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.28 GPMC_ECCj_RESULT

**Figure 11-78. GPMC_ECCj_RESULT**

| 31 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | P2048O | P1024O | P512O | P256O | P128O | P64O | P32O | P16O | P8O | P4O | P2O | P1O |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

| 15 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | P2048E | P1024E | P512E | P256E | P128E | P64E | P32E | P16E | P8E | P4E | P2E | P1E |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-82. GPMC_ECCj_RESULT Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | Reserved | 0 | Reserved |
| 27 | P2048O | 0-1 | Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes |
| 26 | P1024O | 0-1 | Odd Row Parity bit 1024 |
| 25 | P512O | 0-1 | Odd Row Parity bit 512 |
| 24 | P256O | 0-1 | Odd Row Parity bit 256 |
| 23 | P128O | 0-1 | Odd Row Parity bit 128 |
| 22 | P64O | 0-1 | Odd Row Parity bit 64 |
| 21 | P32O | 0-1 | Odd Row Parity bit 32 |
| 20 | P16O | 0-1 | Odd Row Parity bit 16 |
| 19 | P8O | 0-1 | Odd Row Parity bit 8 |
| 18 | P4O | 0-1 | Odd Column Parity bit 4 |
| 17 | P2O | 0-1 | Odd Column Parity bit 2 |
| 16 | P1O | 0-1 | Odd Column Parity bit 1 |
| 15-12 | Reserved | 0 | Reserved |
| 11 | P2048E | 0-1 | Even Row Parity bit 2048, only used for ECC computed on 512 Bytes |
| 10 | P1024E | 0-1 | Even Row Parity bit 1024 |
| 9 | P512E | 0-1 | Even Row Parity bit 512 |
| 8 | P256E | 0-1 | Even Row Parity bit 256 |
| 7 | P128E | 0-1 | Even Row Parity bit 128 |
| 6 | P64E | 0-1 | Even Row Parity bit 64 |
| 5 | P32E | 0-1 | Even Row Parity bit 32 |
| 4 | P16E | 0-1 | Even Row Parity bit 16 |
| 3 | P8E | 0-1 | Even Row Parity bit 8 |
| 2 | P4E | 0-1 | Even Column Parity bit 4 |
| 1 | P2E | 0-1 | Even Column Parity bit 2 |
| 0 | P1E | 0-1 | Even Column Parity bit 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.29 GPMC_BCH_RESULT0_i

**Figure 11-79. GPMC_BCH_RESULT0_i**

| 31 | 0 |
|---|---|
| BCH_RESULT0_i | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-83. GPMC_BCH_RESULT0_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | BCH_RESULT0_i | 0-FFFF FFFFh | BCH ECC result, bits 0 to 31 |

### 11.5.30 GPMC_BCH_RESULT1_i

**Figure 11-80. GPMC_BCH_RESULT1_i**

| 31 | 0 |
|---|---|
| BCH_RESULT1_i | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-84. GPMC_BCH_RESULT1_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | BCH_RESULT1_i | 0-FFFF FFFFh | BCH ECC result, bits 32 to 63 |

### 11.5.31 GPMC_BCH_RESULT2_i

**Figure 11-81. GPMC_BCH_RESULT2_i**

| 31 | 0 |
|---|---|
| BCH_RESULT2_i | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-85. GPMC_BCH_RESULT2_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | BCH_RESULT2_i | 0-FFFF FFFFh | BCH ECC result, bits 64 to 95 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.32 GPMC_BCH_RESULT3_i

**Figure 11-82. GPMC_BCH_RESULT3_i**

| 31 | 0 |
|---|---|
| BCH_RESULT3_i | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-86. GPMC_BCH_RESULT3_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | BCH_RESULT3_i | 0-FFFF FFFFh | BCH ECC result, bits 96 to 127 |

### 11.5.33 GPMC_BCH_RESULT4_i

**Figure 11-83. GPMC_BCH_RESULT4_i**

| 31 | 0 |
|---|---|
| BCH_RESULT4_i | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-87. GPMC_BCH_RESULT4_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | BCH_RESULT4_i | 0-FFFF FFFFh | BCH ECC result, bits 128 to 159 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 11.5.34 GPMC_BCH_RESULT5_i

**Figure 11-84. GPMC_BCH_RESULT5_i**

| 31 | 0 |
|---|---|
| BCH_RESULT5_i | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-88. GPMC_BCH_RESULT5_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | BCH_RESULT5_i | 0-FFFF FFFFh | BCH ECC result, bits 160 to 191 |

### 11.5.35 GPMC_BCH_RESULT6_i

**Figure 11-85. GPMC_BCH_RESULT6_i**

| 31 | 0 |
|---|---|
| BCH_RESULT6_i | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-89. GPMC_BCH_RESULT6_i Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | BCH_RESULT6_i | 0-FFFF FFFFh | BCH ECC result, bits 192 to 207 |

### 11.5.36 GPMC_BCH_SWDATA

This register is used to directly pass data to the BCH ECC calculator without accessing the actual NAND flash interface.

**Figure 11-86. GPMC_BCH_SWDATA**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 0 |
|---|---|
| BCH_DATA | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 11-90. GPMC_BCH_SWDATA Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15-0 | BCH_DATA | 0-FFFFh | Data to be included in the BCH calculation. Only bits 0 to 7 are taken into account, if the calculator is configured to use 8 bits data (GPMC_ECC_CONFIG[7] ECC16B = 0). |


Copyright © 2011–2016, Texas Instruments Incorporated

# High-Definition Video Processing Subsystem (HDVPSS)

**Topic** **Page**

**12.1 Description of the Subsystem ..... 1777**
**12.2 Internal Modules ..... 1808**
**12.3 Registers ..... 2061**



Copyright © 2011–2016, Texas Instruments Incorporated

## 12.1 Description of the Subsystem

### 12.1.1 Overview

The HDVPSS module includes video display and capture processing modules using the latest TI-developed algorithms, flexible compositing and blending engine, and a full range of external video interfaces in order to deliver high-quality video content to end devices.

### 12.1.2 Acronyms and Definitions

**Table 12-1. HDVPSS Acronyms**

| Acronym | Definition |
|---|---|
| HDVPSS | High Definition Video Processing Subsystem |
| DEI | De-Interlacer |
| DVO | Digital Video Output |
| GRPX | Graphics Pipeline |
| HDMI | High Definition Multimedia Interface |
| NF | Noise Filter |
| NTSC | National Television System Committee |
| PAL | Phase Alternating Line |
| SC | Scaler |
| SD | Standard Definition |
| SDK | Software Development Kit |
| COMP | Compositor |
| VENC | Video Encoder |
| VIP | Video Input Port |
| VPDMA | Video Port Direct Memory Access |
| FID | Field ID (Modules inside HDVPSS interpret FID = 0 as a TOP-field and FID = 1 as a BOTTOM field) |

**Table 12-2. HDVPSS Data Format**

| Name | DataFormat A | Arrangement | Tiler Support |
|---|---|---|---|
| 422I/422P | YUV422I_YUYV | Single Buffer: Y U Y V Y U Y V | No |
| 420T | YUV420SP_UV | Y Buffer: Y Y Y Y<br>UV buffer: U V U V | Y: 8-bit container<br>UV: 16-bit container |
| 422S | YUV422SP_UV | Y Buffer: Y Y Y Y<br>UV Buffer: U V U V | Y: 8-bit container<br>UV: 16-bit container |
| 422T | YUV422I_YUYV | Single Buffer: Y U Y V Y U Y V | Yes : 16-bit container |
| 444 | YUV444 | Single Buffer: Y U V | No |
| RGB565 | RGB565 | Single Buffer RGB, 16 bit/pixel | No |
| RGB24 | RGB | Single Buffer RGB, 24 bit/pixel | No |
| ARGB1555 | ARGB1555 | Single Buffer: ARGB, 16 bit/pixel | No |
| ARGB4444 | ARGB4444 | Single Buffer: ARGB, 16 bit/pixel | No |
| ARGB24 | ARGB24 | Single Buffer: ARGB, 24 bit/pixel | No |
| ARGB32 | ARGB32 | Single Buffer: ARGB, 32 bit/pixel | No |
| Bitmap | Bitmap | Single Buffer: Color Lookup Table (RGB input only) | No |

**NOTE:** For the 422T input port, the maximum input width is 960 pixels if the data format is YUV422I_YUYV.


Copyright © 2011–2016, Texas Instruments Incorporated

## 12.1.3 Features

### 12.1.3.1 Overall Features

Features of the HDVPSS module include:

- Supports HD (up to 1080p) and SD (NTSC/PAL) outputs simultaneously.
- Accepts the IVA HD Video Decoder output formats and adjust to several video formats. This includes (but not limited to) scan format conversion, scan rate conversion, aspect-ratio conversion, and frame size conversion.
- Handles both video and graphics efficiently to create high-quality user interfaces. This includes (but not limited to) deinterlacing, scaling, noise reduction, alpha blending, chroma keying, flicker filtering, and pixel format conversion.
- Generates secure video signal with proper content protection mechanisms, that is, HDCP and Macrovision/CGMS-a for digital and analog outputs, respectively.
- Supports VC-1 Range Mapping and Range Reduction on Primary, Auxiliary, and both Independent Transcode 420 Inputs.

### 12.1.3.2 Video Processing Features

- Two parallel video processing pipelines (main and aux) for concurrent HD and SD display are supported.
- The main video pipeline is used for processing video for the full size HD display. The main video pipeline employs motion-adaptive deinterlacer.
- Supported video input formats are 4:2:0 (aligned-chroma, planar, frame/field), 4:2:2 (planar, frame/field), and 444. The output formats of the HDVICP and video formats of captured external digital video data are supported. *Note:* HDVICP is not available on all devices. Refer to the device data manual for details.
- Scan format conversions (that is, interlaced to progressive and vice versa) is supported. Especially the interlaced to progressive conversion will employ a high quality motion-adaptive 3D deinterlacer to properly render both static and dynamic objects in scenes. In addition, sophisticated film-mode detection based on decoded frames (as well as the input bit-stream) is used for best rendering of film-based video.
- Scan rate conversion (up to 1080p 60Hz) is supported to handle video contents with various scan rates. It can be done via frame/field repeating/skipping. . 1080i inputs must result in 1080 lines of vertical resolution when displayed as a 1080i or P output.
- The output of the video processing will be sent to either compositor or external memory. When the output is sent to external memory, context switching between multiple inputs will be handled efficiently.
- Both the main and auxiliary video pipelines will include a write-back path to the external memory to support memory to memory scaling of video frames independently from the display output frame timing. Performance equivalent to scaling up to 16 QVGA (320x240) video windows per 1/60 sec are supported. (ADD video mosaic support).
- Color keying (transparency) is supported.

### 12.1.3.3 Graphics Features

- Three independently generated region-based graphics layers are supported.
- Each graphic layer supports full-screen resolution graphics.
- Each graphics layer can be displayed in either HD or SD output or both.
- Each of the graphics pipelines includes an up/down scaler optimized for graphics application. The scaler supports scaling ratios from 0.25x to 4x with 0.01 scaling step size.
- Scaling of individual regions within a graphics layer will be supported. A scaled region will not overlap with another region
- Supported graphics formats are:
  - 32-bit: ARGB8888
  - 24-bit: RGB888


Copyright © 2011–2016, Texas Instruments Incorporated

- 16-bit: ARGB1555, RGB565, ARGB4444
- Bitmap: 1, 2, 4, 8-bit CLUT table
- Global and pixel-level alpha blending value is supported (256 levels). For pixel-level blending, the alpha value can come from either source or CLUT table.
- Color keying (transparency)

#### 12.1.3.4 Cursor Features

- A dedicated hardware cursor is not supported. If a cursor function is required, it must be implemented in software.

#### 12.1.3.5 HD/SD Compositor Features

- Supports independently controlled compositors (HDMI/DVO1, DVO2, SD) .
- The HD compositors will support composition of video and graphics layers to provide full size video display, graphics overlay, video-in-graphics and picture-in-picture modes for HD video outputs.
- The SD compositor supports video display, graphics overlay, and video-in-graphics for SD video outputs..
- Each input layer will be given a display order priority which determines the display and blending orders.
- Each output supports independent layer visibility control.
- The compositor supports 256-level alpha blending of two overlapping layers.
- The compositor supports display outputs to HDMI/DVO1 and DVO2 at different scan rates or different pixel resolutions in addition to the separate SD output.

#### 12.1.3.6 Video Output and Encoding Features

- A single HDMI 1.3 compliant interface with HDCP to support 1080p (1080p 24, 30, and 60 are mandatory), 1080i, 720p, 480p, 480i. 640x480, 800x600, 1024x768, 1280x768, and 1920x1200.
- Two DVO interfaces that support up to 1080p at 60 MHz 8/16/24 YCbCr/RGB formats are included. Each DVO port supports both embedded and separated sync outputs. VBI data insertion on DVO outputs are not required.
- SD Composite/S-video (NTSC/PAL): meet all requirements defined in ITU-R BT.476-6.
- SD SCART analog outputs.
- All video outputs (SD Composite/S-Video, HDMI/DVO1, and DVO2) may be active simultaneously .
- Supported resolutions are 480i, 480p, 720p, 1080i, 1080p.

#### 12.1.3.7 VBI Features

- VBI Closed Captioning and other data pass-through.
- EIA/CEA-608-B Line 21 Data services – supports both SCTE 20 and SCTE 21 standards.
- EIA-708-B Digital Television (DTV) Closed Captioning – support SCTE21/ATSC A/53e standards.
- Reconstruction of lines 10 through 21 of the VBI from a data source available to the host CPU (video user data DVS706r8 – DVB/SCTE specifications based on PES encoding of VBI data Raw VBI data, Closed Caption (CC), Teletext (NABTS, WSS), CGMS, VPS, VITC, Gemstar 1x and 2x, Extended Data Service(XDS), Moji, V-Chip).

#### 12.1.3.8 Copy Protection

- NTSC Analog outputs (only for SD modes) – Macrovision version 7.1
- HDMI - HDCP
- All analog outputs – CGMS-A on all analog outputs
- Uses 59.94Hz output frame rate for both 59.94 and 60Hz source materials to maintain the same output frame rate on both NTSC and HD outputs


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.1.3.9 Video Capture Features

- HDVPSS supports two independently configurable external video input capture ports.
- Each video input capture port can be operated as one 16/24-bit input channel (with separate Y and Cb/Cr inputs) or two clock independent 8-bit input channels (with interleaved Y/C data input). Embedded sync and external sync modes are supported for all input configurations.

#### 12.1.3.10 Input Data Features

**NOTE:** Due to pinout restrictions, not all combinations are supported on all devices. Refer to the device data manual for details.

- 8-bit input ports:
  - A single non-multiplexed CIF/480i/480p/720p/1080i Y/C data
  - Up to 2-stream multiplexed ED (480p) Y/C data
  - Up to 4-stream multiplexed SD (480i) Y/C data
  - Up to 16-stream multiplexed CIF Y/C data (TBD – 16)
- 16/24-bit input ports:
  - A single non-multiplexed CIF/480i/480p/720p/1080i/1080p Y/C data
  - A single 24-bit YCbCr component or raw RGB data
  - Up to 2-stream multiplexed HD (720p/1080i) Y/C data
  - Up to 4-stream multiplexed ED (480p) Y/C data
  - Up to 8-stream multiplexed SD (480i) Y/C data
  - Up to 8-stream multiplexed CIF Y/C data
- The video capture port channel supports de-multiplexing of both pixel-to-pixel and line-to-line multiplexed streams.
- Up to 1920x1200@60Hz (160MHz) input data rate is supported for 16-bit mode input port.
- Each video capture port supports one scaler capable of both up and down scaling of one non-multiplexed input stream (one of two 8-bit channel inputs or 16-bit input channel input data).
- Each video capture port supports one programmable color space conversion to convert 24-bit RGB data to YCbCr data.
- HDVPSS supports data storage in 444, 422, and 420 formats.
- Each video capture port channel supports chroma down sampling (422 to 420) for any non-multiplexed input data. The chroma down-sampling for multiplexed streams is done as memory-to-memory operations outside of HDVPSS on an individual frame data.
- Deinterlacing is not supported.
- Ancillary data capture is supported for VANC/VBI data (ancillary data within the interval between SAV-EAV sequences during vertical blanking period) on all input channels. HANC data capture is not supported. Decoding of the Ancillary data will be handled by the software.
- Graphics overlay over incoming bitstream is NOT supported.
- Noise Reduction on the incoming video is supported as a separate memory-to-memory operation within the HDVPSS.
- One of the 8-bit channels in each video port may be used as a video write back path for transcoding applications. This channel may also use the color space conversion and/or the scaler. It is up to the application to avoid resource contention.

#### 12.1.3.11 Other Features

- The HDVPSS supports two additional video input source in 420 or 422 tiled format (same as input for main and auxiliary video paths). This path can be used for both video transcode operations (utilizing a scaler and chroma downsampler within one of the VIP ports in a memory to memory configuration) and/or driving video to the SD output display.


Copyright © 2011–2016, Texas Instruments Incorporated

  - Only 1 of the 2 additional video input sources can directly drive the SD display. For display output, there is no processing performed, that is, no scaler or deinterlacing
- The HDVPSS supports video data write back paths for following:
  - Scaled video from aux video channel (for delayed PIP display) saved as 422 video.
  - Scaled video from main video channel saved as 422 video.
  - SD scaled video from the main video compositor (delayed SD output display when CIT is off) saved as 422 video
  - Independently scaled video from main deinterlacer output (transcoding application) – saved as a 420 video using the scaler and chroma_downsampler resources in one of the VIP ports.
  - Independently scaled video from aux output (transcoding application) – saved as a 420 video using the scaler and chroma_downsampler resources in the other VIP port. This path is shared with the independent transcode video input source.
  - Independently scaled video from the 3rd or 4th video input source saved as a 420 video using a scaler and chroma downsampler resource in one of the VIP ports. The 3rd or 4th video input source can be 422 or 420 tiled input data (same as input for main and auxiliary inputs). The 3rd and 4th video input source share the VIP write back paths of the primary and auxiliary deinterlacers.
  - Write back of the scaled video from main, aux or additional video input source video channels supports both tightly (1 frame per 1 display frame period) and loosely (many frames per 1 display frame period) coupled to the display frame period. All other write backs are tightly coupled
  - The HDVPSS includes a highly programmable and bandwidth-efficient DMA controller to support the worst-case bandwidth application.

#### 12.1.3.12 Feature Limitations

The features of the HDVPSS module highlighted above are characteristics of the HDVPSS module and are not supported on all SoCs that use the HDVPSS. Common limitations exist, such as:

- **Device and system bandwidth.** The bandwidth available to the HDVPSS varies based on the frequency and width of the DDR interface provided on the device, the frequency and width of the DDR interface populated on the user's particular board, and the bandwidth demands of the remaining IP in the system. These constraints may limit the data rates that can be concurrently supported on the various input and output interfaces to the HDVPSS.
- **Device and system pinout.** Specific devices within the device family may not provide full availability of all of the Video Input and Video Output ports. In addition, pin multiplexing constraints for a particular end system may further limit the availability of specific HDVPSS signals. Refer to device data manual for details on the device availability and pin muxing with other internal signals.

**NOTE:** The following list is the convention for pin naming:

- Pins with VIN0 are connected to VIP0
- Pins with VIN1 are connected to VIP1
- Pins with VOUT0 are connected to DVO2
- Pins with VOUT1 are connected to DVO1

There is no need for any pinmux configuration for analog outputs.

**NOTE:** The block diagram and text sections of this chapter consistently use the instance numbering for the VIP and GRPX submodules as VIP0/VIP1 and GRPX0/GRPX1/GRPX2. However, the register sections use the numbering VIP1/VIP2 and GRPX1/GRPX2/GRPX3. The following list is the convention for submodule naming to register control:

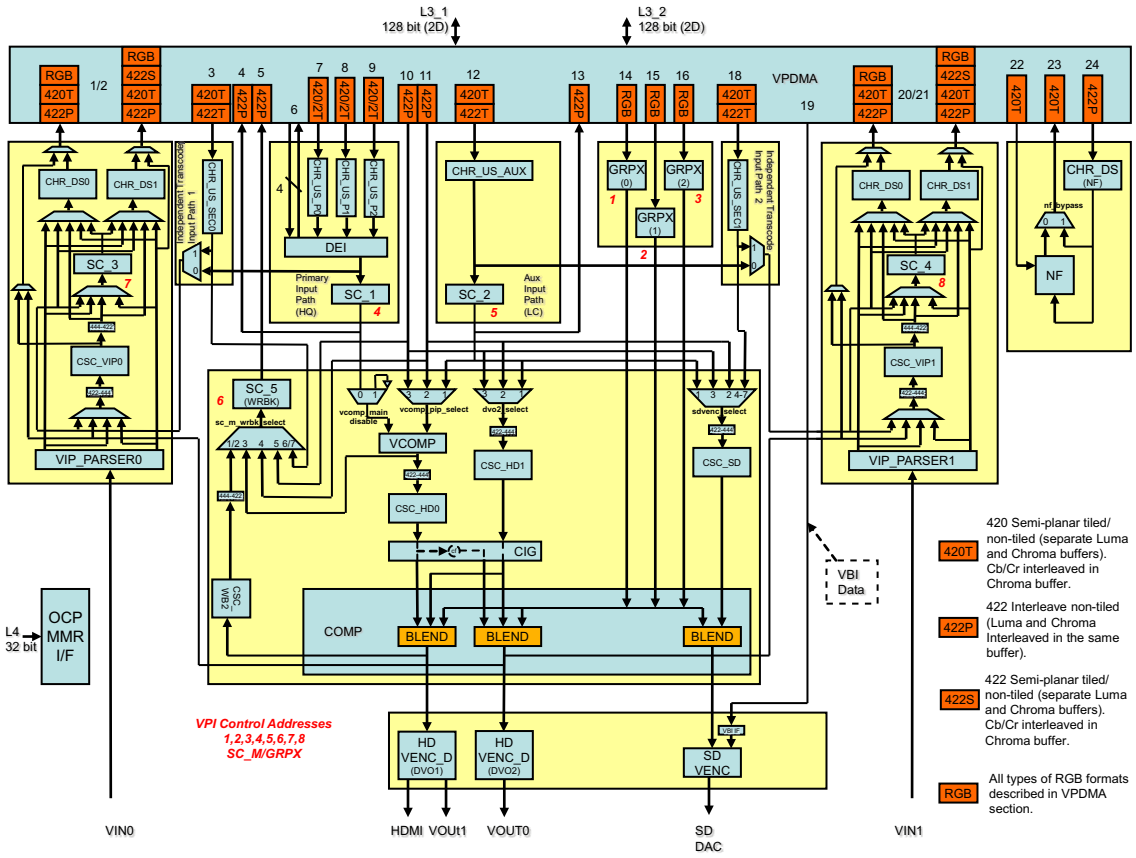
- Submodule VIP0 is controlled by registers VIP1
- Submodule VIP1 is controlled by registers VIP2
- Submodule GRPX0 is controlled by registers GRPX1
- Submodule GRPX1 is controlled by registers GRPX2
- Submodule GRPX2 is controlled by registers GRPX3


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.1.4 Functional Operation

Figure 12-1 shows a block diagram of the HDVPSS.

**Figure 12-1. HDVPSS Detailed Block Diagram**




In Figure 12-1, the primary input path contains a deinterlacer and scaler (down-scale only when the source is VIP_PARSER or the destination is VENC). The auxiliary input path contains a scaler. The main path is capable of deinterlacing up to 1080i sources. The CHR_US modules in both paths are identical.

The 4:2:2–4:4:4 and 4:4:4–4:2:2 modules are used to convert data from YUV422 space to YUV444 space, or from YUV444 space to YUV422 space. These blocks do not have any MMR for configuration and are enabled whenever required by the following modules.

Figure 12-1 also shows the VPDMA VPI Control Addresses for modules within the HDVPSS that use this interface (numbers shown in red). Modules that use the VPDMA VPI Control Address mechanism are any SC (SC_1, SC_2, SC_3, SC_4, SC_5), or GRPX module.

Note that VPDMA only supports 422 paths for display only. In addition, the numbers shown on the VPDMA ports correspond to those listed in Table 12-3.



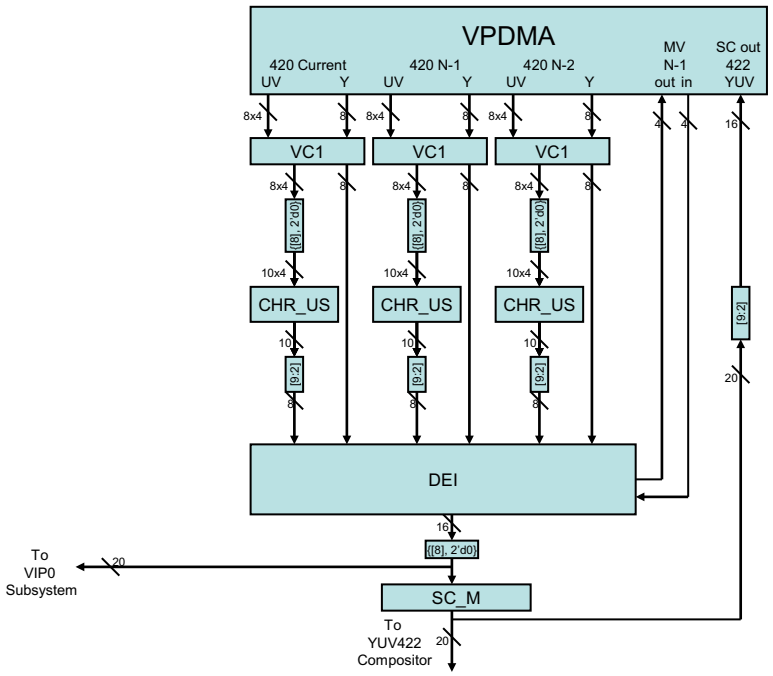
Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-3. VPDMA Client Number**

| VPDMA Client Number | HDVPSS Port |
|---|---|
| 1 / 2 | VIP0 YUV/RGB/Ancillary Data (To VPDMA) |
| 3 | Secondary-0 Input Path (From VPDMA) |
| 4 | Scaler (SC_1) Output Path (To VPDMA) |
| 5 | Scaler (SC_5) Output Path (To VPDMA) |
| 6 | DEI Motion Data (To/From VPDMA) |
| 7 | Primary Video (Fn) Input Path (From VPDMA) |
| 8 | Primary Video (Fn-1) Input Path (From VPDMA) |
| 9 | Primary Video (Fn-2) Input Path (From VPDMA) |
| 10 | Bypass Path-0 Input (BP0) (From VPDMA) |
| 11 | Bypass Path-1 Input (BP1) (From VPDMA) |
| 12 | Auxiliary Video Input Path (AUX) (From VPDMA) |
| 13 | Scaler (SC_2) Output Path (To VPDMA) |
| 14 | GRPX0 RGB/Stencil/CLUT Input Path (From VPDMA) |
| 15 | GRPX1 RGB/Stencil/CLUT Input Path (From VPDMA) |
| 16 | GRPX2 RGB/Stencil/CLUT Input Path (From VPDMA) |
| 18 | Secondary Input Path-1 Input (From VPDMA) |
| 19 | SDVENC VBI Data (From VPDMA) |
| 20 / 21 | VIP1 YUV/RGB/Ancillary Data (To VPDMA) |
| 22 | 420 Noise Filter Data (From VPDMA) |
| 23 | Noise Filter/Chroma Downsampler Output Data (To VPDMA) |
| 24 | 422 Noise Filter/Chroma Downsampler Input Data (From VPDMA) |

Figure 12-2 provides a more detailed view of the Primary Input Path (PRI), showing the interconnection between various modules and bit widths.

**Figure 12-2. Primary Input Path (PRI) Detailed Block Diagram**

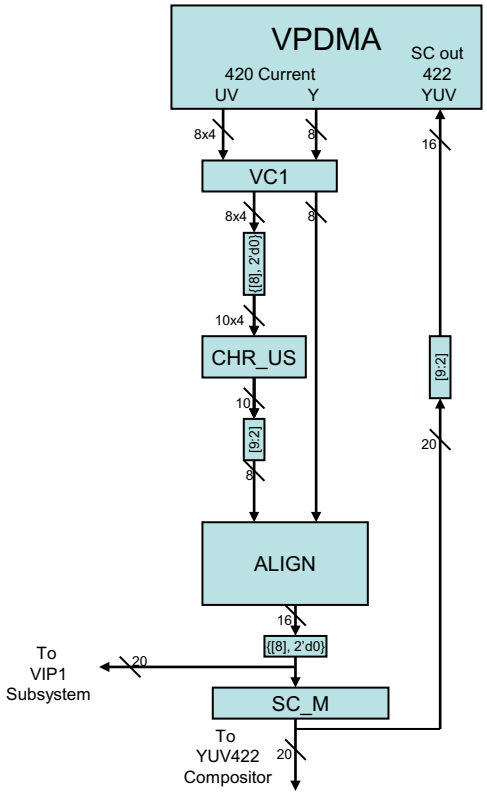




Copyright © 2011–2016, Texas Instruments Incorporated

Figure 12-3 provides a more detailed view of the Auxiliary Input Path (AUX), showing the interconnection between various modules and bit widths.

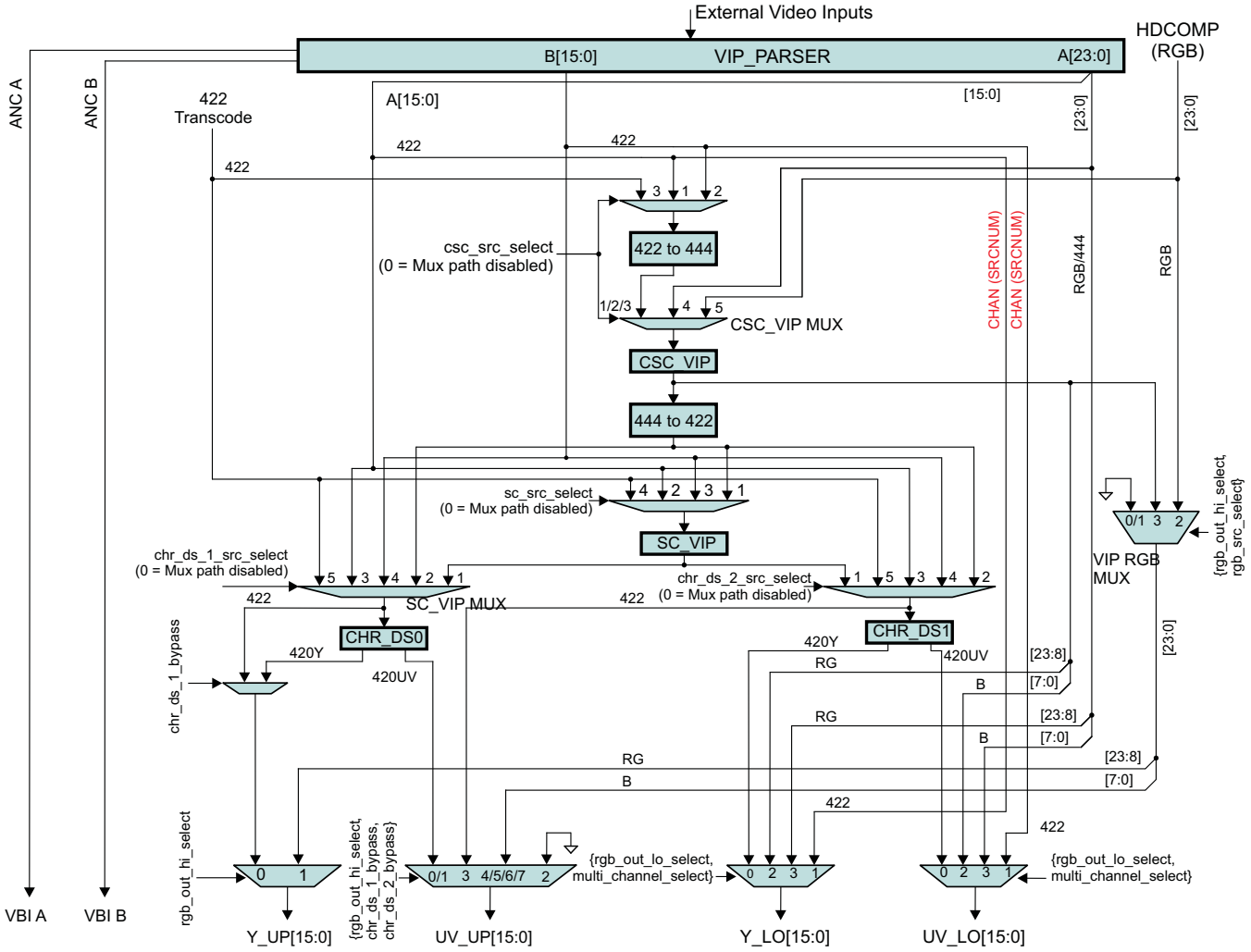
**Figure 12-3. Auxiliary Input Path Detailed Block Diagram**




The previous diagrams show the details of the interconnect and selections possible within the Video Input Ports. In Figure 12-4, the signals shown on the Video Input Port Parser (VIP_PARSER) module are the outputs generated by this module. External video input drives the input side of this module. A_Y can be in either YUV422 format (A_Y[7:0] in the diagram) or RGB/YUV444 format (A_Y[23:0] in the diagram) depending on the external video input source and configuration options within the VIP_PARSER. If the VIP_PARSER is configured to capture 24bit RGB/444 data, A_Y[23:0] is used and the data path inside the Video Input Port must be configured correctly for it. Multiplexor selections and controls are shown above, and described in CLKC VIN0/2 Data Path Register. The output of the Video Input Port drives the VPDMA module, which sends the resulting video to SDRAM.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-4. VIP Subsystem Detailed Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

### 12.1.4.1 Functional Module Descriptions

The HDVPSS module is composed of many processing blocks. There are multiple instances of most of these processing blocks. The following module descriptions provide a high-level description of each processing module.

#### *12.1.4.1.1 VPDMA*

The VPDMA module is capable of transporting data to and from an external memory location, most often external DDR memory, buffering this data and then delivering the data as demanded to HDVPSS modules as programmed. Additionally in a third-party configuration the VPDMA is capable of performing DMA transfers as requested by the pulsing of an event strobe.

The VPDMA module is also capable of generating an address which may reach any location in the range for which it is configured. It is capable of moving this data either to or from a Shared Buffer and ultimately to or from a Client Buffer. For the two type of Client Buffers that are used to drive data into a subsystem (Streaming Buffer and Random Access Buffer) data is either pushed into the buffer or pulled out of the buffer dependent upon the direction of the data transfer.

For third-party DMA operations the data is transferred into a Client buffer, and then transferred out of the Client Buffer to the transfer destination. These transfers are triggered by an Event pulse to each of the Client Buffers which are Routing Buffer types.

##### *12.1.4.1.1.1 VPDMA Details*

The modules that source or sink data are referred to as clients. That is, the physical interface between the processing module and external memory is called a client. A channel is the mechanism inside the VPDMA that connects a specific memory buffer or transfer to a specific client.

Each of the channels has a type of data that it can support based upon the client that it services. Based on this, four kinds of channel has been defined:

- YUV Channel—Clients taking data in YUV format like DEI, NF, and so on.
- RGB Channel—Clients taking RGB data like GRPX
- Miscellaneous Channel—The Miscellaneous channel type is for any data type that is not a normal video type. The Miscellaneous channel type makes no assumptions on data type and just passes the data to the client and supports a single buffer for the client.
- Free Channel—Used in video compositions. The Free channel data type is always ignored as it uses the same data type of the descriptor that first calls the free channel.

The clients of the VPDMA are the physical connection to the processing modules that source or sinks of the data and control when the channel can be updated. Each client can be configured to have specific start event which allows for the channel data to start flowing to or from the external client. The start event can be selected from one of the signals in the FRAME START Interface or it can be selected by a channel attribute or to be controlled by an internal Frame signal controlled by the List Manager.

The VPDMA controller works on lists of descriptors. Descriptors define how data will be transferred through different channel. In other words, channel is described through a Data Transfer descriptor. The client that the channel is mapped to interprets the information in the descriptor to perform the requested data transfer. List is a group of descriptors that makes up a set of DMA transfers that need to be completed. List can contain any kind of descriptor without limitation and be of any size.

The CPU creates a list of descriptors in DDR in the order it wants them executed. The CPU then writes the location of the list to the LIST_ADDR register, followed by writing the size, type of list and list number of the LIST_ATTR register and posts the list. The List Manager module, which manages different lists, will then schedule a DMA transfer to pull in the portion of the list that it can store in the internal VPDMA memory. The List Manager will sequentially process descriptor in the list and performs the data transfers.


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.1.5 HDVPSS Data Flow

#### 12.1.5.1 Data Flow Control

The VPDMA routes input and output video sources to the starting or ending point of each processing pipeline.

Signal routing within the HDVPSS is controlled by an internal protocol referred to as Video Peripheral Protocol (VPI Protocol). This protocol consists of a NewFrame (NF) signal at the start of each frame which indicates to downstream modules that a frame of data is about to start. On the last signal of a frame, an EndofFrame signal is sent to indicate the last pixel of a frame. In addition to preparing downstream modules for frames of data, the NewFrame signal is used in many modules to stage the memory mapped registers. Each processing module will receive a NewFrame signal, and will generate a NewFrame signal to send to its downstream module. Sections in the register section which refer to New Frame, or NF handling, refer to this signal.

Internal Control is accomplished by configuring the selection of data paths available at the input of each of the multiplexers (see Figure 12-1). This configuration can be done in the CLKC Data Path Select Register. This register controls where internal processing pipes are connected.

#### 12.1.5.2 VCOMP Mux

The video compositor (VCOMP) module takes two inputs: MAIN and PIP. The MAIN input comes from Primary Input Path, and PIP input can be selected from one of three sources: Auxiliary Input Path (AUX), Bypass Path0 (BP0), or Bypass Path1 (BP1). The default for the PIP path is disabled, so there is no PIP input unless it is configured.

#### 12.1.5.3 HD Display Mux

The HD display mux selects a data path for HD displays, which include HDMI/DVO1, and DVO2 outputs. The input data path can be selected from three sources: Auxiliary Input Path (AUX), Bypass Path0 (BP0), or Bypass Path1 (BP1). The default for the input data path is disabled, so no input path is selected unless it is configured.

#### 12.1.5.4 SD Display Mux

The SD display mux selects a data path for SD display that includes SDVENC output. The input data path can be selected from one of four sources: Auxiliary Input Path (AUX), Bypass Path0 (BP0), Bypass Path1 (BP1), or Secondary-1 Input Path (SEC1). The default for the input data path is disabled, so no SDVENC input path is selected unless it is configured.

#### 12.1.5.5 SC_5 Mux

The SC_5 mux selects a data path for writing a processed image back to memory. The input data path for this MUX can be selected from one of six sources:

- HDMI Composited Path Output (from COMP)
- VCOMP Block Output
- Bypass Path0 (BP0)
- Bypass Path1 (BP1)
- Secondary-0 Input Path (SEC0)

The default for the input data path is disabled, so there is no scaler input or output.

#### 12.1.5.6 CSC_VIP Mux

The color space converters (CSC_VIP0 and CSC_VIP1) within the VIP subsystem receive data from one of five sources:

- VIP_PARSER PortA Output (422)
- VIP_PARSER PortB Output (422)


Copyright © 2011–2016, Texas Instruments Incorporated

- Secondary Path Output (422): SEC0 for CSC_VIP0 and SEC1 for CSC_VIP1
- VIP_PARSER PortA Output (RGB)
- Compositer Output (RGB) from COMP.

The default for this mux is disabled, so there is no CSC_VIP input.

#### 12.1.5.7 SC_VIP Mux

The scaler modules (SC_VIP0 and SC_VIP1) within the VIP subsystem receive data from one of three sources:

- Color Space Converter Output (from CSC_VIP)
- VIP_PARSER PortA Output
- VIP_PARSER PortB Output
- Secondary Path Output: SEC0 for SC_VIP0 and SEC1 for SC_VIP1

The default for this mux is disabled, so there is no SC_VIP input.

#### 12.1.5.8 VIP RGB Mux

The RGB output of the VIP subsystem can come from one of two sources:

- Compositer Output (from COMP)
- Color Space Converter Output (from CSC_VIP)

#### 12.1.5.9 Video Compositor PIP Input

The Video Compositor PIP input path can come from 1 of 3 sources: The Auxiliary Input Pipe, or from either of the Bypass-422 pipes (non-tiled format). These sources will usually be either the Primary Composited Writeback Path or the memory stored version of the Auxiliary Input Path. However, the only requirement on these two VPDMA input paths is that they have been stored in 422 “private data” format, which is an untiled raster format.

The default for this path is disabled, meaning no PIP input.

#### 12.1.5.10 PIP Display Input

The HD Composite VENC path can come from 1 of 3 sources: The Auxiliary Input Pipe, or from either of the Bypass-422 pipes (non-tiled format). These sources will usually be either the Primary Composited Writeback Path or the memory stored version of the Auxiliary Input Path. However, the only requirement on these two VPDMA input paths is that they have been stored in 422 “private data” format, which is an untiled raster format.

The default for this path is disabled, meaning no input.

#### 12.1.5.11 Standard Definition Video Encoder Input

The standard definition vdeo encoder (SD VENC) input path can come from 1 of 4 sources: The auxiliary input pipe, the secondary-2 input pipe or one of the two bypass-422 input pipes. These sources will usually be either the primary composited writeback path or the memory stored version of the Auxiliary Input Path. However, the only requirement on these two VPDMA input paths is that they have been stored in 422 “private data” format, which is an untiled raster format.

The default for this path is disabled, meaning no SD VENC input.

#### 12.1.5.12 Primary Composited Writeback Scaler Input

The primary composited writeback scaler Input is used to scale the primary input path (normally HD) and send to memory so that this data can be fed back into the HDVPSS for display on either the high-definition VENC or standard-definition VENC. The input to this scaler can come from 1 of 6 sources: the output of the video compositor, the composited HDMI output, one of the two bypass-422 paths, or the secondary-1 input.


Copyright © 2011–2016, Texas Instruments Incorporated

The default for this path is disabled, meaning no scaler input (or output).

#### 12.1.5.13 Video Input Port Color Space Converter Input (2)

The color space converters within the video inputs ports can come from 1 of 3 sources: The Video Input Port Parser (VIP_PARSER), the composited HD DAC output or from either the Primary Input DEI (Video Input Port 0) or the Auxiliary Input DEI (Video Input Port 1). The feeds from the primary and auxiliary DEIs are for dual stream transcode (Secondary Input Paths) with write back to memory as RGB data.

The default for this path is disabled, meaning no CSC input (or output).

#### 12.1.5.14 Video Input Port Scaler Input (2)

The scalers within each of the video input ports can come from 1 of 3 sources: The Video Input Port Parser (A or B channel), the Video Input Port Color Space Converter or either the Primary Input DEI (Video Input Port 0) or either the Auxiliary Input DEI or Secondary Video Path Input (Video Input Port 1). The feeds from the Primary and Auxiliary input DEIs are for dual stream transcoding purposes.

The default for this path is disabled, meaning no scaler input.

#### 12.1.5.15 Video Input Port RGB Output (2)

The RGB output of the Video Input Port can come from 1 of 2 sources: The Video Input Port Parser or the Video Input Port Color Space Converter. The feed from the Video Input Port Color Space Converter are for dual stream transcode (via Secondary Input Path) with write back to memory as RGB data. The source of this data is either the Primary Input DEI (Video Input Port 0) or the Auxiliary Input DEI (Video Input Port 1). When the source of the Video Input Port Color Space Converter is set to one of the DEI inputs, this mux will pass the output of the Color Space Converter out instead of the output of the Video Input Port Parser.

The default for this path is from the Video Input Port Parser (VIP_PARSER).

#### 12.1.5.16 Noise Filter Output

The Noise Filter Output can come from 1 of 2 inputs: The Noise Filter or from the Noise Filter Chroma Downsampler (allowing this chroma downsampler to be used standalone).

The default for this path is connected to the Noise Filter output.

##### *12.1.5.16.1 Data Formats*

The following section describes the data formats for the HDVPSS. Each of the channels has a type of data that it can support based upon the client that it services. Also, for each channel the data type will assume that data is packed in memory a certain way and will be presented to the client in the same manner no matter what the format of the data in memory.

The YUV formatted channels expect video data that is in YUV color space. The YUV data type is for YUV data and it can support both interleaved data where Luma and Chroma are in the same data buffer or it can support co-planar data where the Luma and Chroma are in separate data buffers. The YUV channel also can be configured to give a different start position for the client instead of the default upper left hand corner of the frame. The storage format for YUV data depends on the data type field. The data type must be set to a data type that the channel can support. All the clients that data are provided to will either accept Luma Only, Chroma Only or Luma and Chroma in a parallel data bus.

The data formats in Figure 12-5 map as follows:

422P data:

- 422 interleave non-tiled (luma and chroma interleaved in the same buffer)

422S data:

- 422 semi-planar tiled/non-tiled (separate luma and chroma buffers). Cb/Cr interleaved in chroma buffer.

420T data:


Copyright © 2011–2016, Texas Instruments Incorporated

- 420 semi-planar tiled/non-tiled (separate luma and chroma buffers). Cb/Cr interleaved in Chroma buffer.

RGB data:

- All types of RGB formats described in VPDMA section.

Each of these data formats are discussed in the VPDMA data formats section.

### 12.1.6 Example Data Flows

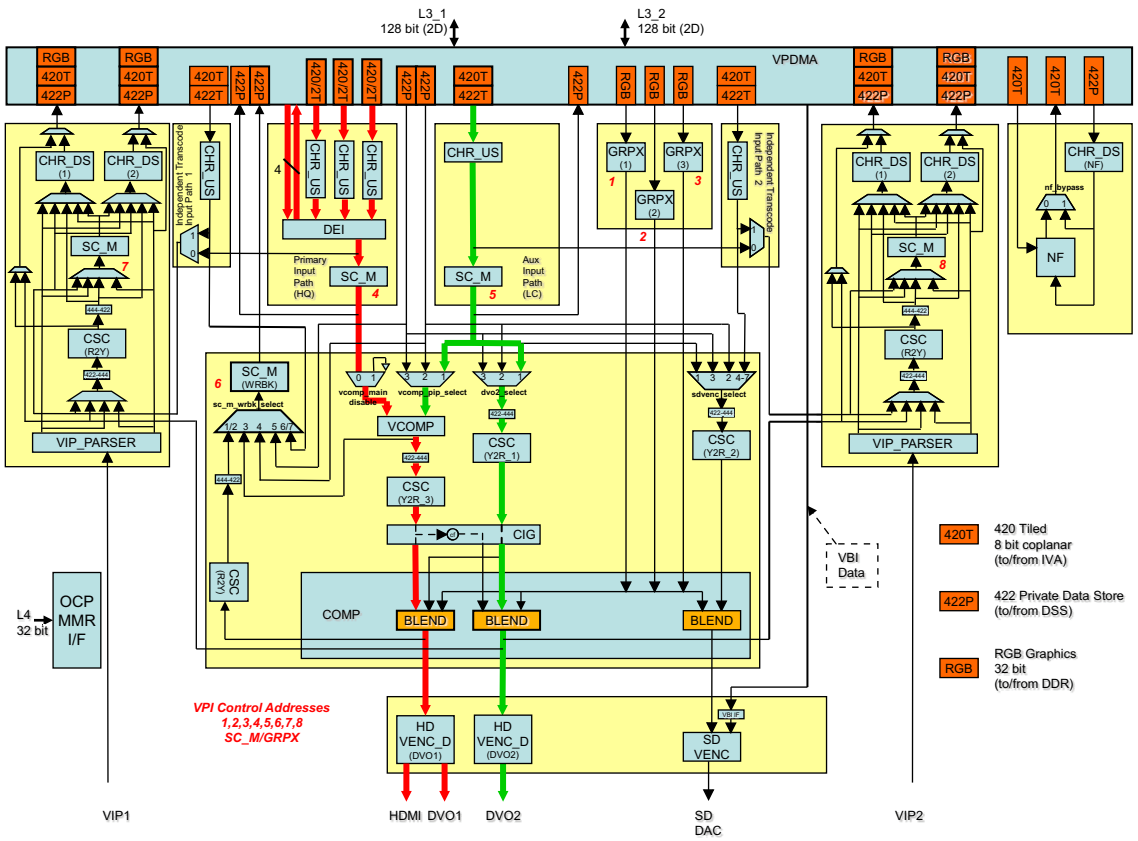
Several concurrent data flows within the HDVPSS are possible. The following sections outline several of the possible data flow scenarios capable within the HDVPSS, including the HDVPSS level mux register selections required for the flow. The flows shown are not all encompassing, and there are variations within each. For example, the first data flow shown is Dual Display. Single Display is also possible. Dual Transcode (Secondary Input Path 1 and 2) is possible, or Single Transcode (Secondary Input Path 1). Transcode is possible without display, etc.

#### 12.1.6.1 Dual Display

A standard dual display data flow is shown in Figure 12-5.

In this case, the primary display (HDMI) contains two video sources in a Picture-in-Picture format. On the secondary HD display, the auxiliary display is being shown.

**Figure 12-5. HDVPSS Dual Display (PIP) Data Flow**






Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-4. HDVPSS Mux Select Register 1**

| HDVPSS Mux Select Register | Value Required |
|---|---|
| vcomp mux select | 1 |
| hdcomp mux select | 1 |
| sd mux select | 0 |
| sc_5 mux select | 0 |
| sec0 mux select | 0 |
| sec1 mux select | 0 |
| csc_vip0 mux select | 0 |
| sc_vip0 mux select | 0 |
| chr_ds0_vip0 mux select | 0 |
| chr_ds1_vip0 mux select | 0 |
| csc_vip1 mux select | 0 |
| sc_vip1 mux select | 0 |
| chr_ds0_vip1 mux select | 0 |
| chr_ds1_vip1 mux select | 0 |

#### 12.1.6.2 Tri Display

There are several display configurations for the Tri-Display format.

The first configuration, Figure 12-6, shows the auxiliary video path on the SD display. In this case, graphics overlays are also being applied to each video encoder output. Because HD DV02 and SD DAC paths are sharing the same input (meaning same size and interlaced format), they share a graphics input. In this case, because the SD path must be interlaced, all video paths must be interlaced.

The second configuration, Figure 12-7, shows the HDMI composited Picture-in-Picture display content on the SD display. In this case, the three output displays are independent. The SC WRBK scaler will be used to scale the primary blended display and convert to interlaced format for the SD output.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-6. HDVPSS Tri Display Data Flow 1**

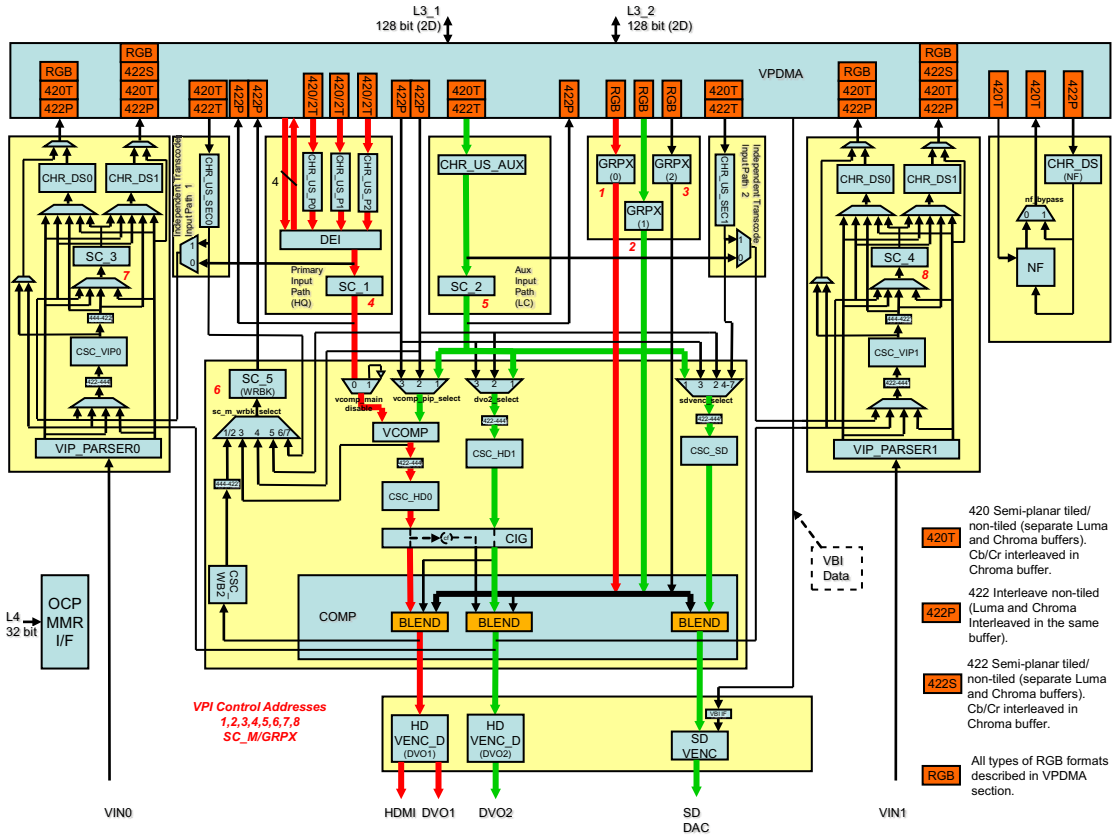



**Table 12-5. HDVPSS Mux Select Register 2**

| HDVPSS Mux Select Register | Value Required |
|---|---|
| vcomp mux select | 1 |
| hdcomp mux select | 1 |
| sd mux select | 1 |
| sc_5 mux select | 0 |
| sec0 mux select | 0 |
| sec1 mux select | 0 |
| csc_vip0 mux select | 0 |
| sc_vip0 mux select | 0 |
| chr_ds0_vip0 mux select | 0 |
| chr_ds1_vip0 mux select | 0 |
| csc_vip1 mux select | 0 |
| sc_vip1 mux select | 0 |
| chr_ds0_vip1 mux select | 0 |
| chr_ds1_vip1 mux select | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-7. HDVPSS Tri Display Data Flow 2**

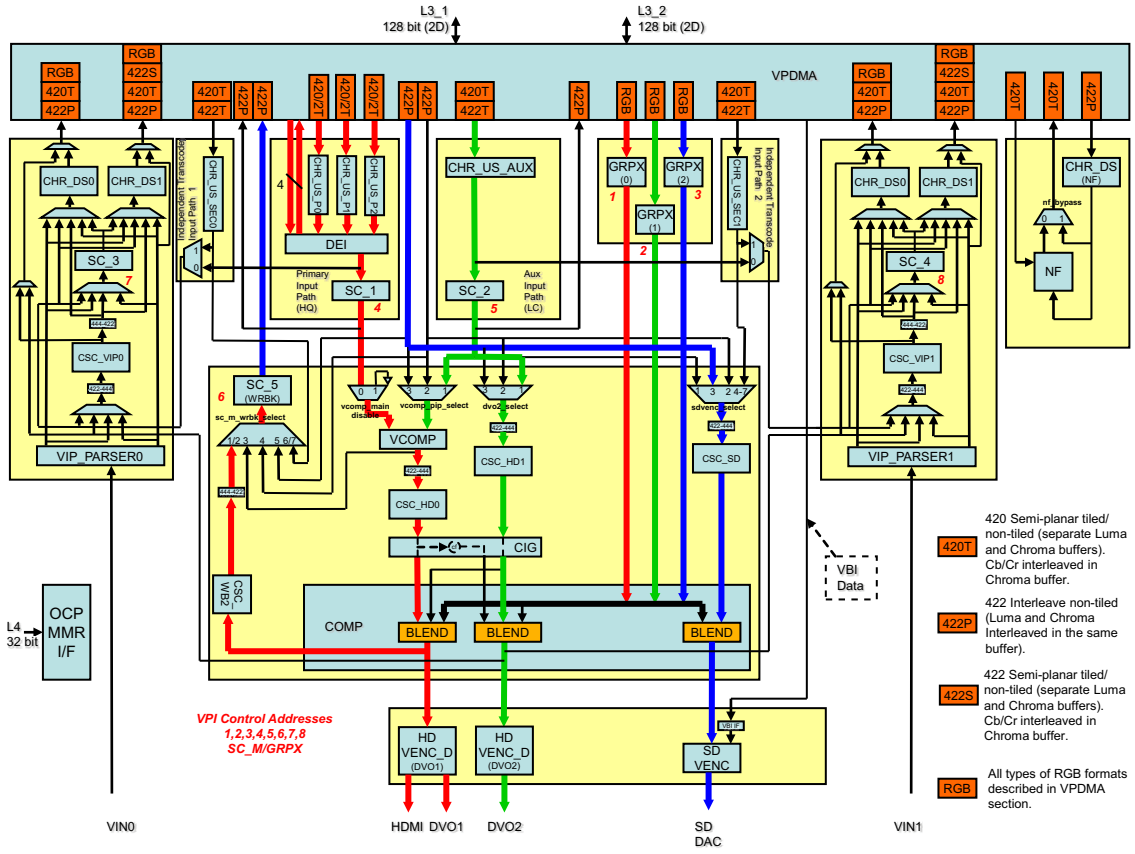



**Table 12-6. HDVPSS Mux Select Register 3**

| HDVPSS Mux Select Register | Value Required |
|---|---|
| vcomp mux select | 1 |
| hdcomp mux select | 1 |
| sd mux select | 2 |
| sc_5 mux select | 1 |
| sec0 mux select | 0 |
| sec1 mux select | 0 |
| csc_vip0 mux select | 0 |
| sc_vip0 mux select | 0 |
| chr_ds0_vip0 mux select | 0 |
| chr_ds1_vip0 mux select | 0 |
| csc_vip1 mux select | 0 |
| sc_vip1 mux select | 0 |
| chr_ds0_vip1 mux select | 0 |
| chr_ds1_vip1 mux select | 0 |

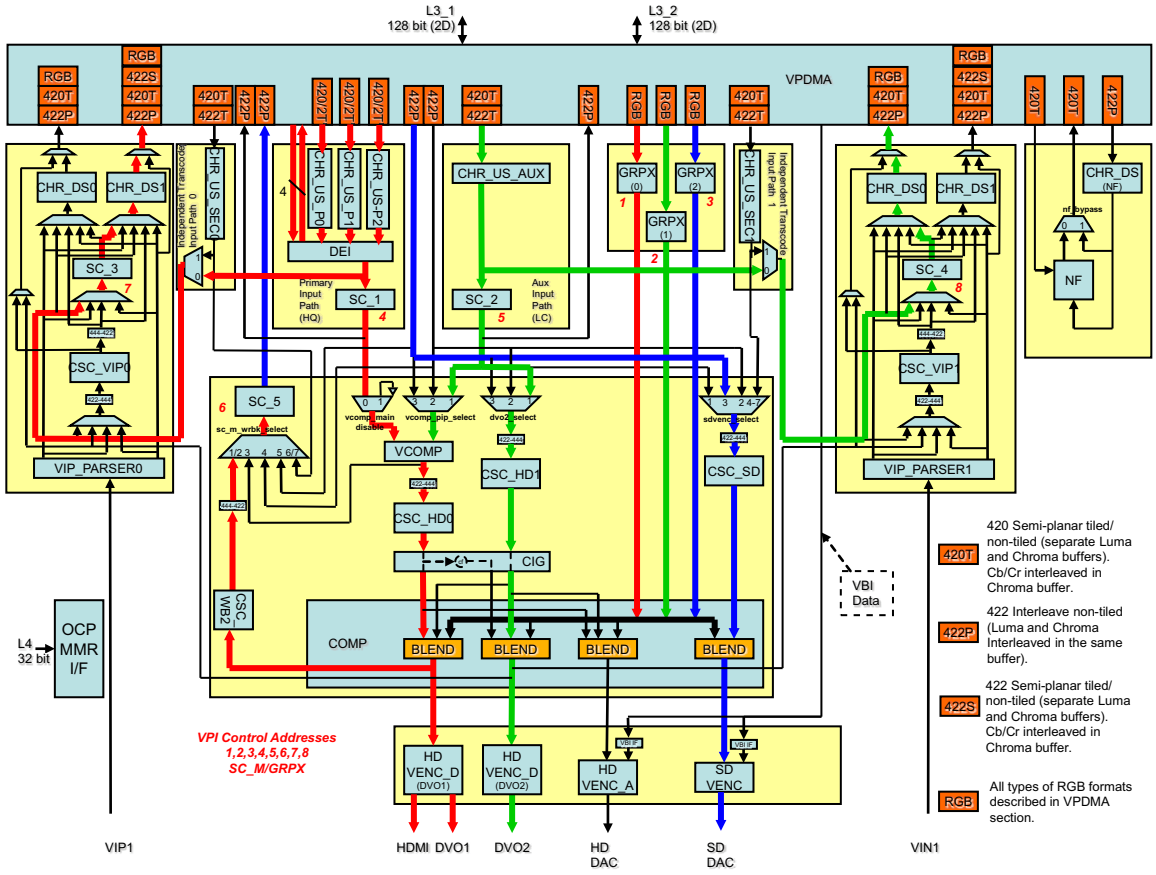

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.1.6.3 Tri Display and Dual Transcode

In addition to display, each video source can be transcoded while being displayed through the use of the Secondary Input Paths.

Either or both video sources can be transcoded while simultaneously being displayed.

**Figure 12-8. HDVPSS Tri Display Dual Transcode**






Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-7. HDVPSS Mux Select Register 4**

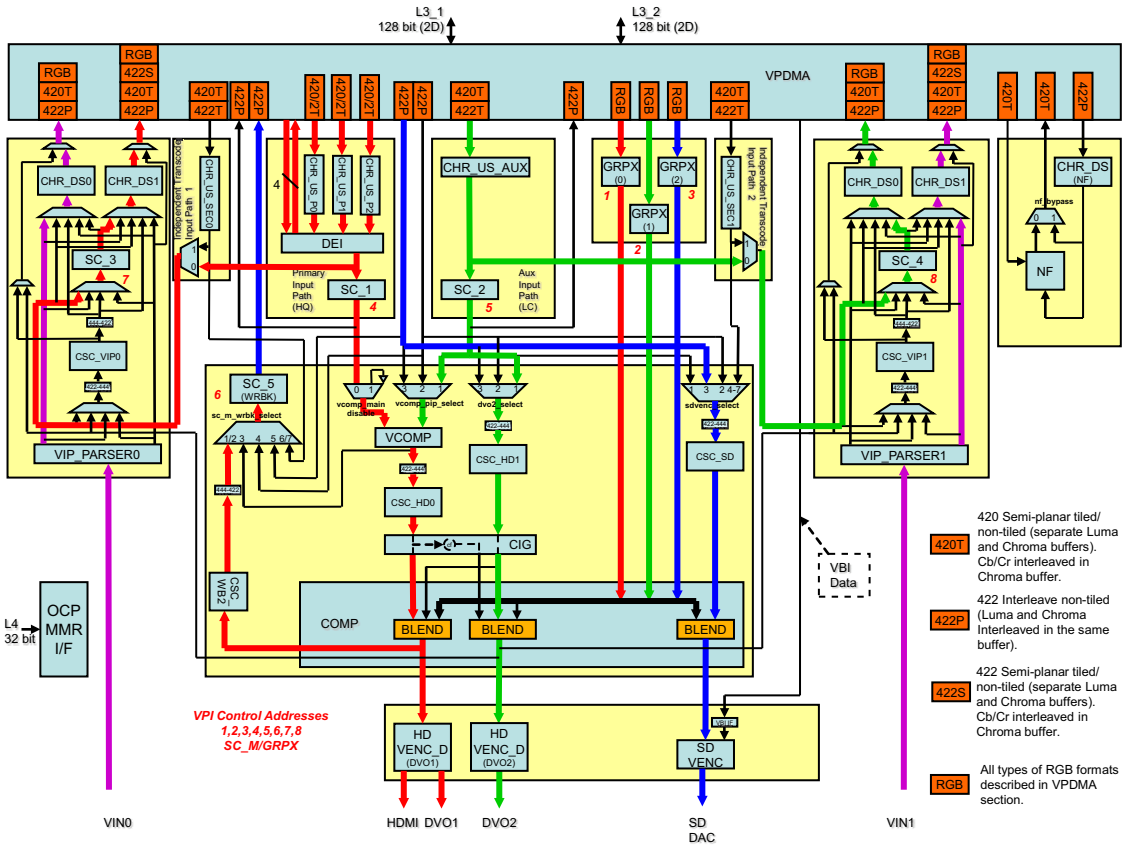
| HDVPSS Mux Select Register | Value Required |
|---|---|
| vcomp mux select | 1 |
| hdcomp mux select | 1 |
| sd mux select | 2 |
| sc_5 mux select | 1 |
| sec0 mux select | 0 |
| sec1 mux select | 0 |
| csc_vip0 mux select | 0 |
| sc_vip0 mux select | 4 |
| chr_ds0_vip0 mux select | 0 |
| chr_ds1_vip0 mux select | 1 |
| vin0_rgb_out_hi_select | 0 |
| vin0_rgb_out_lo_select | 0 |
| vin0_multi_channel_select | 0 |
| csc_vip1 mux select | 0 |
| sc_vip1 mux select | 4 |
| chr_ds0_vip1 mux select | 1 |
| chr_ds1_vip1 mux select | 0 |
| vin1_rgb_out_hi_select | 0 |
| vin1_rgb_out_lo_select | 0 |
| vin1_multi_channel_select | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.1.6.4 Tri Display, Dual Transcode, and Video Capture

The Video Input Port has the capability to capture either one or two video sources. In the previous case, only one video source can be captured per Video Input Port, since the other path is dedicated to transcode. Also in this case, video captured cannot be resized.

**Figure 12-9. HDVPSS Tri Display Dual Transcode with Video Capture**






Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-8. HDVPSS Mux Select Register 5**

| HDVPSS Mux Select Register | Value Required |
|---|---|
| vcomp mux select | 1 |
| hdcomp mux select | 1 |
| sd mux select | 2 |
| sc_5 mux select | 1 |
| sec0 mux select | 0 |
| sec1 mux select | 0 |
| csc_vip0 mux select | 0 |
| sc_vip0 mux select | 4 |
| chr_ds0_vip0 mux select | 3 |
| chr_ds1_vip0 mux select | 1 |
| vin0_rgb_out_hi_select | 0 |
| vin0_rgb_out_lo_select | 0 |
| vin0_multi_channel_select | 0 |
| csc_vip1 mux select | 0 |
| sc_vip1 mux select | 4 |
| chr_ds0_vip1 mux select | 1 |
| chr_ds1_vip1 mux select | 3 |
| vin1_rgb_out_hi_select | 0 |
| vin1_rgb_out_lo_select | 0 |
| vin1_multi_channel_select | 0 |

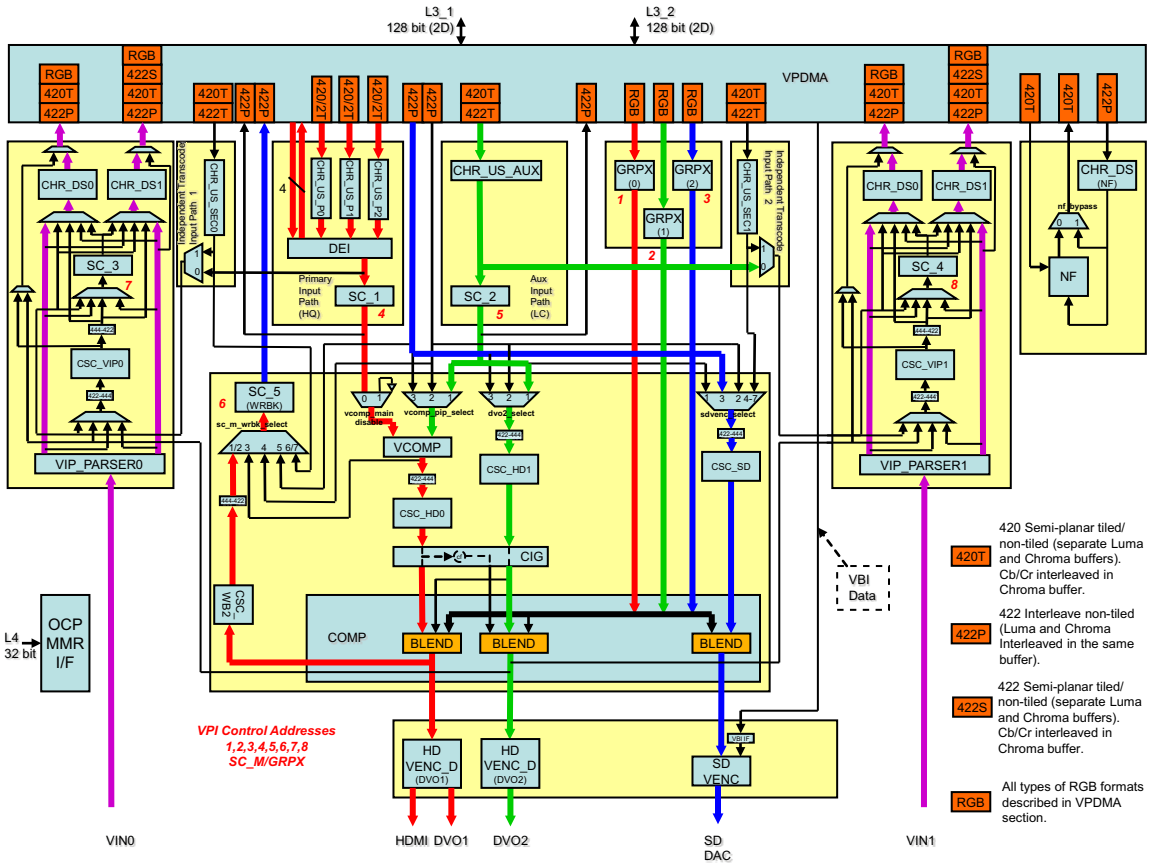

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.1.6.5 Tri Display and Video Capture

As can be seen in Figure 12-10, as long as there is no transcode operations, video display and video capture are independent. In this usage, up to four video sources can be captured (two per video input port) while display is active.

To improve the quality of captured video sources, the Noise Filter (NF) can be used on captured video (see Figure 12-11). This Noise Filter is a memory to memory operation, meaning that captured video goes to memory, and then is read back into to the Noise Filter, operated on, and written back to memory. The below diagram shows this flow (with display active).

**Figure 12-10. HDVPSS Tri Display with Video Capture**





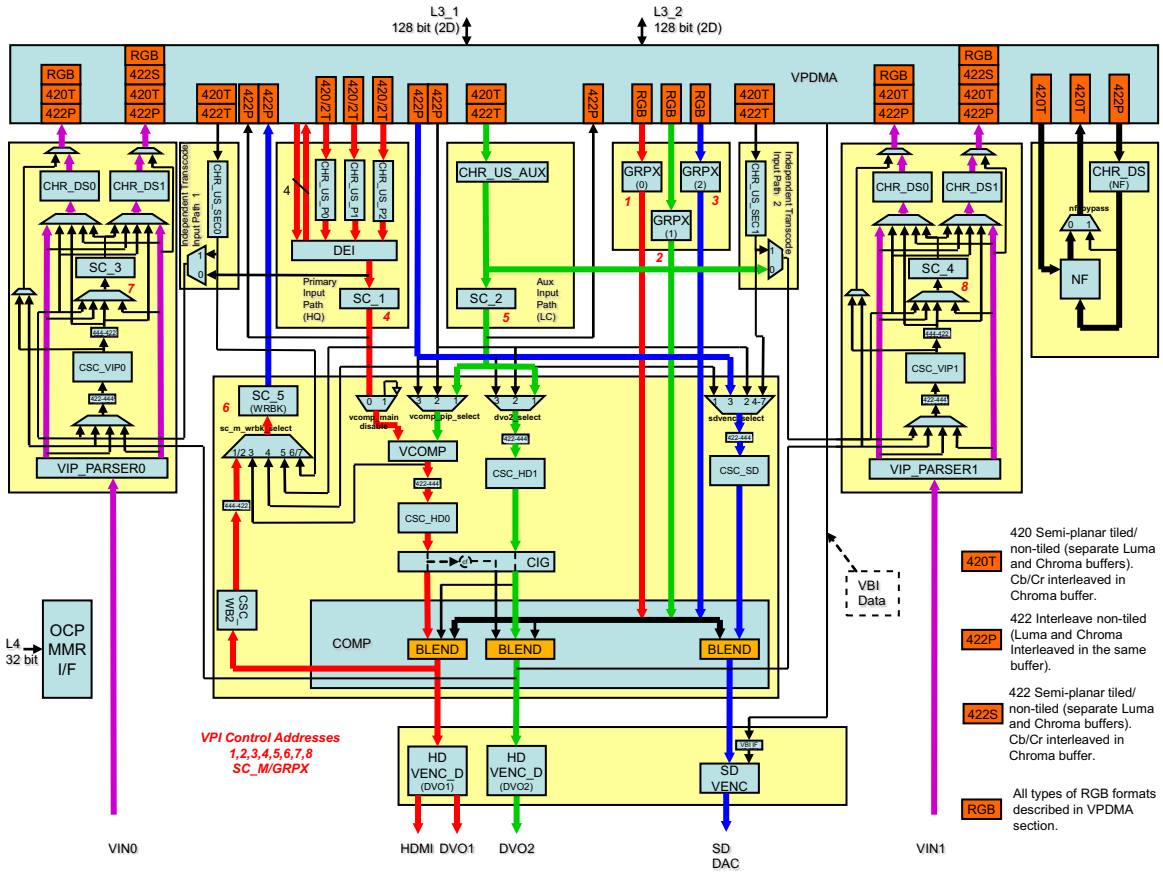

Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-9. HDVPSS Mux Select Register 6**

| HDVPSS Mux Select Register | Value Required |
|---|---|
| vcomp mux select | 1 |
| hdcomp mux select | 1 |
| sd mux select | 2 |
| sc_5 mux select | 1 |
| sec0 mux select | 0 |
| sec1 mux select | 0 |
| csc_vip0 mux select | 0 |
| sc_vip0 mux select | 0 |
| chr_ds0_vip0 mux select | 3 |
| chr_ds1_vip0 mux select | 4 |
| vin0_rgb_out_hi_select | 0 |
| vin0_rgb_out_lo_select | 0 |
| vin0_multi_channel_select | 0 |
| csc_vip1 mux select | 0 |
| sc_vip1 mux select | 0 |
| chr_ds0_vip1 mux select | 3 |
| chr_ds1_vip1 mux select | 4 |
| vin1_rgb_out_hi_select | 0 |
| vin1_rgb_out_lo_select | 0 |
| vin1_multi_channel_select | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-11. HDVPSS Tri Display with Video Capture and Noise Filter**






Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-10. HDVPSS Mux Select Register 7**

| HDVPSS Mux Select Register | Value Required |
|---|---|
| vcomp mux select | 1 |
| hdcomp mux select | 1 |
| sd mux select | 2 |
| sc_5 mux select | 1 |
| sec0 mux select | 0 |
| sec1 mux select | 0 |
| csc_vip0 mux select | 0 |
| sc_vip0 mux select | 0 |
| chr_ds0_vip0 mux select | 3 |
| chr_ds1_vip0 mux select | 4 |
| vin0_rgb_out_hi_select | 0 |
| vin0_rgb_out_lo_select | 0 |
| vin0_multi_channel_select | 0 |
| csc_vip1 mux select | 0 |
| sc_vip1 mux select | 0 |
| chr_ds0_vip1 mux select | 3 |
| chr_ds1_vip1 mux select | 4 |
| vin1_rgb_out_hi_select | 0 |
| vin1_rgb_out_lo_select | 0 |
| vin1_multi_channel_select | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.1.7 Interrupt Mapping

HDVPSS generates four interrupts which can be mapped different processors as listed in Table 12-11.

**Table 12-11. HDVPSS Interrupts**

| Interrupt No. | Interrupt Name (INTRx) | Mapped Processor | MMR Address Offset Range |
|---|---|---|---|
| 0 | INTR0 | Cortex-A8 | 0x20 - 0x3F |
| 1 | INTR1 | GEM | 0x40 - 0x5F |
| 2 | INTR2 | Media controller processors | 0x60 - 0x7F |
| 3 | INTR3 | Media controller processors | 0x80 - 0x9F |

Interrupts INTR2 and INTR3 are mapped to both media controller processors (Video Media Controller and HDVPSS Media Controller).

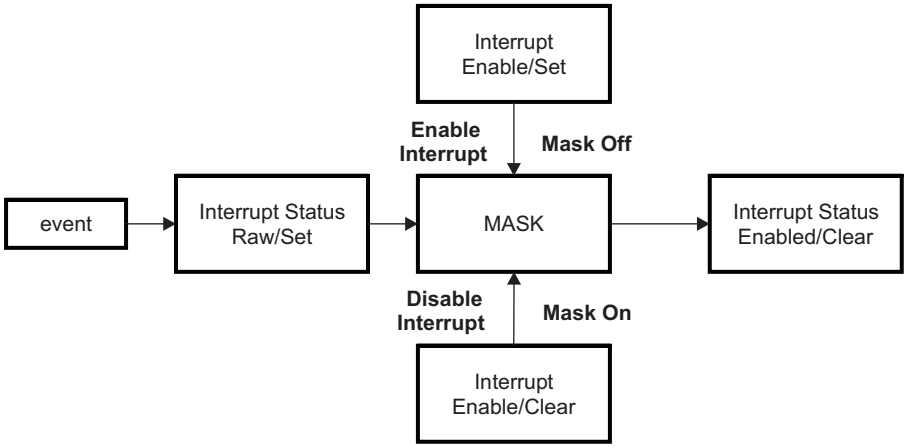
The configuration of each of the above interrupts (INTRx) is controlled by the MMR present in the offset address range mentioned above in INTC (offset: 0x0100) module.

The following group of MMR constitutes the configuration of interrupts.

- Interrupt Status Raw/Set
- Interrupt Status Enabled/Clear
- Interrupt Enable/Set
- Interrupt Enable/Clear

Figure 12-12 shows the functionality of these MMR.

**Figure 12-12. MMR Functionality**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.1.7.1 Status Before Mask

Interrupt Status Raw/Set register bit-fields show the status of interrupts before the MASK. A bit-field is set to ‘1’ when either an event happened on the signal mentioned in the corresponding bit-field or manual write of ‘1’ to the corresponding bit-field. Writing ‘0’ has no effect.

#### 12.1.7.2 Status After Mask

Interrupt Status Enabled/Clear register bit-fields show the status of interrupts after the MASK. The value of Interrupt Status Raw/Set register is transferred to Interrupt Status Enabled/Clear register for the bit-fields with MASK off. Writing ‘1’ to a bit-field clears its value to ‘0’ and also clears the corresponding bit-field in Interrupt Status Raw/Set register.

#### 12.1.7.3 Mask OFF (Enable Interrupt)

Interrupt Enable/Set register is used to disable the MASK for selected signals in Interrupt Status Raw/Set register. Writing ‘1’ to a bit-field enables corresponding interrupt to propagate to the processor. Writing ‘0’ has no effect.

#### 12.1.7.4 Mask ON (Disable Interrupt)

Interrupt Enable/Clear register is used to enable the MASK for selected signals in Interrupt Status Raw/Set register. Writing ‘1’ to a bit-field disables corresponding interrupt to the processor. Writing ‘0’ has no effect.

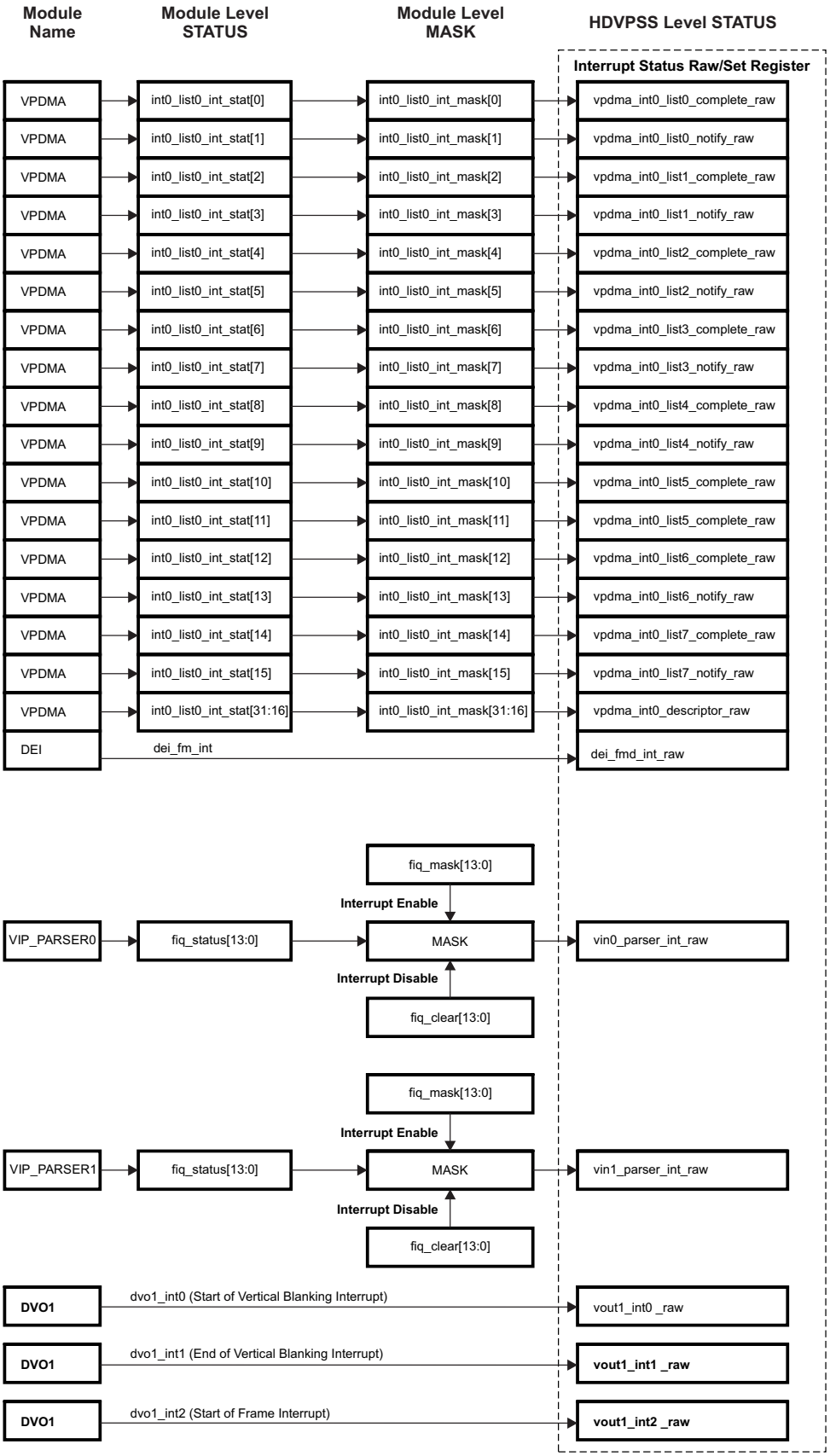
Please note that all interrupts are disabled by default.

There are 50 interrupts that can be mapped to each INTRx at HDVPSS level and everyone of the four INTRx has access to all the 50 available interrupts. One ‘bit’ is allocated for each interrupt in the MMR and spread across two memory mapped registers (Register0 and Register1) to fit in all the 50 interrupts.

The source of interrupts described at HDVPSS level come from Sub-Modules of HDVPSS like VPDMA, DEI, VIP0, etc. Figure 12-13 shows interrupt mapping from module level to HDVPSS level.


Copyright © 2011–2016, Texas Instruments Incorporated

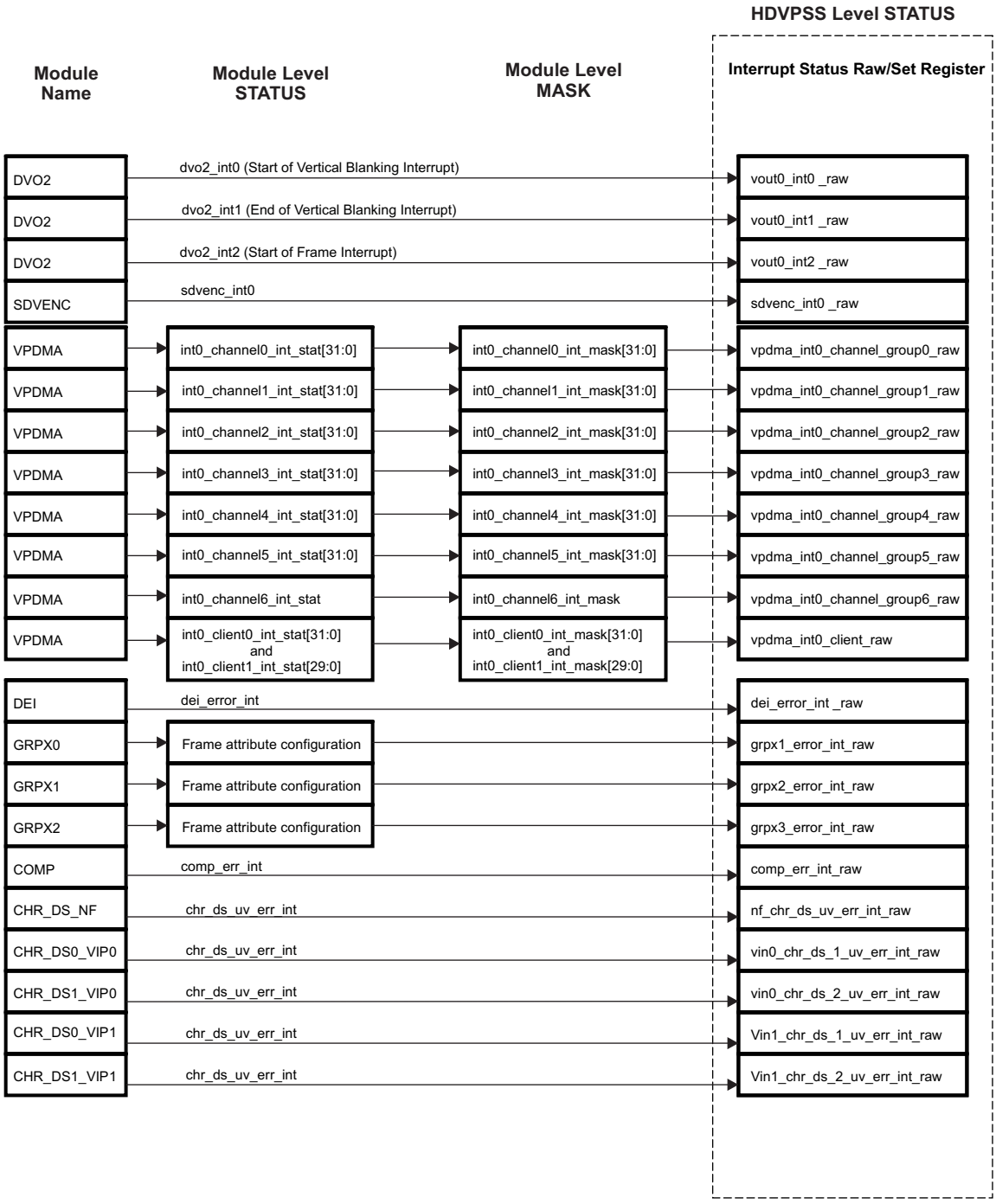
**Figure 12-13. Module to HDVPSS Interrupt Mapping Level 1**






Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-14. Module to HDVPSS Interrupt Mapping Level 2**





Copyright © 2011–2016, Texas Instruments Incorporated

### 12.1.8 Clocking

The following section discusses the steps needed to enable HDVPSS clocking.

#### 12.1.8.1 Enable PRCM for HDVPSS Module

Please refer to the *PRCM* chapter to enable PRCM for HDVPSS.

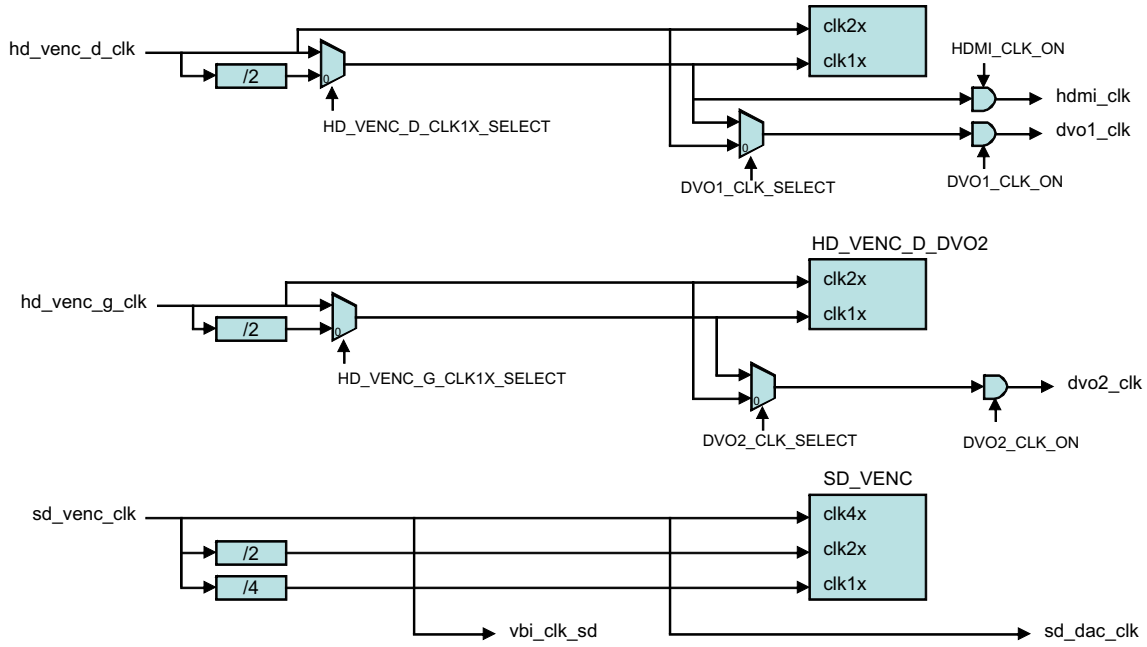
#### 12.1.8.2 Enable HDVPSS Internal Clocks

In HDVPSS, some of the data paths and modules are divided into different clock domains. This enables the user to switch off the paths/modules that are not used, at HDVPSS level. The CLKC→CLKC Module Clock Enable Register (address: 0x48100100) is used to enable the paths that fall in separate clock domains (as mentioned in the CLKC register).

#### 12.1.8.3 VENC Clock Source (HDMI or DVO1/DVO2/SDVENC Pixel Clock)

Figure 12-15 shows the derivation of pixel clock for different VENCs. In this figure, hd_venc_d_clk (external clock source) directly drives the HDMI/DVO1 (hd_venc_d) video encoder and hd_venc_g_clk (external clock source) directly drives the DVO2 (hd_venc_d) video encoder.

**Figure 12-15. Video Encoder Clock/Control Diagram**




The output of the muxes (shown in the above diagram) can be controlled in the CLKC Video Encoder Clock Select (address: 0x48100114) register. Please make sure the required VENCs are enabled in the CLKC Video Encoder Enable (address: 0x48100118) register.


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.1.9 Reset

HDVPSS provides separate resets for each of the paths/modules.

The following sequence is used to reset the entire HDVPSS module:

1. Take VPDMA to standby via force standby MMR write. This disconnects master ports.
2. Ensure all VENCs are disabled via MMR writes.
3. Power down VENCs via appropriate MMR writes.
4. Shut down VENC clocks via MMR writes.
5. Shut down VIP clocks.
6. Shut down all module clocks by writing to all enable bits.
7. Reset all modules and Vencs by writing reset bits.
8. Request IDLE, on IDLE acknowledgement, clocks can be shut down.


Copyright © 2011–2016, Texas Instruments Incorporated

## 12.2 Internal Modules

This section describes the internal modules of the high-definition video processing subsystem (HDVPSS).

### *12.2.1 Chroma Up-Sampler (CHR_US)*

The chroma up-sampler (chr_us) module is used to convert from YCbCr 4:2:0 data format input to YCbCr 4:2:2 format output.

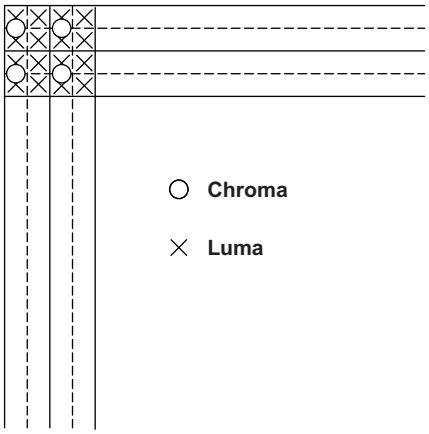
#### 12.2.1.1 Features

- Supports both interlaced and progressive inputs
- 4-tap interpolation filtering
- Filter coefficients are all programmable
- Four sets of coefficients (each set has four coefficients) corresponding to anchor pixels and interpolated pixels of top field and bottom field
- For progressive inputs, the coefficients corresponding to top and bottom field must be identical
- Each coefficient is 14 bit (4.10 format)
- Default filter coefficients are based on Catmull-Rom algorithm
- Capable of removing the half pel vertical offset so all chroma samples are on-grid. This step ensures that the output does not suffer from any kind of rainbow effect due to chroma-upsampling.
- Provides a 10-bit interface in both directions: 10-bit input and 10-bit output
- Support bypass mode for 4:2:2 input

#### 12.2.1.2 Functional Description

The YUV420 input to Chroma Upsampler module must be in the format shown in Figure 12-16, in which the chroma sample lies in the left column of a 2x2 pixel block with half pel vertical shift.

**Figure 12-16. 4:2:0 YCrCb Color Space with Chroma Left-aligned**




The upsampling is performed by an interpolation filter which uses Catmull-Rom algorithm. The Catmull-Rom Filter is based on four anchor pixels representing a four tap filter. The general 4-tap filter Catmull-Rom filter is defined in Figure 12-17.

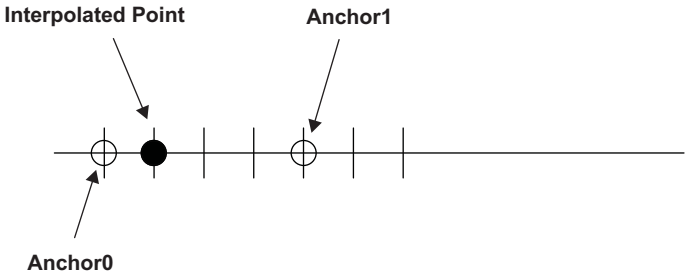

Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-17. Catmull-Rom Filter Definition**

$$[\,1 \quad x \quad x^2 \quad x^3\,] * \begin{bmatrix} 0 & 1 & 0 & 0 \\ -a & 0 & a & 0 \\ 2a & -3+a & 3-2a & -a \\ -a & 2-a & -(2-a) & a \end{bmatrix}$$

In the previous figure, x is the distance to the interpolated point between the two anchor points. In Figure 12-18, the example shows the desired interpolated point to be ¼ of the distance between Anchor0 and Anchor1. Thus, x = ¼.

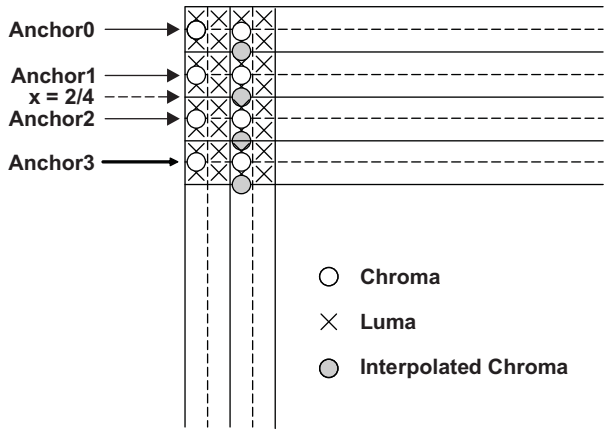
**Figure 12-18. Definition of 'X'**




The variable 'a' determines the characteristics of the filter. a = ½ is generally used because the filter will produce an interpolated output that is an exact match to a linear input curve. In the literature, some people have noted that a=0.75 or a=1.0 may be more pleasing to the eye. In the implementation, the filter coefficients are programmable through MMR.

For the interpolated pixel, the variable x defines the positional offset relative to anchor pixel1. Figure 12-19 shows four anchor pixels with Anchor0 being near the top of the image and Anchor3 near the bottom. x is relative to Anchor1. Positive values of x goes down towards Anchor3. Negative x values imply a direction towards Anchor0. For example, if we want to interpolate a pixel midway between Anchor1 and Anchor2, x would be 2/4=1/2. There are four half pels between Anchor1 and Anchor2. Midway is two pels, so x=2/4.

**Figure 12-19. Anchor Pixels**





Copyright © 2011–2016, Texas Instruments Incorporated

In the implementation, we need to interpolate the anchor pixel to get it on-grid. Then, we also need to interpolate a completely new pixel for YPrPb 4:2:2.

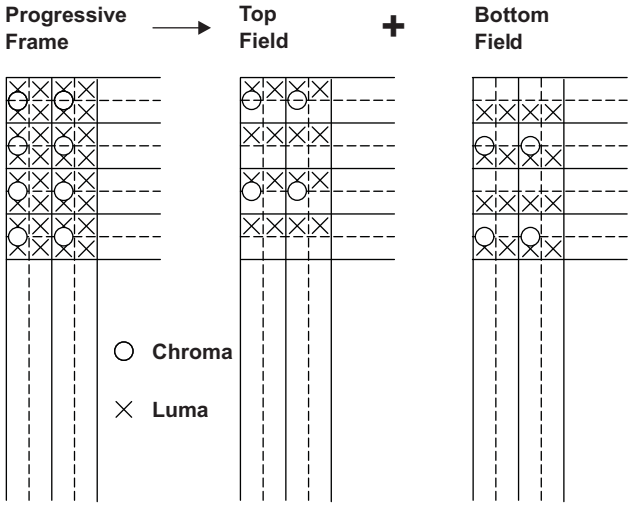
For a progressive input, x = -1/4 for getting the anchor pixel on-grid. x = 1/4 for generating the new pixel.

Lines are scanned from the top of the picture to the bottom.

#### 12.2.1.2.1 For Interlaced YUV420 Input Data

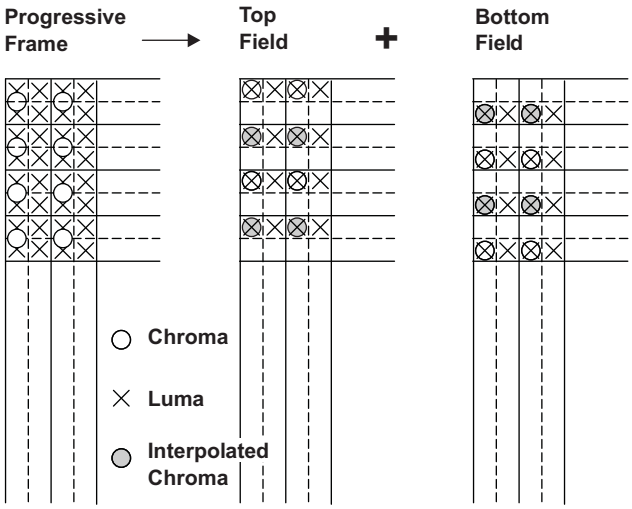
Figure 12-20 shows how 4:2:0 video is split into top and bottom fields in interlaced format. Chroma is attached to alternating Luma lines in the top and bottom fields such that color is equally spread out between the top field and the bottom field. This 4:2:0 interlaced chroma representation is shown in the following figure.

**Figure 12-20. 4:2:0 Interlaced Scan**




In each interlaced field, the chroma anchor points are separated by 4 pixel lines, or 8 half pels. The 4:2:2 chroma interpolated color space representation of interlaced pictures is shown in Figure 12-21.

**Figure 12-21. Ideal 4:2:2 Chroma Upsampling for Interlaced Scan**




Following interpolation, the chroma samples lie on a 4:2:2 grid.


Copyright © 2011–2016, Texas Instruments Incorporated

Anchor pixels for the top field have been sited up by half a pel (x = -1/8). The new interpolated pixel is sited 3 half pels (x=3/8) down. Samples from the top field are distinct from samples in the bottom field.

For the bottom field, anchor pixels are sited 3 half pels up (x = -3/8). The new interpolated pixel is sited 1 half pel down (x=1/8).

The chroma upsampling filter accepts different coefficients for the top field and the bottom field. In the case of progressive input, the coefficients for the top field and bottom field must be the same.

It should be noted that a different implementation could have been chosen to use the same coefficients for the top and bottom fields. Instead of pushing pixels from the top to bottom of a picture, it can be shown that pushing the bottom field through the upsampling filter from the bottom to the top of the picture permits using the same values for x as in the top field case.

#### *12.2.1.2.2 Edge Effects*

Several methods with increasing levels of difficulty resulting in increasing quality can be employed to deal with chroma pixels near the edges. In this module, the edge pixels can be mathematically approximated using the same filter as the rest of the picture. Edge pixels are duplicated going into the filter.

#### *12.2.1.2.3 Modes of Operation (VPDMA)*

In both primary (PRI) and auxiliary (AUX) paths, the mode in which the VPDMA needs to be operated depends whether the chroma upsampler and de-interlacers are enabled or not. Table 12-12 shows the modes of operation.

**Table 12-12. VPDMA Modes of Operation**

| Input data is 4:2:0 | Input data is 4:2:2 |
|---|---|
| Mode A | Mode B |

These modes need to be set in the following register bit-fields of particular instances being used:

CHR_US_reg0.cfg_mode

Line_mode bitfield in VPDMA registers of format RD_LB_CLIENT_CTL_STATUS need to configured as follows:

Mode A corresponds to MMR value 0
Mode B corresponds to MMR value 1

The following are the VPDMA client related bit-fields that need to be configured:

VPDMA_pri_chroma_cstat.line_mode
VPDMA_aux_chroma_cstat.line_mode
VPDMA_sec0_chroma_cstat.line_mode
VPDMA_sec1_chroma_cstat.line_mode
VPDMA_pri_luma_cstat.line_mode
VPDMA_aux_luma_cstat.line_mode
VPDMA_sec0_luma_cstat.line_mode
VPDMA_sec1_luma_cstat.line_mode

**Important Notes:**

1. For VPDMA luma clients, Mode B should always be used.
2. For VPDMA chroma clients, Mode A is used for 420 data and Mode B is used for 422 data.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.1.2.4 Coefficient Configuration

The filter coefficients are left-aligned 14-bit binary values in signed Q4.10 format. The decimal point is between bits 9 and 10 using the convention of the least significant bit being at position zero. The most significant bit, 13, is the sign bit.

In the register map, the most significant nibble of the coefficient is the sign and the integer portion of the value. The next 10 bits represent the fractional portion of the coefficient value.

Chroma upsampling requires two sets of coefficients. Each coefficient set is comprised of four 14-bit Q4.10 values. One set is used for the top field of an interlaced picture, and the other set is used for the bottom field of an interlaced picture. For a progressive picture, both sets must be identical.

The coefficients and settings should be used for the following video source types:

**4:2:2 input (progressive or interlaced input)**

VPDMA line mode = 1

CHR_US_reg0.cfg_mode = 0x1 (mode B)

CHR_US coefficients are not used in this mode, so values are "don't care"

**4:2:0 input (interlaced input):**

VPDMA line mode = 0

CHR_US_reg0.cfg_mode = 0x0 (mode A)
CHR_US_reg0.anchor_fid0_c0 = 0x51
CHR_US_reg0.anchor_fid0_c1 = 0x3d5
CHR_US_reg1.anchor_fid0_c2 = 0x3fe3
CHR_US_reg1.anchor_fid0_c3 = 0x3ff7
CHR_US_reg2.interp_fid0_c0 = 0x3fb5
CHR_US_reg2.interp_fid0_c1 = 0x2e9
CHR_US_reg3.interp_fid0_c2 = 0x18f
CHR_US_reg3.interp_fid0_c3 = 0x3fd3
CHR_US_reg4.anchor_fid1_c0 = 0x16b
CHR_US_reg4.anchor_fid1_c1 = 0x247
CHR_US_reg5.anchor_fid1_c2 = 0xb1
CHR_US_reg5.anchor_fid1_c3 = 0x3f9d
CHR_US_reg6.interp_fid1_c0 = 0x3fcf
CHR_US_reg6.interp_fid1_c1 = 0x3db
CHR_US_reg7.interp_fid1_c2 = 0x5d
CHR_US_reg7.interp_fid1_c3 = 0x3ff9

**4:2:0 input (progressive input):**

VPDMA line mode = 0

CHR_US_reg0.cfg_mode = 0x0 (mode A)
CHR_US_reg0.anchor_fid0_c0 = 0x00C8
CHR_US_reg0.anchor_fid0_c1 = 0x0348
CHR_US_reg1.anchor_fid0_c2 = 0x0018
CHR_US_reg1.anchor_fid0_c3 = 0x3fd8
CHR_US_reg2.interp_fid0_c0 = 0x3fb8
CHR_US_reg2.interp_fid0_c1 = 0x0378
CHR_US_reg3.interp_fid0_c2 = 0x00e8
CHR_US_reg3.interp_fid0_c3 = 0x3fe8
CHR_US_reg4 to CHR_US_reg7 are not used so their values are "don't care".


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.2 Color Space Converter (CSC)

The color space converter (CSC) module is used to convert video data from one color space to another with nine programmable integer multipliers.

#### 12.2.2.1 Features

- All parameters are programmable
- Each parameter is configurable in signed 13-bits
- Support for Bypass Mode

#### 12.2.2.2 Functional Description

The conversion between the different color spaces requires addition and multiplication operations on color and intensity components. The mathematical expression of the conversion can be written as:

```
Y   = A0*R + B0*G + C0*B + D0
Cb  = A1*R + B1*G + C1*B + D1
Cr  = A2*R + B2*G + C2*B + D2
```

Using YUV to RGB conversion as an example: YUV represents one color space and RGB represents another color space. The conversion can be written in the matrix format shown in Figure 12-22.

**Figure 12-22. Matrix Format**

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} A0 & B0 & C0 \\ A1 & B1 & C1 \\ A2 & B2 & C2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} D0 \\ D1 \\ D2 \end{bmatrix}$$

Since HDTV and SDTV have different conversion requirements, both conversions of RGB-to-YCbCr and YCbCr-to-RGB are described. The details of derivations of these matrixes will be given in the following subsections.

##### 12.2.2.2.1 HDTV Application

###### 12.2.2.2.1.1 HDTV Application with Video Data Range

The two equations presented in this section are for the HDTV application. The chromaticity parameters are defined by ITU-R709 standard.

The input video data for these equations should be within the range that is defined for video application.

In an 8-bit system:

- Rd, Gd, Bd, and Yd will be in the range [16-235]
- Cb and Cr will be the range [16-240]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, and Yd will be in the range [64-940]
- Cb and Cr will be in the range [64-960]
- D = 4


Copyright © 2011–2016, Texas Instruments Incorporated

Conversion from RGB to YCbCr:

**Figure 12-23. Conversion from RGB to YCbCr**

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4646 & -0.0468 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

**Figure 12-24. Conversion from YCbCr to RGB**

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5396 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -197 \\ 82 \\ -232 \end{bmatrix} D$$

#### *12.2.2.2.1.2 HDTV Application with Graphics Data Range*

The two equations presented in this section are for the HDTV application with graphics range data input. The main application is for computer graphics display. The chromaticity parameters are defined by ITU-R709 standard.

The input data ranges for these equations are as follows.

In an 8-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be the range [0-255]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be in the range [0-1023]
- D = 4

Conversion from RGB to YCbCr:

**Figure 12-25. Conversion from RGB to YCbCr**

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.1826 & 0.6142 & 0.0620 \\ -0.1006 & -0.3385 & 0.4392 \\ 0.4392 & -0.3990 & -0.0402 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

**Figure 12-26. Conversion from YCbCr to RGB**

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1.1644 & -0.0003 & 1.7927 \\ 1.1644 & -0.2132 & -0.5329 \\ 1.1642 & 2.1125 & -0.0001 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -248 \\ 77 \\ -289 \end{bmatrix} D$$


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.2.2.1.3 Quantized Coefficients for Color Space Converter in HDTV

This section quantizes all the coefficients of color space conversion based on a 10-bit system for HDTV application. These coefficients can be input to the registers of programmable color space converter. Refer to register section for the corresponding register setting.

**Table 12-13. Quantized Coefficients of HDTV Application with Video Data Range**

| Conversion from RGB to YCbCr | | | | Conversion from YCbCr to RGB | | | |
|---|---|---|---|---|---|---|---|
| Coefficient Names | Real Number Format | Quantized Format | Hex Format | Coefficient Names | Real Number Format | Quantized Format | Hex Format |
| A0(13-bit) | 0.2126 | 218 | 0x00DA | A0(13-bit) | 1 | 1024 | 0x0400 |
| B0(13-bit) | 0.7152 | 732 | 0x02DC | B0(13-bit) | 0 | 0 | 0x0000 |
| C0(13-bit) | 0.0722 | 74 | 0x004A | C0(13-bit) | 1.5396 | 1577 | 0x0629 |
| A1(13-bit) | -0.1172 | -120 | 0x1F88 | A1(13-bit) | 1 | 1024 | 0x0400 |
| B1(13-bit) | -0.3942 | -404 | 0x1E6C | B1(13-bit) | -0.1831 | -187 | 0x1F45 |
| C1(13-bit) | 0.5114 | 524 | 0x020C | C1(13-bit) | -0.4577 | -469 | 0x1E2B |
| A2(13-bit) | 0.5114 | 524 | 0x020C | A2(13-bit) | 1 | 1024 | 0x0400 |
| B2(13-bit) | -0.4646 | -476 | 0x1E24 | B2(13-bit) | 1.8142 | 1858 | 0x0742 |
| C2(13-bit) | -0.0468 | -48 | 0x1FD0 | C2(13-bit) | 0 | 0 | 0x0000 |
| D0(12-bit) | 0 | 0 | 0x000 | D0(12-bit) | -197 | -788 | 0xCEC |
| D1(12-bit) | 128 | 512 | 0x200 | D1(12-bit) | 82 | 328 | 0x148 |
| D2(12-bit) | 128 | 512 | 0x200 | D2(12-bit) | -232 | -928 | 0xC60 |

**Table 12-14. Quantized Coefficients of HDTV Application with Graphics Data Range**

| Conversion from RGB to YCbCr | | | | Conversion from YCbCr to RGB | | | |
|---|---|---|---|---|---|---|---|
| Coefficient Names | Real Number Format | Quantized Format | Hex Format | Coefficient Names | Real Number Format | Quantized Format | Hex Format |
| A0(13-bit) | 0.1826 | 187 | 0x00BB | A0(13-bit) | 1.1644 | 1192 | 0x04A8 |
| B0(13-bit) | 0.6142 | 629 | 0x0275 | B0(13-bit) | -0.0003 | 0 | 0x0000 |
| C0(13-bit) | 0.062 | 63 | 0x003F | C0(13-bit) | 1.7927 | 1836 | 0x072C |
| A1(13-bit) | -0.1006 | -103 | 0x1F99 | A1(13-bit) | 1.1644 | 1192 | 0x04A8 |
| B1(13-bit) | -0.3385 | -347 | 0x1EA5 | B1(13-bit) | -0.2132 | -218 | 0x1F26 |
| C1(13-bit) | 0.4392 | 450 | 0x01C2 | C1(13-bit) | -0.5329 | -546 | 0x1DDE |
| A2(13-bit) | 0.4392 | 450 | 0x01C2 | A2(13-bit) | 1.1642 | 1192 | 0x04A8 |
| B2(13-bit) | -0.399 | -409 | 0x1E67 | B2(13-bit) | 2.1125 | 2163 | 0x0873 |
| C2(13-bit) | -0.0402 | -41 | 0x1FD7 | C2(13-bit) | -0.0001 | 0 | 0x0000 |
| D0(12-bit) | 16 | 64 | 0x040 | D0(12-bit) | -248 | -992 | 0xC20 |
| D1(12-bit) | 128 | 512 | 0x200 | D1(12-bit) | 77 | 308 | 0x134 |
| D2(12-bit) | 128 | 512 | 0x200 | D2(12-bit) | -289 | -1156 | 0xB7C |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.2.2.2 SDTV Application

##### 12.2.2.2.2.1 SDTV Application with Video Data Range

The two equations presented in this section are for the SDTV application. The chromaticity parameters are defined by ITU-R601 standard.

The input video data for these equations should be within the range that is defined for video application.

In an 8-bit system:

- Rd, Gd, Bd, and Yd will be in the range [16-235]
- Cb and Cr will be the range [16-240]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, and Yd will be in the range [64-940]
- Cb and Cr will be in the range [64-960]
- D = 4

Conversion from RGB to YCbCr:

**Figure 12-27. Conversion from RGB to YCbCr**

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.172 & -0.339 & 0.511 \\ 0.511 & -0.428 & -0.083 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

**Figure 12-28. Conversion from YCbCr to RGB**

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1 & -0.0003 & 1.3717 \\ 1 & -0.3365 & -0.6984 \\ 1 & 1.7336 & -0.0016 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -176 \\ 132 \\ -222 \end{bmatrix} D$$


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.2.2.2.2 SDTV Application with Graphics Data Range

The two equations presented in this section are for the SDTV application with graphics range data input. The main application is for computer graphics display. The chromaticity parameters are defined by ITU-R601 standard.

The input data ranges for these equations are as following.

In an 8-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be the range [0-255]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be in the range [0-1023]
- D = 4

Conversion from RGB to YCbCr:

**Figure 12-29. Conversion from RGB to YCbCr**

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

**Figure 12-30. Conversion from YCbCr to RGB**

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1.1641 & -0.0018 & 1.5958 \\ 1.1641 & -0.3914 & -0.8135 \\ 1.1641 & 2.0178 & -0.0012 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -223 \\ 136 \\ -277 \end{bmatrix} D$$


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.2.2.2.3 Quantized Coefficients for Color Space Converter in SDTV

This section quantizes all the coefficients of color space conversion based on a 10-bit system for SDTV application. These coefficients can be input to the registers of programmable color space converter. Refer to register section for the corresponding register setting.

**Table 12-15. Quantized Coefficients of SDTV Application with Video Data Range**

| Conversion from RGB to YCbCr | | | | Conversion from YCbCr to RGB | | | |
|---|---|---|---|---|---|---|---|
| Coefficient Names | Real Number Format | Quantized Format | Hex Format | Coefficient Names | Real Number Format | Quantized Format | Hex Format |
| A0(13-bit) | 0.299 | 306 | 0x0132 | A0(13-bit) | 1 | 1024 | 0x0400 |
| B0(13-bit) | 0.587 | 601 | 0x0259 | B0(13-bit) | -0.0003 | 0 | 0x0000 |
| C0(13-bit) | 0.114 | 117 | 0x0075 | C0(13-bit) | 1.3717 | 1405 | 0x057D |
| A1(13-bit) | -0.172 | -176 | 0x1F50 | A1(13-bit) | 1 | 1024 | 0x0400 |
| B1(13-bit) | -0.339 | -347 | 0x1EA5 | B1(13-bit) | -0.3365 | -345 | 0x1EA7 |
| C1(13-bit) | 0.511 | 523 | 0x020B | C1(13-bit) | -0.6984 | -715 | 0x1D35 |
| A2(13-bit) | 0.511 | 523 | 0x020B | A2(13-bit) | 1 | 1024 | 0x0400 |
| B2(13-bit) | -0.428 | -438 | 0x1E4A | B2(13-bit) | 1.7336 | 1775 | 0x06EF |
| C2(13-bit) | -0.083 | -85 | 0x1FAB | C2(13-bit) | -0.0016 | -2 | 0x1FFE |
| D0(12-bit) | 0 | 0 | 0x000 | D0(12-bit) | -176 | -704 | 0xD40 |
| D1(12-bit) | 128 | 512 | 0x200 | D1(12-bit) | 132 | 528 | 0x210 |
| D2(12-bit) | 128 | 512 | 0x200 | D2(12-bit) | -222 | -888 | 0xC88 |

**Table 12-16. Quantized Coefficients of SDTV Application with Graphics Data Range**

| Conversion from RGB to YCbCr | | | | Conversion from YCbCr to RGB | | | |
|---|---|---|---|---|---|---|---|
| Coefficient Names | Real Number Format | Quantized Format | Hex Format | Coefficient Names | Real Number Format | Quantized Format | Hex Format |
| A0(13-bit) | 0.257 | 263 | 0x0107 | A0(13-bit) | 1.1641 | 1192 | 0x04A8 |
| B0(13-bit) | 0.504 | 516 | 0x0204 | B0(13-bit) | -0.0018 | -2 | 0x1FFE |
| C0(13-bit) | 0.098 | 100 | 0x0064 | C0(13-bit) | 1.5958 | 1634 | 0x0662 |
| A1(13-bit) | -0.148 | -152 | 0x1F68 | A1(13-bit) | 1.1641 | 1192 | 0x04A8 |
| B1(13-bit) | -0.291 | -298 | 0x1ED6 | B1(13-bit) | -0.3914 | -401 | 0x1E6F |
| C1(13-bit) | 0.439 | 450 | 0x01C2 | C1(13-bit) | -0.8135 | -833 | 0x1CBF |
| A2(13-bit) | 0.439 | 450 | 0x01C2 | A2(13-bit) | 1.1641 | 1192 | 0x04A8 |
| B2(13-bit) | -0.368 | -377 | 0x1E87 | B2(13-bit) | 2.0178 | 2066 | 0x0812 |
| C2(13-bit) | -0.071 | -73 | 0x1FB7 | C2(13-bit) | -0.0012 | -1 | 0x1FFF |
| D0(12-bit) | 16 | 64 | 0x040 | D0(12-bit) | -223 | -892 | 0xC84 |
| D1(12-bit) | 128 | 512 | 0x200 | D1(12-bit) | 136 | 544 | 0x220 |
| D2(12-bit) | 128 | 512 | 0x200 | D2(12-bit) | -277 | -1108 | 0xBAC |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.2.3 Bypass Mode

CSC module can be bypassed by setting the following bit-field to 1:

CSC→CSC05.Bypass

### *12.2.3 Compositor Module*

The Compositor (COMP) module is used to blend the input video and graphics sources into a single stream to drive video encoders.

#### 12.2.3.1 Features Supported

COMP supports the following features:

- Three displays: 2 HDs and 1 SD
- Up to five input layers for each HD display: two videos and three graphics
- Up to four input layers for the SD display: one video and three graphics
- Cascade alpha blending for two input layers first and graphic blending later
- Blended outputs to memory from two of the blenders

#### 12.2.3.2 Functional Description

The COMP module has three independently controlled blenders (compositors): two for HD output and one for SD output. Each HD blender can take up to five input layers (two videos and three graphics) to generate one composite output layer. The SD blender can take up to four input layers (one video and three graphics) to generate one composite output layer. This is shown in Figure 12-31.

Each blender is associated with one of the three VENCs to drive the display data as shown.

Blender0 → HDMI/DVO1

Blender1 → DVO2

Blender2 → SD

Following are the input layers available:

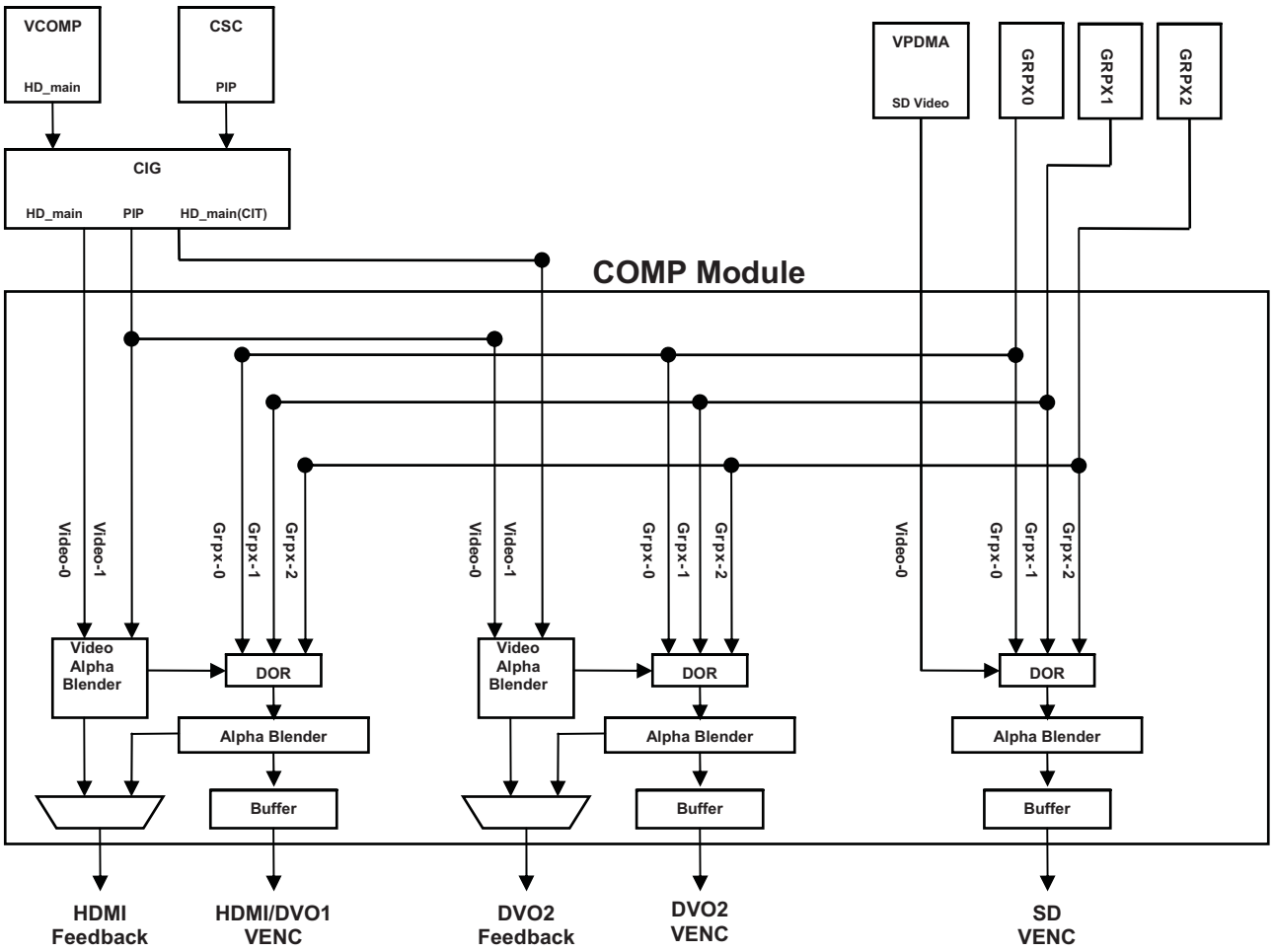
- HD - Main Video Input
- HD - Main Video Input constrained (CIT)
- HD - PIP Video Input
- SD Video Input
- Graphics0 input (GRPX-0)
- Graphics1 input (GRPX-1)
- Graphics2 input (GRPX-2)

Each of the input layers can be associated to one or more VENCs in HD-VPSS as shown in the above figure. All the input layers associated with a VENC should have the same size and type (progressive or interlaced) of the display. This implies that if an input layer is shared with multiple VENCs, all those VENCs should have the same size, type and frame rate.

For example, if the 1080p HDMI output is desired to have main video input and PIP video input with 1 graphics, then the main and PIP video input must be 1080p, and one of the graphics must be 1080p. If another blender is setup 480i, it cannot use either the main or PIP video inputs, and must use a different graphics input.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-31. COMP Module Block Diagram**




All graphic sources have both interlaced or progressive formats. Muxes are used for each display to choose the right format. If the graphics module has interlaced input, it can only provide interlaced output through progressive data path. Therefore, the interlaced input data path can not be used in this case.

Input layers associated to a VENC are grouped together, reordered according to the user programmable display priority order list, and blended using embedded alpha values between overlapped layers. Each composed output is then buffered in a memory (inside COMP) from which the associated VENC pulls the data out at its pixel clock rate.

As mentioned above, each blender can take up to five input layers (four for SD) out of the available seven input layers. The accessibility of each blender to the input layer is shown in Table 12-17.

**Table 12-17. Input Layers and Associated Blenders**

| Input layers and Blenders | Blender0 | Blender1 | Blender2 |
|---|---|---|---|
| HD main video | YES | NO | NO |
| HD main video (CIT) | NO | YES | NO |
| SD video | NO | NO | YES |
| HD pip video | YES | YES | NO |
| Grpx-0 | YES | YES | YES |
| Grpx-1 | YES | YES | YES |
| Grpx-2 | YES | YES | YES |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.3.3 Input Sources

Each blender in COMP module receives up to five input sources (four for SD) to blend and send the composited output to the corresponding VENC. Five input sources to a blender are as follows :

1. Video-0 (HD main video/ HD main video CIT/ SD video)
2. Video-1 (HD pip video; Not available for SD output)
3. Grpx-0
4. Grpx-1
5. Grpx-2

All the inputs that are connected to a single blender should:

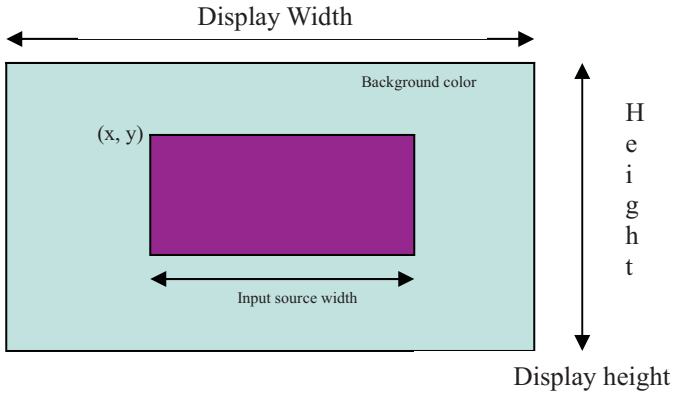
- Have the same size of its corresponding VENC display size. This is done by forming a WINDOW (shown below) in the upstream modules. Please note that it does NOT imply any restriction on the actual input video sizes in external memory to be the same. For example, Video-0 can be 1920x1080 and Video-1 can be 720x480 in the memory.
- Carry their priority levels (if g_reorder =0) to re-order themselves as layers on the final display stream (only for graphic inputs) and
- Carry alpha values used to blend with each other using the above priority levels.

It is necessary to configure upstream modules (to blender) to supply each input of full display size to the blender. If the size of any window is not the same as the display size, background color is inserted in the remaining portion as shown below. The upstream modules include:

1. VCOMP (for HD main video - CIT and non-CIT)
2. CIG (for HD PIP video)
3. VPDMA SD Video (for SD Video)
4. GRPX-0 (for Grpx-0 input)
5. GRPX-1 (for Grpx-1 input)
6. GRPX-2 (for Grpx-2 input)

Display width and display height parameters come from VENC module settings, as shown in Figure 12-32.

**Figure 12-32. Display Width and Height Parameters**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.3.3.1 HD Main Video Window (CIT or non_CIT)

This window is formed in the VCOMP module with two input sources: main and auxiliary. The composition of the input sources and background color is explained in VCOMP module functional description.

VCOMP→reg6. cfg_dsply_numlines = CIG→CIG_REG1.disp_h = Display_Height of VENC associated with HD main video

VCOMP→reg6. cfg_dsply_numpix_per_line = CIG→CIG_REG1.disp_w = Display_Width of VENC associated with HD main video

Background color for HD main window is selected using following bitfields:

VCOMP→reg9.cfg_dsply_bckgrnd_cr_val

VCOMP→reg9.cfg_dsply_bckgrnd_cb_val

VCOMP→reg9.cfg_dsply_bckgrnd_y_val

The alpha value of HD PIP window is configured using following bit-fields:

CIG→CIG_REG2.bl_enable

CIG→CIG_REG2.bl_level

The layer order of HD video windows (main and pip) is configured in COMP <VENC> settings. The <VENC>_vid_bld_ord bitfield is described in the video alpha blender section.

#### 12.2.3.3.2 HD PIP Video Window

CIG module forms the PIP video window after receiving PIP video source from CSC module

CIG→CIG_REG6.pip_disp_h = Display_Height of VENC associated with PIP video

CIG→CIG_REG6.pip_disp_w = Display_Width of VENC associated with PIP video

Background color for PIP window is transparent and carries an alpha value of zero. It gets the background color set by COMP→back_clr at the time of composition.

The alpha value of HD PIP window is configured using following bit-fields:

CIG→CIG_REG9.bl_enable

CIG→CIG_REG9.bl_level

The priority of HD video windows (both the main and PIP windows) is described in Section 12.2.3.4.1.

#### 12.2.3.3.3 SD Video Window

For the SD video window, the SD video input from the memory needs to be the same as the Display_Width of SD_VENC output display.

For the SD video window:

Input source width = Display_Width

Input source height = Display_Height

x=0

y=0

and hence no background color

The alpha value for SD video is 0xFF.

The priority level used SD video is set in: COMP→SD_setting.vid_order (irrespective of COMP→SD_setting.g_reorder bit-field value).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.3.3.4 Graphics Windows

These windows are formed in the corresponding GPRX modules by configuring below:

GRPX→frame_display_attributes.frame_height = Display_Height of VENC associated with corresponding Grpx

GRPX→frame_display_attributes.frame_width = Display_Width of VENC associated with corresponding Grpx

The origin of a Grpx region input in the window (x,y) is configured as below:

GRPX→region_display_attributes.disp_pos_x = x

GRPX→region_display_attributes.disp_pos_y = y

Background color for Grpx window is transparent and carries an alpha value of zero. It gets the background color set by COMP→back_clr at the time of composition.

The alpha value configuration for blending is explained in GRPX module. Blend type for a region is selected using following bitfield:

GRPX→region_display_attributes.Disp_blend_cfg

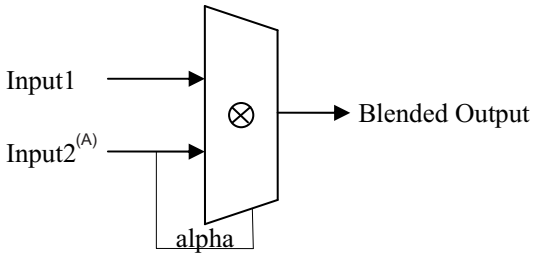
The priority level of a graphics region is set, in corresponding region, using the bitfield mentioned below:

GRPX→region_display_attributes.Disp_prior

### 12.2.3.4 Blender

A Blender in the COMP module works as mentioned in Figure 12-33:

**Figure 12-33. Blender Diagram**




A Input2 is tied to 0 for the SD Blender.

When two input sources, say input1 and input2, are blended, the resulting source is calculated as follows:

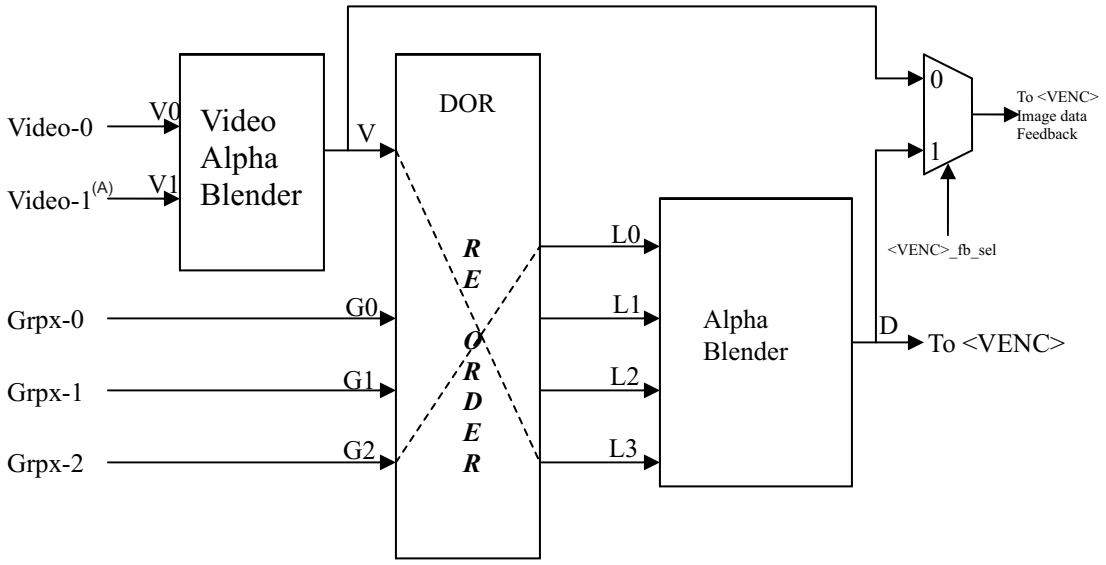
| | | |
|---|---|---|
| Blended Output | =(a)*input1+ (1-a)*input2 | if input1 is top layer and input2 is bottom layer |
| | =(1-b)*input1+ (b)*input2 | if input2 is top layer and input1 is bottom layer |


Copyright © 2011–2016, Texas Instruments Incorporated

Where a and b are alpha values of input1 and input2 windows respectively. The layer order (top or bottom) is decided by priority levels of the two video inputs.

The blending and re-ordering in each blender happens as shown in Figure 12-34

**Figure 12-34. Blending and Reordering**




A Video-1 is tied to 0 for the SD Blender.

#### 12.2.3.4.1 Video Alpha Blender

Two video inputs, Video-0 and Video-1 (sometimes known as MAIN and PIP windows) are first blended together to generate a single video output as shown in the figure above. The priority level, which determines which video to be on top, is decided by COMP→<VENC> settings. <VENC>_vid.bld_ord bitfield.

The alpha values for both the videos are configured in CIG module as described in the sections above.

The alpha value (blended alpha) of the blended video output is calculated as follows.

| | | |
|---|---|---|
| Blended alpha | =alpha_bottom video layer | if alpha_top video layer < 0xFF |
| | =alpha_top video layer | if alpha_top video layer = 0xFF |

Blended alpha (calculated above) is used as alpha value for the Video input to the DOR module described below:

The priority of blended video is configured in COMP→<VENC>Settings.vid_order fit field.

#### 12.2.3.4.2 Display Order Re-shuffling (DOR)

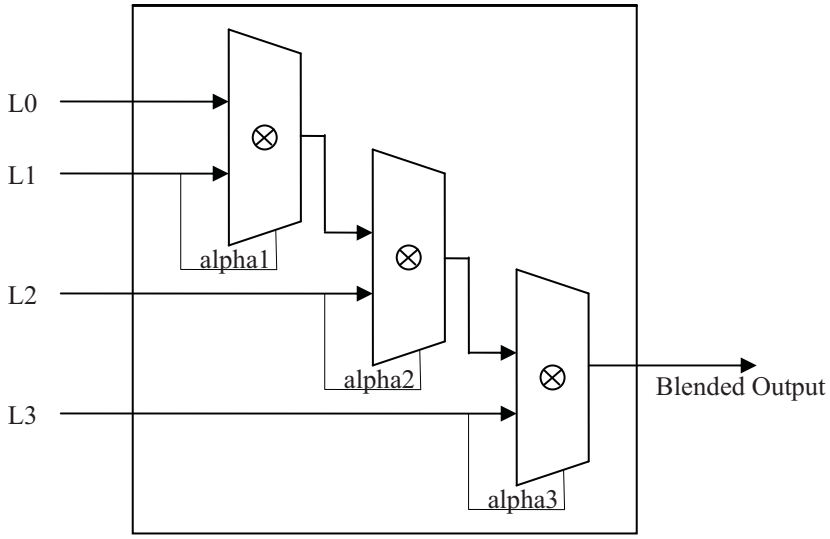
The DOR module is a fully programmable crossbar switch to allow flexible display order change for all input layers. Video layer data is placed in the layer based on its register setting (vid_order bit-field). The graphic data comes with a four-bit priority level configured in the GRPX module. If the global re-order is set (g_reorder=1), the graphics layers are reordered (from top to bottom) based on COMP register settings (g0_order, g1_order, g2_order bit-fields). Otherwise, graphics layers are reordered based on GRPX register (priority) settings as mentioned in the above sections.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.3.4.3 Alpha Blender

The Alpha Blender uses the alpha values of the input layers and blends all of them as per the input order. The block diagram of alpha blender is shown in Figure 12-35.

**Figure 12-35. Alpha Blender Block Diagram**




In Figure 12-35, L0 is the top layer and L3 is the bottom layer. alpha1, alpha2, alpha3 are the blending levels of layers L1, L2, and L3 respectively.

Note: The total number of pixels per frame/field of any source driving the COMP module should be even.

### 12.2.3.5 Feedback Output

Blenders 0 and 2 (associated with DVO1 and DVO2) have a feedback output from blended image data. The feedback image data can be selected from video alpha blending or final alpha blending, and sent to external memory. Since feedback path stalls also stall the entire data pipeline, the amount of the stalls in this path can not be such that they will cause an under-flow to the VENC. Hence, the feedback path needs to be given proper high priority in the system for memory access when the corresponding VENC is also functional. Both the paths (feedback path and Venc data path) are optional.

### 12.2.3.6 Background Color

There is a programmable background color in COMP module. Background color is configured in RGB format with 10-bit in each R, G, and B. Any pixel with an alpha value of 0 will be replaced by this color. If the video bottom layer is disabled, the video top layer will be blended with background color. If both video layers are disabled, the background color is used as output of video alpha blending. For SD channel, only one video stream comes in and it will be blended with the background color.

If the COMP is disabled, or the channel within the COMP driving a particular VENC is disabled, or all inputs to a channel driving particular VENC are disabled, the underflow condition occurs at the VENC. This underflow condition is due to data is not being driven out of the COMP fast enough to keep up with the VENC read rate. Underflow can also happen due to lack of sufficient memory bandwidth or improper VPDMA configuration for any of the input channel. When underflow condition happens, the background color programmed in the COMP module is sent for the underflowed pixels.


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.4 Constrained Image Generator (CIG)

The Constrained Image Generator (CIG) module is used to generate a resolution constrained version of the high-quality video to meet the CableCard Copy Protection Specification`s requirement on the handling of Constrained Image Trigger (CIT) information.

#### 12.2.4.1 Features

- Generates both constrained and non-constrained output from a single input video source
  - 1/4x vertical/horizontal resolution reduction using 3-tap vertical/horizontal decimation filters
  - 2x vertical/horizontal resolution restoration using 2-tap average interpolation
  - Video Interlacing to match final scan format
- PIP window framer: Positions a PIP window onto a full display screen.
- Individual Chroma keying and blend value application for all outputs
- Can generate interlaced output on both PIP and main window from a progressive input

#### 12.2.4.2 Functional Description

Copy control information (CCI) is passed from the video service provider across the data channel to inform the Set Top Box of the level of copy protection required. Product shall be able to constrain, when required by the CIT CCI bit, the resolution of content that is High Definition to be output through a connection capable of transmitting content in High Definition Analogue Form, to a Constrained Image. The basic requirement of a constrained image is that it contains the visual equivalent of not more than 520,000 Pixels per frame (e.g. an image with resolution of 540 vertical lines by 960 horizontal lines for a 16:9 aspect ratio). A Constrained Image can be output or displayed using video processing techniques such as line doubling or sharpening to improve the perceived quality of the image.

Although not specifically mentioned in these sources, it is also generally accepted that the degraded video will still need to look better than standard DVDs, which generally are encoded at 720x480.

The CIG module takes in a single non-constrained HD video and generates following two outputs:

- The same non-constrained video which may optionally be interlaced
- The same or constrained version of the source video which may optionally be interlaced

The first output is sent to the DVO1 (HDMI)/DVO2 digital output (via COMP/HD_VENC_D) and the second output is sent to the digital HD output HD_DVO2 (via COMP/HD_VENC_A).

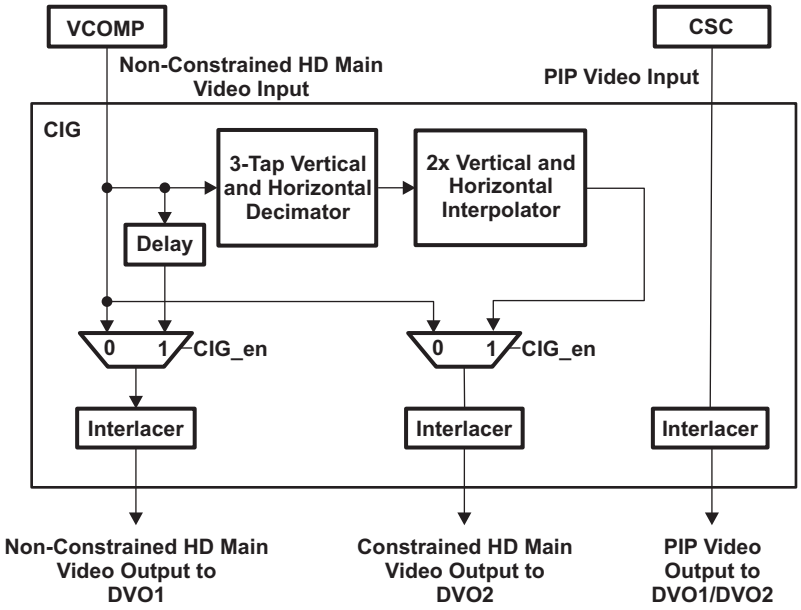
In addition, the CIG module takes in a second video input and positions the video in a full display output screen if the input video is a PIP sized. The PIP video output from CIG module can optionally be interlaced. This path is enabled for following two use cases:

- Displaying un-constrained PIP with constrained/unconstrained main Video (PIP mode)
- Displaying aux video channel output directly on the HD_DVO2 output (Dual-Video Channel mode)

The block diagram of CIG module is shown in Figure 12-36.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-36. CIG Block Diagram**




#### *12.2.4.2.1 Constrained Video Mode*

When the image constraining is required, the CIG requires the incoming unconstrained video to be in progressive scan format in order to apply vertical decimation filter properly. If the output is required to be in interlaced format, the CIG can optionally interlace the outgoing video on both outputs. The PIP video output format (interlaced or progressive) should match the main video output format if it is used as a PIP over the main video.

For the non-constrained video, the output is simply a delayed version (same delay as that of constrained video) of the original video. For the constrained video, ½ decimation filters (3-tap) and 2x interpolation filters (2-tap average) are used to first reduce the overall source resolution by ¼ (½ in vertical ½ in horizontal direction) and to later recover the original display size.

The CIG module performs ½ decimation in both vertical and horizontal directions to achieve the resolution reduction by ¼th (½ in vertical ½ in horizontal direction) on the constrained output and 2x interpolation to restore the total pixel counts. To minimize further filtering of previously optimized image data, a simple 3-taps half-band-filter (¼, ½, ¼) is used for both vertical and horizontal decimation filters while a simple average filter is used for interpolation.

#### *12.2.4.2.2 Interlacer*

The CIG module supports interlacing on each output to generate an interlaced output from the progressive input if needed. The interlacing is performed simply by sending only the current field lines and discarding the others. The current field id is passed on from the upstream module (CSC for HD main video and CSC for PIP video).

#### *12.2.4.2.3 Alpha assignment for Chroma Keying (Transparency) and Blending*

The CIG module receives RGB data and assigns an alpha value to each pixel data based on programmable transparency and blending configurations. When enabled, the blender block in the COMP module uses the configured 8-bit alpha value for both HD main and PIP video windows. Also, it will force a pixel's alpha value to be 0 if the transparency mode is enabled and pixel's color is same as the user-configured transparency color value. A bit masking feature is provided to ignore bottom 1-3 bits when making comparison for transparency color value. The blending alpha value configured in CIG module is carried onto COMP module.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.4.2.4 PIP Video Path

The CIG module includes PIP windowing logic to support composition of an un-constrained PIP window with the constrained video. (Note that if the VCOMP is used to composite two video sources, the Constrain filter is applied to both main and pip video which may not be desired.)

This path can also be used as a non-PIP datapath logic to simply route a second video source to the COMP module.

The PIP path supports independent PIP_DISP_W and PIP_DISP_H to set the output screen size. PIP_W and PIP_H are used to configure the size of the PIP window. When configured as a “fullsize”, then the incoming video must be same as this output size. i.e PIP_DISP_W = PIP_W and PIP_DISP_H = PIP_H. When configured as a non-fullsize, then PIP_X and PIP_Y will position the incoming video onto the output screen. PIP_X, PIP_Y = (0,0) means the PIP window is positioned at top left corner. Note that PIP_X + PIP_W must be less than or equal to PIP_DISP_W. Similiarly, PIP_Y + PIP_H must be less than or equal to PIP_DISP_H.

As mentioned, output interlacing is also supported on this path just in case the incoming video is in progressive scan mode but it needs to be fed into interlaced output. Note that all W/H/X/Y configurations need to be in FRAME mode.

#### 12.2.4.2.5 CIG Bypass Mode

When the image constraining is not required (CIT bit in the encoded bitstream is not set or the application does not require image constraining), the image constraining should be disabled and the input field or frame data should simply be bypassed to the output as shown in Figure 12-36.

To enable the bypass mode:

CIG->CIG_REG0.cig_en = 0

CIG->CIG_REG1 register needs to be configured with correct HD video image size.

Note that the chroma keying and alpha assignment for blending feature as well as output interlacing are still available even in the bypass mode.



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.5 Graphics Module (GRPX)

The graphics module (GRPX) is a region-based graphics processor that composes one or more graphics regions to create a display plane input for the video compositor (COMP) module. For this device, only one region application is supported.

#### 12.2.5.1 Features

- Input:
  - Supports interlace or progressive input format
  - Supports several input formats like RGB 32/24/16-bit, bitmap data (8/4/2/1-bit) with configurable CLUT
  - Supports optional stenciling (1-bit mask) table for each region
- Output:
  - Output data format is ARGB32
  - Provides both progressive and interlaced outputs through different ports simultaneously when input source is progressive
- Multi-region support. Each region can have:
  - Width, height, placement (X and Y) parameters
  - Display priority
  - Scalar enable/disable
  - Scaling ratio attributes
  - Stenciling enable/disable
  - Support for global color and pixel level blending
  - Configurable chroma keying options
  - Bounding box support to reduce artifacts at region boundaries

#### 12.2.5.2 Functional Description

GRPX module reads a RGB or bitmap image data and applies the frame/region attributes to create a graphics plane for the COMP module. The block diagram of GRPX is shown in Figure 12-37.

Two output ports, progressive and interlaced, are available (as shown in Figure 12-37) at the output of GRPX module to output both progressive and interlaced output data to COMP module when the input source is progressive. If the input source is interlaced format, interlaced output is available on both the ports.

One example of the output frame of GRPX module with three regions is shown in Figure 12-38.

As shown in Figure 12-38, a frame is typically composited by rectangular boxes called as regions. Any two regions should not overlap in vertical direction. There is no limitation on number of regions that can be supported in a frame.

The GRPX module is configured in 128-bit words which are logically divided into three groups:

1. Frame configuration attribute – configures display format/sizes (required during initialization)
2. Region configuration attribute – configures all region specific attributes (required on each frame display regions)
3. Scalar configuration attributes (13 words) – configure scaler specific attributes (only required if scaling is enabled)


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-37. GRPX Functional Block Diagram**

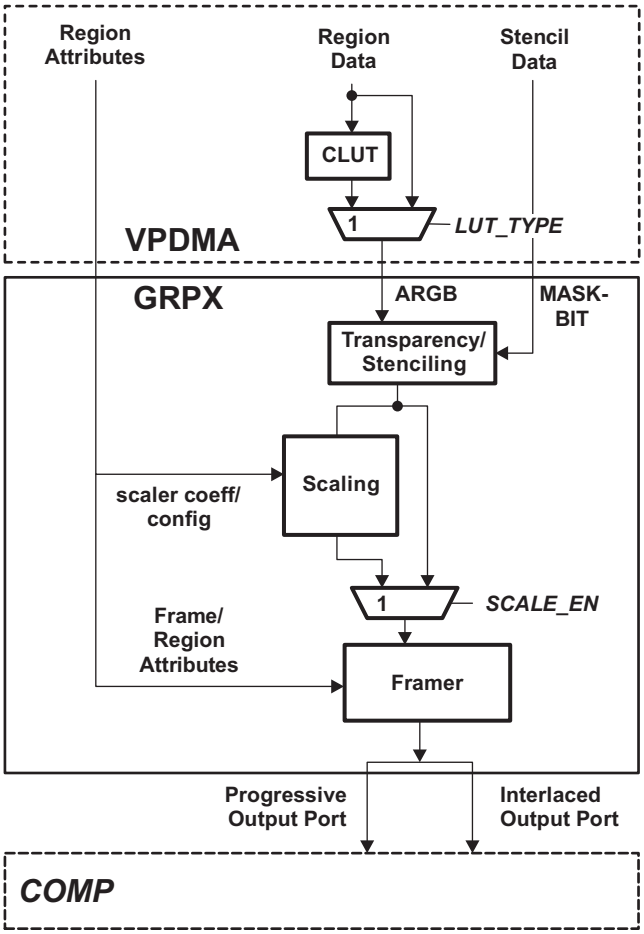
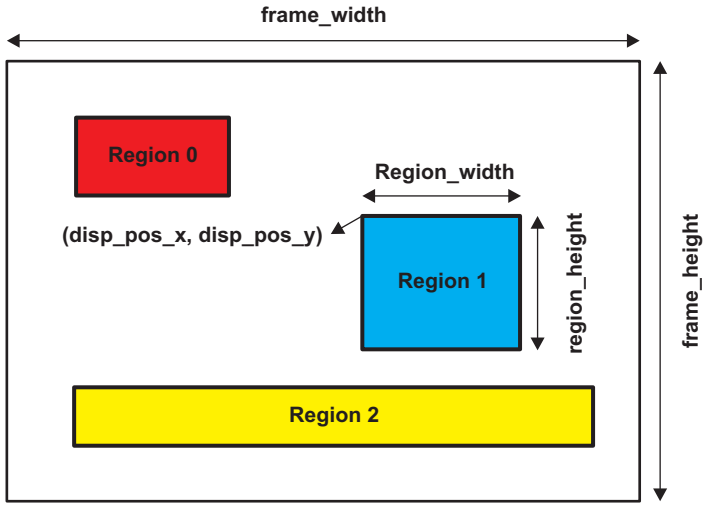



**Figure 12-38. GRPX Output**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.5.2.1 Frame Configuration Attribute

To enable the graphics output plane, a frame configuration attribute has to be written first (typically during the initialization by using a VPDMA configuration descriptor) to set the frame display size (width, height and input data format).

The size of the frame from GRPX module should match the VENC display output size to which the GRPX output is redirected by COMP module. For example, if GRPX0 output is used in DVO2 display output then GRPX0 frame configuration parameters should be same as DVO2 display output frame size.

**Table 12-18. Frame Configuration Attribute (Configuration descriptor payload word 0-3)**

| Attribute | Bits | Description |
|---|---|---|
| Reserved | 31-0 | Reserved |
| frame_height | 47-32 | Frame height (default: 0x0, max 0x7FF)<br>If src_fmt_interlaced is set, frame_height is field_height. (for example, for 480i it should be set to 240).<br>Frame height should be an even number to ensure interlaced outputs are same in height for both fields. |
| frame_width | 63-48 | Frame width (default: 0x0, max 0x780) |
| Reserved | 125-64 | Reserved |
| src_fmt_interlaced | 126 | Format of the source image:<br>0 = Progressive (default)<br>1 = Interlaced |
| Soft reset | 127 | Software reset of GRPX data pipeline:<br>Writing 1 resets GRPX data pipeline<br>Returns 0 on read always |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.5.2.2 Region Configuration Attribute

To display a region, region attribute parameters are set up using the client specific attributes field in the VPDMA data inbound descriptor. Below table shows the details of these parameters.

**Table 12-19. Region Configuration Attribute (Inbound data descriptor word 4-7)**

| Attribute | Bits | Description |
|---|---|---|
| Region_height | 15-0 | Height of the region (number of input region data lines) (max: 0x7FF) |
| Region_width | 31-16 | Width of the region (max: 0x780) |
| disp_pos_y | 47-32 | Position of region (Y origin) (default: 0x00) |
| disp_pos_x | 63-48 | Position of region (X origin) (default: 0x00) |
| Disp_prior | 67-64 | Region display priority |
| Reserved | 70-68 | Reserved |
| Disp_first_region | 71 | 1: First region of the frame |
| Disp_last_region | 72 | 1: First region of the frame |
| scaler_enable | 73 | Scaler enable |
| Reserved | 74 | Reserved |
| stenciling_en | 75 | Stenciling enable for the region |
| Reserved | 78-76 | Reserved |
| bound_box_en | 79 | 0: Bounding box disable<br>1: Bounding box enable |
| bb_alpha | 87-80 | Replace bounding box alpha value when bounding box is enabled. Default value is 0x80. |
| Disp_blend_cfg | 97-88 | Blending configuration<br>95-88: Blending (alpha) value<br>97-96: Blending Type:<br>0 = No blending<br>1 = Region global blending<br>2 = Color (palette) blending<br>3 = Pixel (embedded alpha) blending |
| disp_trans_cfg | 127-101 | Transparency (color keying enable) configuration bits<br>103-102: Transparency LSB bit masking<br>00 = No masking<br>01 = Mask[0] during pixel data comparison to TRANS_COLOR<br>10 = Mask[1:0]<br>11 = Mask[2:0]<br>Example: If TR_LSB_MASK is set to 2'b11, top 5 bits of R/G/B component data are compared to the corresponding 5 bits of TRANS_COLOR R/G/B color.<br>101: Transparency enable<br>0 = Disable<br>1 = Enable transparency for video layer<br>127-104: Transparency color (RGB24) |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.5.2.2.1 Parameter Configuration

Region_height, Region_width, disp_pos_y, disp_pos_x are configured as shown in Figure 12-38.

##### 12.2.5.2.2.1.1 Disp_prior

Each region can be specified to have an inter-graphics-pipeline display priority level which may be used to switch the order between two overlapping regions from two GRPX pipelines. The GRPX simply passes these values along with each pixel data to the COMP module and the COMP uses it to reorder between two GRPX data. This should be used only when both GRPX outputs are tied to the same display output format and rate. The blended output is calculated differently (in the COMP module) for each region based on each its priority level.

##### 12.2.5.2.2.1.2 Disp_first_region

This bit needs to be set to `1` if the region is the first in vertical direction from top of the frame. Region0 in Figure 12-38 is the first region in the frame.

##### 12.2.5.2.2.1.3 Disp_last_region

This bit needs to be set to `1` if the region is the last in vertical direction from top of the frame. Region2 in Figure 12-38 is the last region in the frame.

##### 12.2.5.2.2.1.4 Scaler

The scaler used in GRPX module is consisted of a 5-tap/8-phase horizontal and a 4-tap/8-phase vertical polyphase filters. It supports 0.25x – 4x scaling with a step size of 0.01. Individual regions can have a different set of scaling parameters (coefficients, scale factor, and offset).

The input source format should be progressive if scaling of region is required.

If a region scaling is enabled, the first pixel location (origin) of the region serves as an anchor point for the region display. Since the width and height of the region change after the scaling, application should take care of the correct scaler attributes such that output of the scaler does not go out of the frame display.

Since the scaler stores one or more input lines before outputting any output lines, the following constraints are imposed when multiple regions are displayed on the screen and scaling is enable:

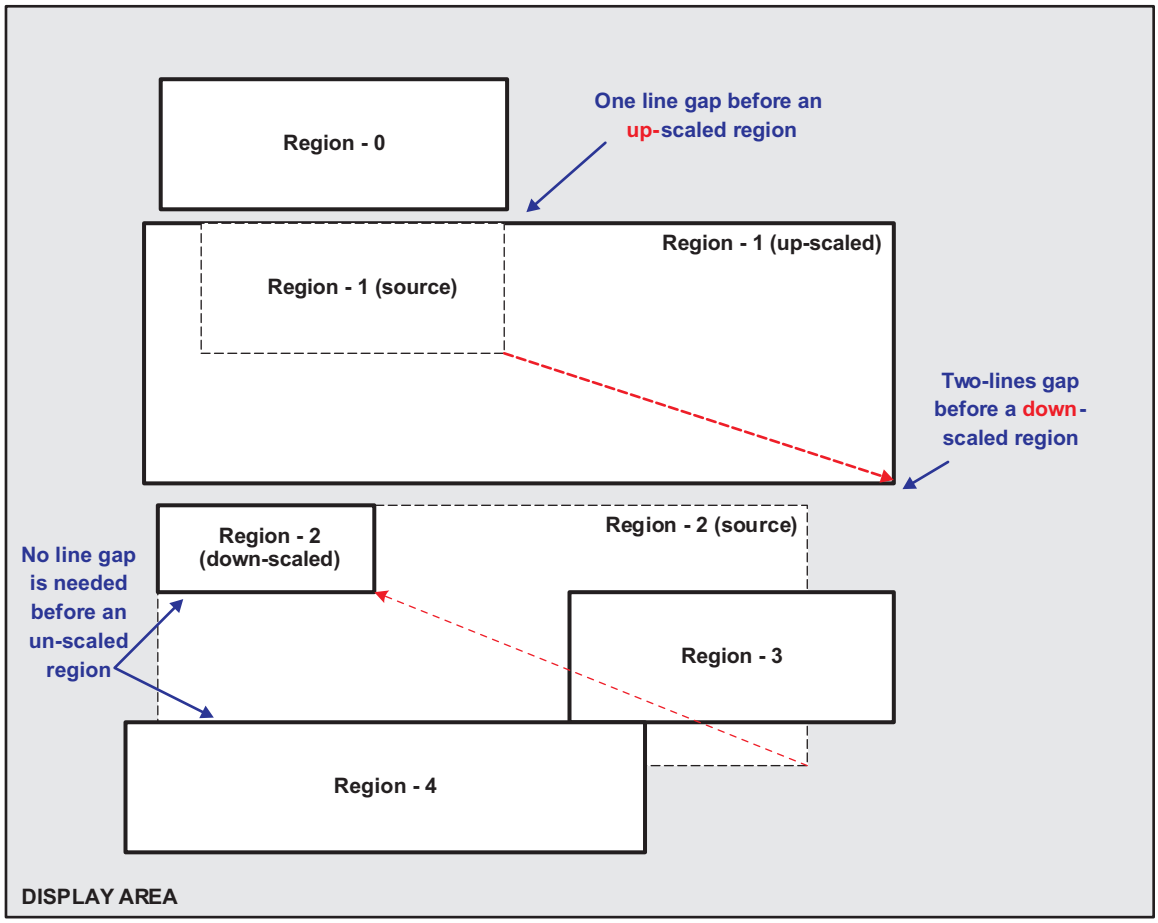
1. One display line gap is required before a vertically up-scaled region
2. Two display line gap is required before a vertically down-scaled region

These constraints do not apply to the first region on the screen. This is illustrated in Figure 12-39.

The minimum size of vertical region is one line, and no multiple regions on a same line is allowed (nor is possible) per “region” graphics definition.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-39. Region Display Position and Gap Requirement Illustration**




Scaler configuration attributes (see Section 12.2.5.2.3) are to be filled for a given scaling ratio when scaler is enabled.

#### *12.2.5.2.2.1.4.1 Anti-flicker Filter Implementation*

When a region is scaled up or down, the poly-phase vertical filter will inherently provide anti-flicker filtering. Even when a region is not needed to be scaled, the scaler can still be enabled to perform low-pass filtering along the vertical direction with scaling ratio=1. Similarly, when the target display of the GRPX is in an interlaced scan mode, the scaler can be used to perform anti-flicker filtering.

#### *12.2.5.2.2.1.5 Stenciling*

The VPDMA sends 1-bit for each pixel (from the region stenciling data buffer) of a region that has “stenciling” feature enabled. This bit is used to force the alpha value of the pixel to ZERO in order to “mask”-off the pixel (i.e., make the pixel transparent). This feature can be used by application to assign an arbitrary mask shape to a rectangular region graphics. Enabling of this feature requires “stenciling_en” bit-field to be set to `1`.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.5.2.2.1.6 Boundary Box Blending

Even with anti-flickering, the flicker around the top and bottom edges of a region may still exists when the anti-flickered region is composed onto the output display in the COMP module. The GRPX supports overwriting alpha values of pixels that make up a 1-pixel wide boundary box of a region with a semi-transparent (alpha) value (default – 0x80) so that the flickering around the edges can be minimized. This feature is enabled when bound_box_en is set to `1`. The alpha value for the boundary box is programmable in the bit-field bb_alpha.

### 12.2.5.2.3 Blending/Transparency Handling

The GRPX supports four blending modes (no-blending, global, pixel/CLUT, pixel) and a transparency mode for each region to determine how alpha value for each pixel is set. The configuration for blending and transparency are in the region attributes and are shown again in Table 12-20.

**Table 12-20. Blending and Transparency Configuration**

| Attribute | Bits | Description |
|---|---|---|
| Disp_blend_cfg | [97:88] | Blending configuration<br>[95:88]: Blending (alpha) value<br>[97:96]: Blending type<br>0 = No Blending<br>1 = Region global blending<br>2 = Color (palette) blending<br>3 = Pixel (embedded alpha) blending |
| disp_trans_cfg | [127:101] | Transparency (color keying enable) configuration bits<br>[103:102]: Transparency LSB bit masking<br>00 = No masking<br>01 = Mask[0] during pixel data comparison to TRANS_COLOR<br>10 = Mask[1:0]<br>11 = Mask[2:0]<br>Example: If TR_LSB_MASK is set to 2'b11, top 5 bits of R/G/B component data are compared to the corresponding 5 bits of TRANS_COLOR R/G/B color.<br>[101]: Transparency enable<br>0 = Disable<br>1 = Enable transparency for video layer<br>[127:104]: Transparency color (RGB24) |

When no-blending is selected, the GRPX forces each alpha to FFh to make the region total opaque. In region global-blending, a programmed alpha value in bits [95:88] (also called as “global blend” value) is assigned to each pixel similar to the no-blending case.

For color or pixel blending, the VPDMA sets the alpha value with CLUT mapping or with embedded Alpha value (argb format source) respectively. In these two modes, the GRPX simply passes the alpha value from the VPDMA to COMP module. Pixel level blending is performed in the COMP module for both color and pixel blending modes.

When transparency is enabled, each pixel color is compared against the transparency color to determine whether the pixel is to be transparent or not. If the colors match, the alpha value is forced to 00h. Otherwise, the alpha value remains as programmed.

The blending level supported is from 0 (transparent) to 255 (totally opaque).

Because the alpha value is also scaled in the same way the color components (RGB) are scaled, all enabled blending/transparency related tasks are performed in the following order:

1. Application of No-blending (force alpha = FFh) or Global-blending (force alpha=programmed global blend level)
2. Transparency check to force alpha = 0 for transparent pixels
3. Stenciling application to force masked pixels (stencil data = 1) with alpha = 0
4. Scaling of Alpha (if scaling enabled)
5. Box Blending (force alpha = programmed box_blend_level)


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.5.2.4 Region Scaler Configuration Attributes

**Table 12-21. Region Scaler Configuration Attributes 1**

| Attribute | Bits | Description |
|---|---|---|
| coefh0_p0 | [9:0] | Coefficient for Horizontal TAP 0 Phase 0 |
| Reserved | [15:10] | reserved |
| coefh0_p1 | [25:16] | Coefficient for Horizontal TAP 0 Phase 1 |
| Reserved | [31:26] | reserved |
| coefh0_p2 | [41:32] | Coefficient for Horizontal TAP 0 Phase 2 |
| Reserved | [47:42] | reserved |
| coefh0_p3 | [57:48] | Coefficient for Horizontal TAP 0 Phase 3 |
| Reserved | [63:58] | reserved |
| coefh0_p4 | [73:64] | Coefficient for Horizontal TAP 0 Phase 4 |
| Reserved | [79:74] | reserved |
| coefh0_p5 | [89:80] | Coefficient for Horizontal TAP 0 Phase 5 |
| Reserved | [95:90] | reserved |
| coefh0_p6 | [105:96] | Coefficient for Horizontal TAP 0 Phase 6 |
| Reserved | [111:106] | reserved |
| coefh0_p7 | [121:112] | Coefficient for Horizontal TAP 0 Phase 7 |
| Reserved | [127:122] | reserved |

**Table 12-22. Region Scaler Attributes 2**

| Attribute | Bits | Description |
|---|---|---|
| coefh1_p0 | [9:0] | Coefficient for Horizontal TAP 1 Phase 0 |
| Reserved | [15:10] | reserved |
| coefh1_p1 | [25:16] | Coefficient for Horizontal TAP 1 Phase 1 |
| Reserved | [31:26] | reserved |
| coefh1_p2 | [41:32] | Coefficient for Horizontal TAP 1 Phase 2 |
| Reserved | [47:42] | reserved |
| coefh1_p3 | [57:48] | Coefficient for Horizontal TAP 1 Phase 3 |
| Reserved | [63:58] | reserved |
| coefh1_p4 | [73:64] | Coefficient for Horizontal TAP 1 Phase 4 |
| Reserved | [79:74] | reserved |
| coefh1_p5 | [89:80] | Coefficient for Horizontal TAP 1 Phase 5 |
| Reserved | [95:90] | reserved |
| coefh1_p6 | [105:96] | Coefficient for Horizontal TAP 1 Phase 6 |
| Reserved | [111:106] | reserved |
| coefh1_p7 | [121:112] | Coefficient for Horizontal TAP 1 Phase 7 |
| Reserved | [127:122] | reserved |

**Table 12-23. Region Scaler Attributes 3**

| Attribute | Bits | Description |
|---|---|---|
| coefh2_p0 | [9:0] | Coefficient for Horizontal TAP 2 Phase 0 |
| Reserved | [15:10] | reserved |
| coefh2_p1 | [25:16] | Coefficient for Horizontal TAP 2 Phase 1 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-23. Region Scaler Attributes 3 (continued)**

| Attribute | Bits | Description |
|---|---|---|
| Reserved | [31:26] | reserved |
| coefh2_p2 | [41:32] | Coefficient for Horizontal TAP 2 Phase 2 |
| Reserved | [47:42] | reserved |
| coefh2_p3 | [57:48] | Coefficient for Horizontal TAP 2 Phase 3 |
| Reserved | [63:58] | reserved |
| coefh2_p4 | [73:64] | Coefficient for Horizontal TAP 2 Phase 4 |
| Reserved | [79:74] | reserved |
| coefh2_p5 | [89:80] | Coefficient for Horizontal TAP 2 Phase 5 |
| Reserved | [95:90] | reserved |
| coefh2_p6 | [105:96] | Coefficient for Horizontal TAP 2 Phase 6 |
| Reserved | [111:106] | reserved |
| coefh2_p7 | [121:112] | Coefficient for Horizontal TAP 2 Phase 7 |
| Reserved | [127:122] | reserved |

**Table 12-24. Region Scaler Attributes 4**

| Attribute | Bits | Description |
|---|---|---|
| coefh3_p0 | [9:0] | Coefficient for Horizontal TAP 3 Phase 0 |
| Reserved | [15:10] | reserved |
| coefh3_p1 | [25:16] | Coefficient for Horizontal TAP 3 Phase 1 |
| Reserved | [31:26] | reserved |
| coefh3_p2 | [41:32] | Coefficient for Horizontal TAP 3 Phase 2 |
| Reserved | [47:42] | reserved |
| coefh3_p3 | [57:48] | Coefficient for Horizontal TAP 3 Phase 3 |
| Reserved | [63:58] | reserved |
| coefh3_p4 | [73:64] | Coefficient for Horizontal TAP 3 Phase 4 |
| Reserved | [79:74] | reserved |
| coefh3_p5 | [89:80] | Coefficient for Horizontal TAP 3 Phase 5 |
| Reserved | [95:90] | reserved |
| coefh3_p6 | [105:96] | Coefficient for Horizontal TAP 3 Phase 6 |
| Reserved | [111:106] | reserved |
| coefh3_p7 | [121:112] | Coefficient for Horizontal TAP 3 Phase 7 |
| Reserved | [127:122] | reserved |

**Table 12-25. Region Scaler Attributes 5**

| Attribute | Bits | Description |
|---|---|---|
| coefh4_p0 | [9:0] | Coefficient for Horizontal TAP 4 Phase 0 |
| Reserved | [15:10] | Reserved |
| coefh4_p1 | [25:16] | Coefficient for Horizontal TAP 4 Phase 1 |
| Reserved | [31:26] | Reserved |
| coefh4_p2 | [41:32] | Coefficient for Horizontal TAP 4 Phase 2 |
| Reserved | [47:42] | Reserved |
| coefh4_p3 | [57:48] | Coefficient for Horizontal TAP 4 Phase 3 |
| Reserved | [63:58] | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-25. Region Scaler Attributes 5 (continued)**

| Attribute | Bits | Description |
|---|---|---|
| coefh4_p4 | [73:64] | Coefficient for Horizontal TAP 4 Phase 4 |
| Reserved | [79:74] | Reserved |
| coefh4_p5 | [89:80] | Coefficient for Horizontal TAP 4 Phase 5 |
| Reserved | [95:90] | Reserved |
| coefh4_p6 | [105:96] | Coefficient for Horizontal TAP 4 Phase 6 |
| Reserved | [111:106] | Reserved |
| coefh4_p7 | [121:112] | Coefficient for Horizontal TAP 4 Phase 7 |
| Reserved | [127:122] | reserved |

**Table 12-26. Region Scaler Attributes 6**

| Attribute | Bits | Description |
|---|---|---|
| countvalhorz | [14:0] | Numerator value for fractional counter countval / 2048<br>coefv0_p0 |
| Reserved | [15] | Reserved |
| fineoffsethorz | [30:16] | Count offset for start of line |
| Reserved | [127:31] | Reserved |

**Table 12-27. Region Scaler Attributes 7**

| Attribute | Bits | Description |
|---|---|---|
| coefv0_p0 | [9:0] | Coefficient for Vertical TAP 0 Phase 0 |
| Reserved | [15:10] | reserved |
| coefv0_p1 | [25:16] | Coefficient for Vertical TAP 0 Phase 1 |
| Reserved | [31:26] | reserved |
| coefv0_p2 | [41:32] | Coefficient for Vertical TAP 0 Phase 2 |
| Reserved | [47:42] | reserved |
| coefv0_p3 | [57:48] | Coefficient for Vertical TAP 0 Phase 3 |
| Reserved | [63:58] | reserved |
| coefv0_p4 | [73:64] | Coefficient for Vertical TAP 0 Phase 4 |
| Reserved | [79:74] | reserved |
| coefv0_p5 | [89:80] | Coefficient for Vertical TAP 0 Phase 5 |
| Reserved | [95:90] | reserved |
| coefv0_p6 | [105:96] | Coefficient for Vertical TAP 0 Phase 6 |
| Reserved | [111:106] | reserved |
| coefv0_p7 | [121:112] | Coefficient for Vertical TAP 0 Phase 7 |
| Reserved | [127:122] | reserved |

**Table 12-28. Region Scaler Attributes 8**

| Attribute | Bits | Description |
|---|---|---|
| coefv1_p0 | [9:0] | Coefficient for Vertical TAP 1 Phase 0 |
| Reserved | [15:10] | reserved |
| coefv1_p1 | [25:16] | Coefficient for Vertical TAP 1 Phase 1 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-28. Region Scaler Attributes 8 (continued)**

| Attribute | Bits | Description |
|---|---|---|
| Reserved | [31:26] | reserved |
| coefv1_p2 | [41:32] | Coefficient for Vertical TAP 1 Phase 2 |
| Reserved | [47:42] | reserved |
| coefv1_p3 | [57:48] | Coefficient for Vertical TAP 1 Phase 3 |
| Reserved | [63:58] | reserved |
| coefv1_p4 | [73:64] | Coefficient for Vertical TAP 1 Phase 4 |
| Reserved | [79:74] | reserved |
| coefv1_p5 | [89:80] | Coefficient for Vertical TAP 1 Phase 5 |
| Reserved | [95:90] | reserved |
| coefv1_p6 | [105:96] | Coefficient for Vertical TAP 1 Phase 6 |
| Reserved | [111:106] | reserved |
| coefv1_p7 | [121:112] | Coefficient for Vertical TAP 1 Phase 7 |
| Reserved | [127:122] | reserved |

**Table 12-29. Region Scaler Attributes 9**

| Attribute | Bits | Description |
|---|---|---|
| coefv2_p0 | [9:0] | Coefficient for Vertical TAP 2 Phase 0 |
| Reserved | [15:10] | reserved |
| coefv2_p1 | [25:16] | Coefficient for Vertical TAP 2 Phase 1 |
| Reserved | [31:26] | reserved |
| coefv2_p2 | [41:32] | Coefficient for Vertical TAP 2 Phase 2 |
| Reserved | [47:42] | reserved |
| coefv2_p3 | [57:48] | Coefficient for Vertical TAP 2 Phase 3 |
| Reserved | [63:58] | reserved |
| coefv2_p4 | [73:64] | Coefficient for Vertical TAP 2 Phase 4 |
| Reserved | [79:74] | reserved |
| coefv2_p5 | [89:80] | Coefficient for Vertical TAP 2 Phase 5 |
| Reserved | [95:90] | reserved |
| coefv2_p6 | [105:96] | Coefficient for Vertical TAP 2 Phase 6 |
| Reserved | [111:106] | reserved |
| coefv2_p7 | [121:112] | Coefficient for Vertical TAP 2 Phase 7 |
| Reserved | [127:122] | reserved |

**Table 12-30. Region Scaler Attributes 10**

| Attribute | Bits | Description |
|---|---|---|
| coefv3_p0 | [9:0] | Coefficient for Vertical TAP 3 Phase 0 |
| Reserved | [15:10] | reserved |
| coefv3_p1 | [25:16] | Coefficient for Vertical TAP 3 Phase 1 |
| Reserved | [31:26] | reserved |
| coefv3_p2 | [41:32] | Coefficient for Vertical TAP 3 Phase 2 |
| Reserved | [47:42] | reserved |
| coefv3_p3 | [57:48] | Coefficient for Vertical TAP 3 Phase 3 |
| Reserved | [63:58] | reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-30. Region Scaler Attributes 10 (continued)**

| Attribute | Bits | Description |
|---|---|---|
| coefv3_p4 | [73:64] | Coefficient for Vertical TAP 3 Phase 4 |
| Reserved | [79:74] | reserved |
| coefv3_p5 | [89:80] | Coefficient for Vertical TAP 3 Phase 5 |
| Reserved | [95:90] | reserved |
| coefv3_p6 | [105:96] | Coefficient for Vertical TAP 3 Phase 6 |
| Reserved | [111:106] | reserved |
| coefv3_p7 | [121:112] | Coefficient for Vertical TAP 3 Phase 7 |
| Reserved | [127:122] | reserved |

**Table 12-31. Region Scaler Attributes 11**

| Attribute | Bits | Description |
|---|---|---|
| countvalvert | [14:0] | Numerator value for fractional counter<br>countval / 248<br>coefv0_p0 |
| Reserved | [15] | Reserved |
| fineoffsetvert | [30:16] | Count offset for start of line |
| Reserved | [127:31] | Reserved |

#### *12.2.5.2.5 VPDMA Configuration*

VPDMA needs to be setup to configure the GRPX module and input data transfer. The initialization sequence for each frame (assuming frame/region attributes change for every frame) should be as follows:

- Frame Attribute Configuration Descriptor
- Region Scalar Co-efficient/Attribute Configuration Descriptor (if required)
- Region Look-Up table Configuration Descriptor (if required)
- Region Stencil Data Descriptor (if required)
- Region Data Descriptor (different from regular data descriptor and it configures region attributes also)

Some of the previous configuration descriptors can be combined into single configuration descriptor (using class 0 type).

If the configuration in any of the previous configuration descriptors is not changed from region to region or frame to frame, it is not required to set up those configuration descriptors each time in the VPDMA LIST. The configuration of descriptors is explained and illustrated in the *VPDMA* chapter.


Copyright © 2011–2016, Texas Instruments Incorporated

## 12.2.6 Standard-Definition Video Encoder (SD_VENC)

### 12.2.6.1 Overview

The VENC (video encoder) converts digital component YCbCr/RGB video signals to conform to the various TV standard analog video. CVBS and S-Video outputs are supported.

#### 12.2.6.1.1 Standards

- SMPTE 170M
  - Composite Analog Video Signal – NTSC for Studio Application
- ITU-R BT.470
  - Conventional television systems
- ITU-R BT.601
  - Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios
- ITU-R BT.1119
  - Wide-screen signaling for broadcasting (Signaling for wide-screen and other enhanced television parameters)
- EIAJ CPR-1204
  - Transfer Method of Video ID information using Vertical Blanking Interval (525 line system)
- ETSI EN 300 294
  - Television systems; 625-line television Wide Screen Signaling (WSS)
- EIA/CEA-608-E
  - Line 21 Data Services

#### 12.2.6.1.2 Features

- SDTV Feature
  - Master Clock Input - 27 MHz
  - Support Following TV standard
    - NTSC-M
    - NTSC-J
    - NTSC-4.43
    - PAL-B/D/G/H/K/I
    - PAL-M
    - PAL-N
    - PAL-Nc
    - PAL-60
    - SECAM-B/D/G/H/K/L
  - Composite (CVBS)
  - S-Video (Y/C)
  - Macrovision (Rev7.1) Anticopy Protection
  - Non-Interlace
  - CGMS/WSS
  - Line 21 Closed Caption Data Encoding
  - Programmable Sync Amplitude
  - Programmable Color-burst Amplitude
  - Programmable Color Space Converter
  - Programmable Sub-carrier Frequency and SCH phase


Copyright © 2011–2016, Texas Instruments Incorporated

  - Programmable Luma Delay
- 24-bit Input Interface (either RGB 4:4:4 or YCbCr 4:4:4)
- Two channel 12-bit 2x Over-Sampled DAC Output
- Programmable Sync Output
- Master/Slave Operation
- Internal Color Bar Generation
- Support VBI I/F for the generic VBI data insertion

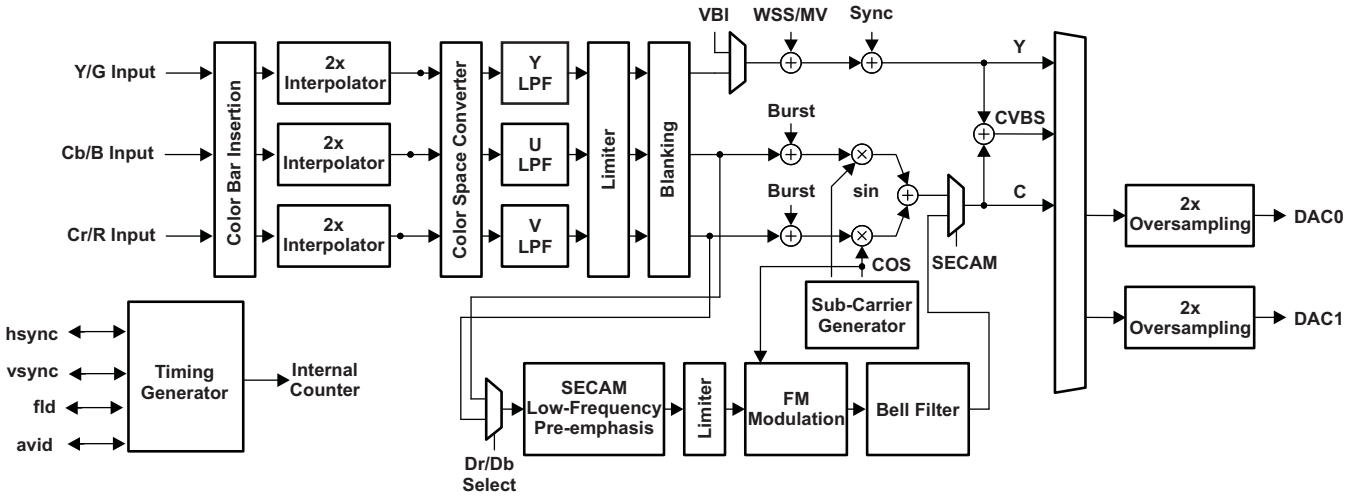
#### 12.2.6.1.3 Features Not Supported

- Square pixel sampling
- Input with embedded sync conforming to ITU-R BT.656
- Analog component video output

### 12.2.6.2 Functional Description

#### 12.2.6.2.1 Block Diagram

**Figure 12-40. VENC Block Diagram**




#### 12.2.6.2.2 Operating Modes

##### 12.2.6.2.2.1 Synchronous Mode

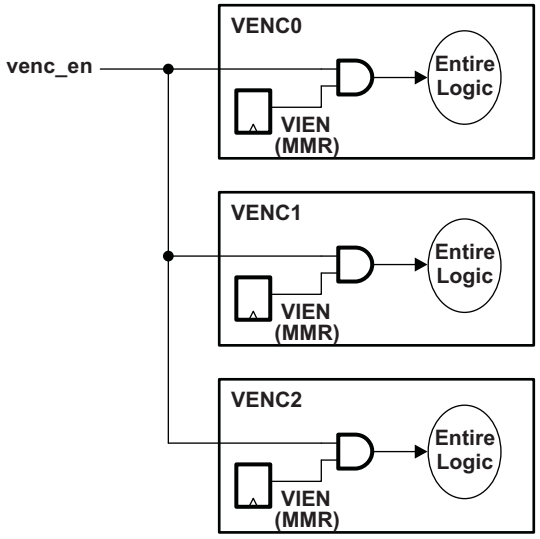
The VENC supports two synchronous modes; master and slave. Each mode can be used in above two video modes.

- Master mode: This mode operates in synchronization with horizontal / vertical sync signals generated in built-in sync signal generator..
- Slave mode: This mode operates in synchronization with sync signals input from outside. Setting the SLV register to 1 enables the slave mode. The external sync inputs are active high by default and their polarity can be independently inverted by registers.

The VENC starts the operation when the external input venc_en is HIGH and the VIEN register is set to 1. The venc_en signal is used for synchronizing multiple video encoders for the multichannel video systems. An external timing manager should take care of the generation of the venc_en signal. When only a single video encoder is used or no synchronization is required among the multiple video encoders, the venc_en can be tied to HIGH. Figure 12-41 provides an example of this synchronization operation.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-41. Example of Multiple VENC Synchronization**

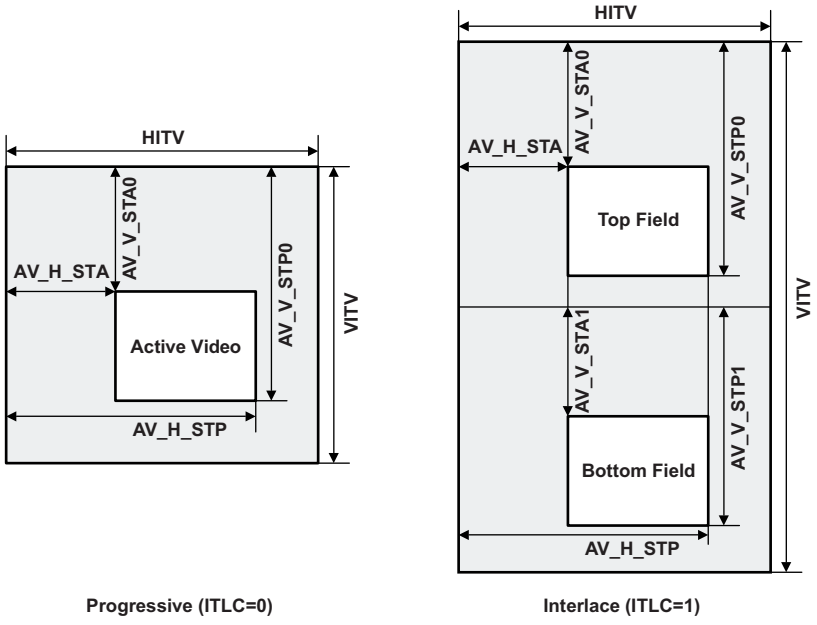



#### 12.2.6.2.3 Timing

##### 12.2.6.2.3.1 Video Timing

The video timing is programmable as shown in Figure 12-42. The HITV and VITV registers set the horizontal and vertical frame sizes. When the target video is interlaced, set ITLC=1. Line per field is VITV/2. Unequal line per field is optionally available. When UEL=1, line per field alternately toggles between (VITV-1)/2 and (VITV+1)/2. The active video position is set by the AV registers.

**Figure 12-42. Video Timing**





Copyright © 2011–2016, Texas Instruments Incorporated

For the analog video output, one of the following TV formats in Table 12-32 also needs to be specified by the FMT register accordingly. The FMT setting specifies the format-dependent timing parameters such as sync rise/fall build-up time, active video rise/fall time, vertical sync and equalizing pulse position, WSS/CC waveform position and so on.

**Table 12-32. TV Formats**

| FMT | Video Format | Type |
|---|---|---|
| 0 | 525i | SDTV |
| 1 | 625i | |

For SDTV (525i/625i), composite video (CVBS) and S-video are available. The output combination from two DACs is described in Section 12.2.6.2.14.2.

#### *12.2.6.2.3.2 Input I/F Timing*

The input I/F consist of venc_dtv_hs (horizontal sync), venc_dtv_vs (vertical sync), venc_dtv_fid (field ID) and venc_dtv_avid (active video). The timing of each signal is programmable. The timing reference is the internal base counter. The internal base counter starts when VIEN register is set to 1 and the venc_en signal is HIGH. The phase between clk1x and the internal base counter is maintained so that the horizontal counter begins increment at phase1x = 1 as shown in Figure 12-43.

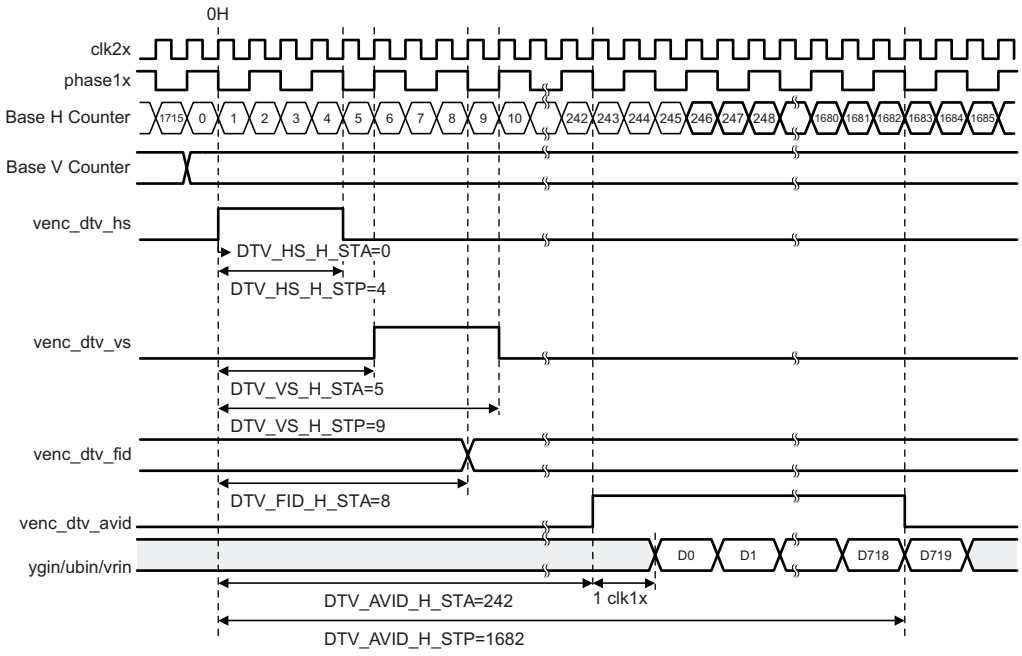
The horizontal timing of each signal is configured in unit of clk2x. The AVID timing should be configured considering the input latency of one cycle of clk1x.

Vertical timing resolution is 1/2H. Independent AVID timings are available for each field for an interlaced format. VS and FID have a unique configuration for both fields

For progressive or non-interlaced formats, the internal base FID is fixed to low. However, it is possible to force toggling venc_dtv_fid at the specified line by setting DTV_FID_F_STA0 register to 1.

Figure 12-43 and Figure 12-44 show the input I/F timing chart with example settings.

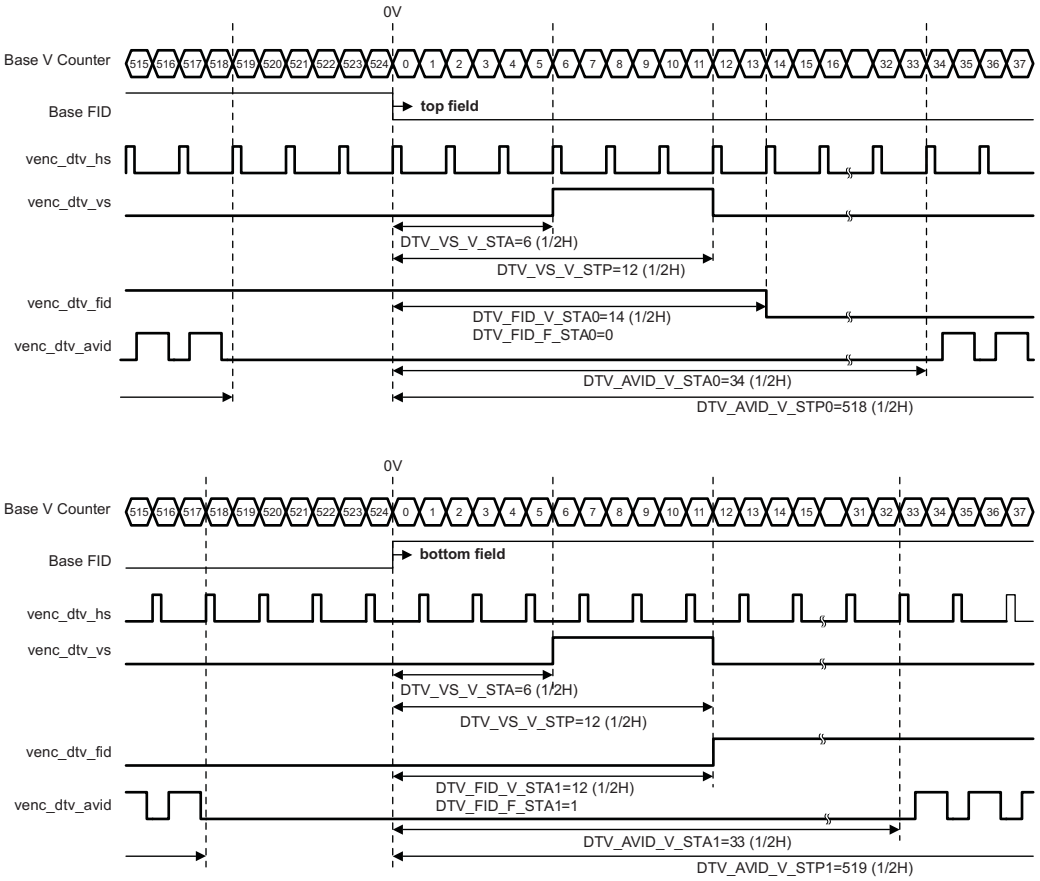
**Figure 12-43. Input I/F Horizontal Timing**






Copyright © 2011–2016, Texas Instruments Incorporated

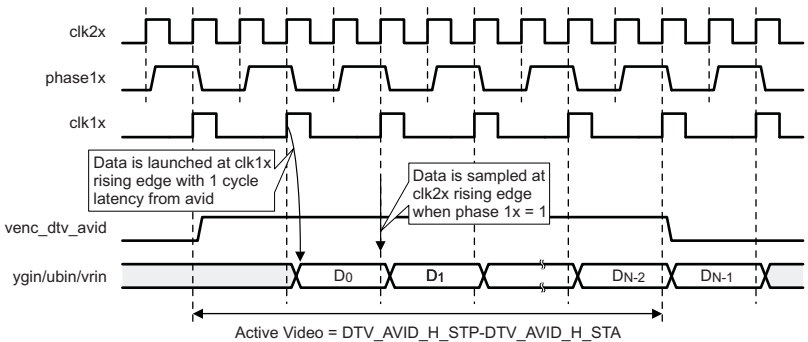
**Figure 12-44. Input I/F Vertical Timing**




The VENC generates a venc_dtv_avid signal for data request purpose for the upstream module. Input data is expected to arrive at the VENC one clk1x cycle (two clk2x cycles) after an avid assertion. The supported input data format is RGB 4:4:4 or YCbCr 4:4:4. Input data is sampled at clk2x rising edge at phase1x=1. Figure 12-45 shows the timing chart of the input interface.

It is possible to sample the input data with an inversion. When the DIIV register is 1, the input data is inverted before sampling.

**Figure 12-45. Input Data Timing**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.3.3 Interrupt Timing

An interrupt pulse is output from the venc_irq port with a single clk2x cycle pulse width every field. Its horizontal and vertical assert position is configured by IRQ_H_STA and IRQ_V_STA registers, respectively.

**Figure 12-46. Interrupt Horizontal Timing**

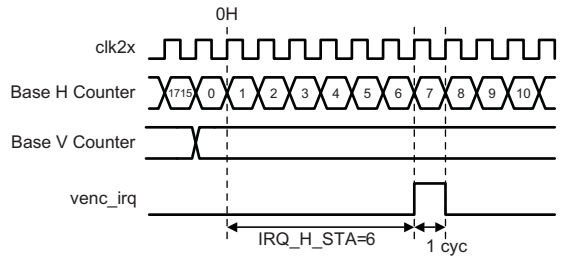
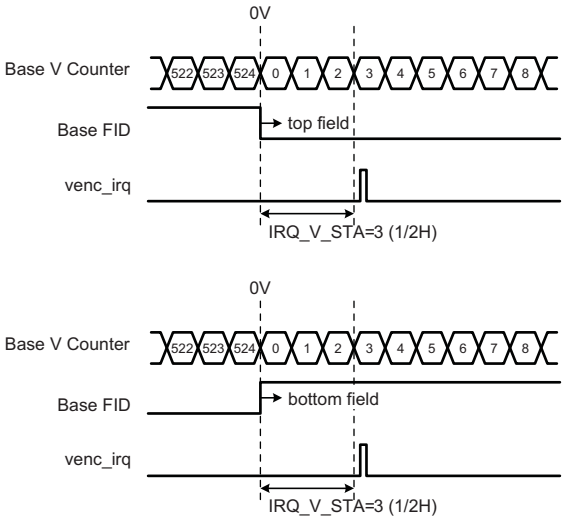



**Figure 12-47. Interrupt Vertical Timing**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.3.4 TV Detection Pulse Timing

The VENC supports a TVDET gate pulse generation to connect with the DAC which has the TV detect feature. Its horizontal timing (see Figure 12-48) is configured by TVDETGP_H_STAand TVDETGP_H_STP registers, and vertical timing (see Figure 12-49) is configured by TVDETGP_V_STA and TVDETGP_V_STP registers.

**Figure 12-48. TVDETGP Horizontal Timing**

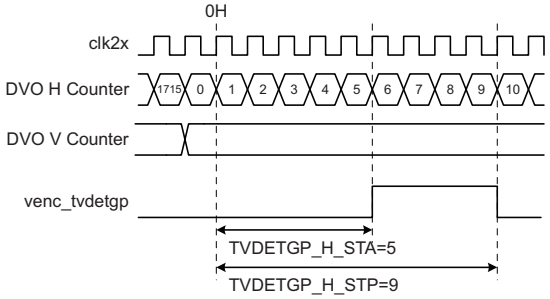
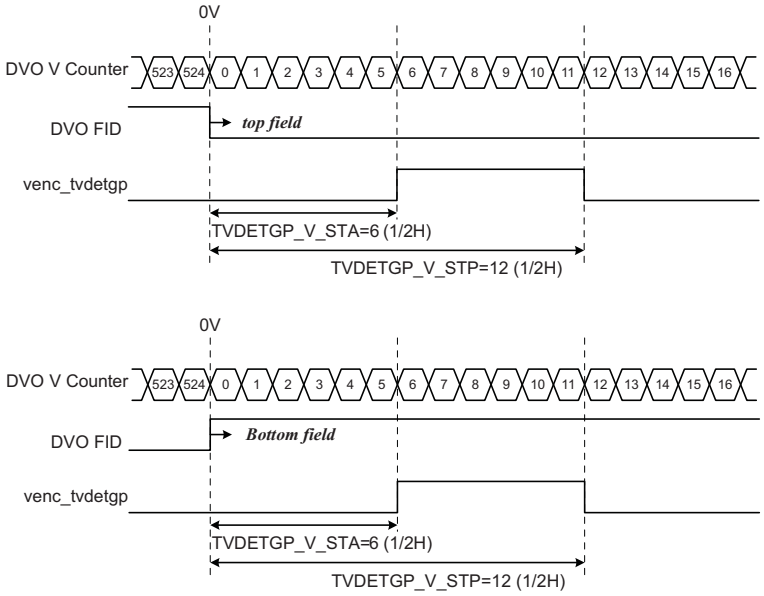



**Figure 12-49. TVDETGP Vertical Timing**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.3.5 DAC Video Output Timing

##### 12.2.6.2.3.5.1 SDTV Timing

Figure 12-50 shows horizontal timing characteristics of the SDTV DAC video output. The number of pixels per line is determined by the TV scan format register FMT. The horizontal sync pulse width is also fixed depending on the FMT setting. The color burst start/end positions are specified by BST_H_STA and BST_H_STP registers, and the active video start/end positions are specified by AV_H_STA and AV_H_STP registers. The timing reference of these registers is the internal TV counter.

The AV_H registers specify the window position for the active video. There are similar registers to specify the AVID timing which controls the input video timing. They are independent.

Serration and equalization pulses are inserted on the appropriate lines defined in the standard. Color burst is automatically inserted on the appropriate lines defined in the standard. Figure 12-51 and Figure 12-52 show the vertical timing characteristics of NTSC and PAL. The vertical sync duration is selectable from either 3H or 2.5H by the SVSW register. Typically, 3H VSYNC is used for 525i and 2.5H VSYNC is used for 625i, but this is not always the case. For example, PAL-N has 3H VSYNC even though it is a 625i system.

The color burst is not indicated in these figures but its sequence is managed by hardware. The PAL meander sequence is supported for PAL mode.

**Figure 12-50. SDTV Horizontal Timing**

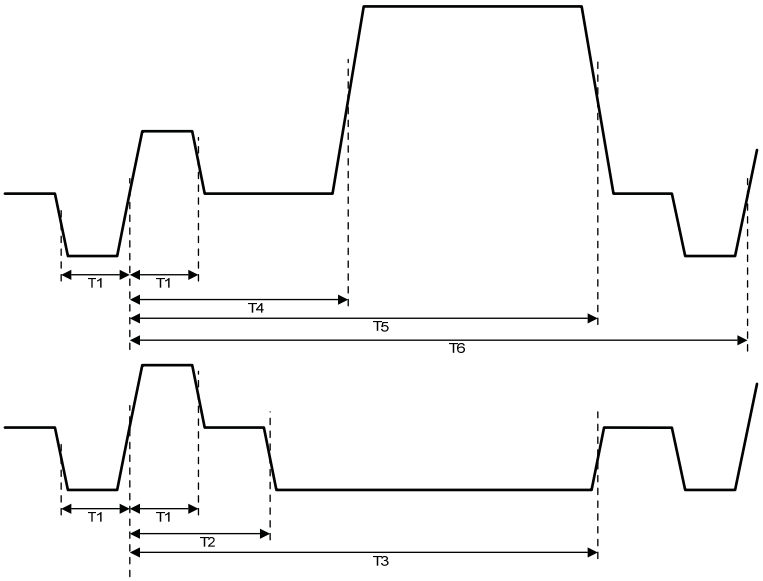



**Table 12-33. SDTV Horizontal Timing Parameter**

| Parameter | Item | 525I (FMT = 0) | 625I (FMT = 1) | Unit |
|---|---|---|---|---|
| T1 | Horizontal Sync Pulse Width | 127 | | clk2x |
| T2 | H Ref to Burst Start | BST_H_STA | | |
| T3 | H Ref to Burst End | BST_H_STP | | |
| T4 | H ref to H Blanking End | AV_H_STA | | |
| T5 | H ref to H Blanking Start | AV_H_STP | | |
| T6 | 1H | HITV | | |
| T7 | Equalizing Pulse Width | 62 | 63 | |
| T8 | Serration Pulse Width | 127 | | |
| T9 | 1/2H | HITV/2 | | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-51. 525i Vertical Timing**

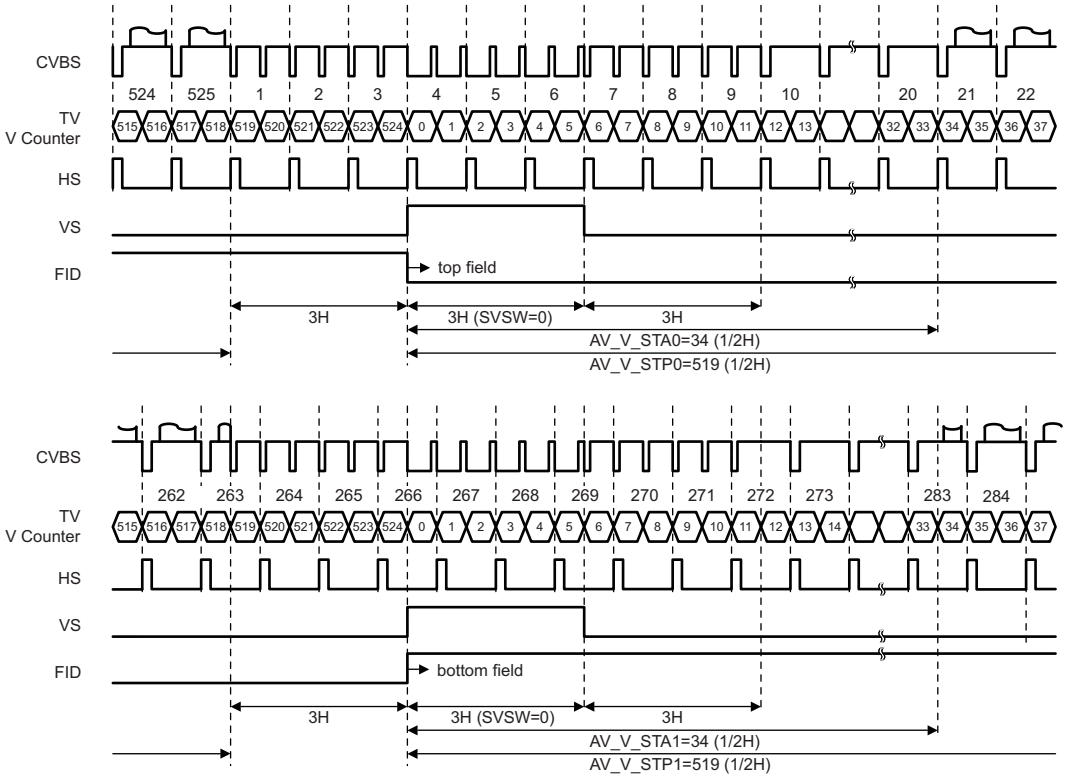
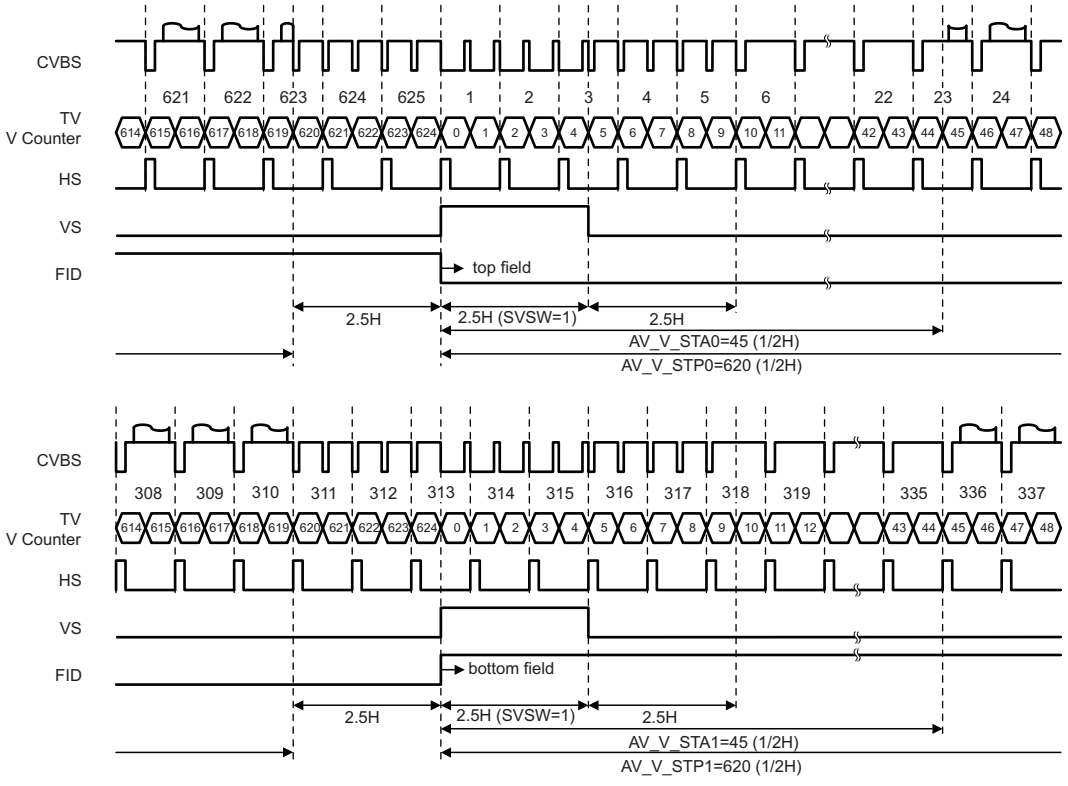



**Figure 12-52. 625i Vertical Timing**





Copyright © 2011–2016, Texas Instruments Incorporated

Figure 12-53 and Figure 12-54 show the DAC video output pipeline delay against the internal base counter. The pipeline delay latency is determined by the input pixel rate. The input pixel rate is specified by the PXLR register. The SDTV input pixel rate is typically 13.5 MHz, which is half of clk2x (PXLR = 0). However, some unusual systems may apply the 27 MHz input pixel rate. It is possible to set PXLR = 1.

**Figure 12-53. SDTV DAC Video Output Pipeline Delay (PXLR = 0)**

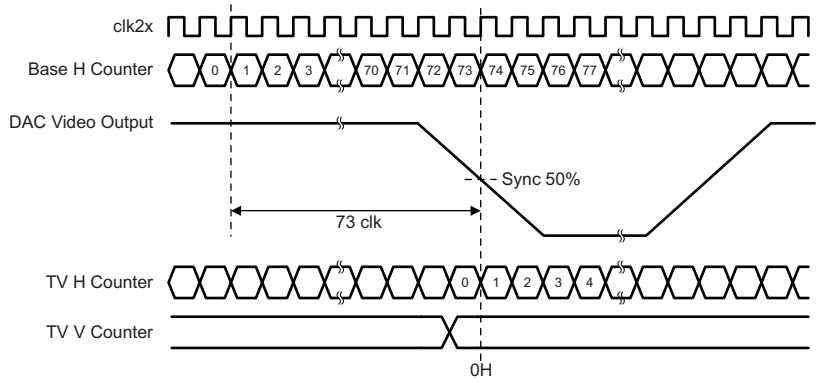
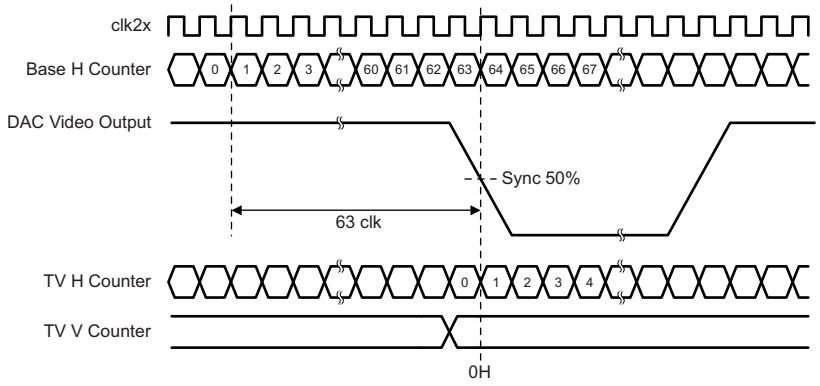



**Figure 12-54. SDTV DAC Video Output Pipeline Delay (PXLR = 1)**





Copyright © 2011–2016, Texas Instruments Incorporated

Figure 12-55 and Figure 12-56 show the non-interlaced operation for 262p and 312p. Table 12-34 shows the register setting for the non-interlaced format. Although it is progressive scan, the ITLC register has to be 1 because the internal hardware still relies on the interlaced timing control. Note that the FID is forced to low for non-interlaced mode.

**Figure 12-55. 262p (Non-Interlaced NTSC) Vertical Timing**

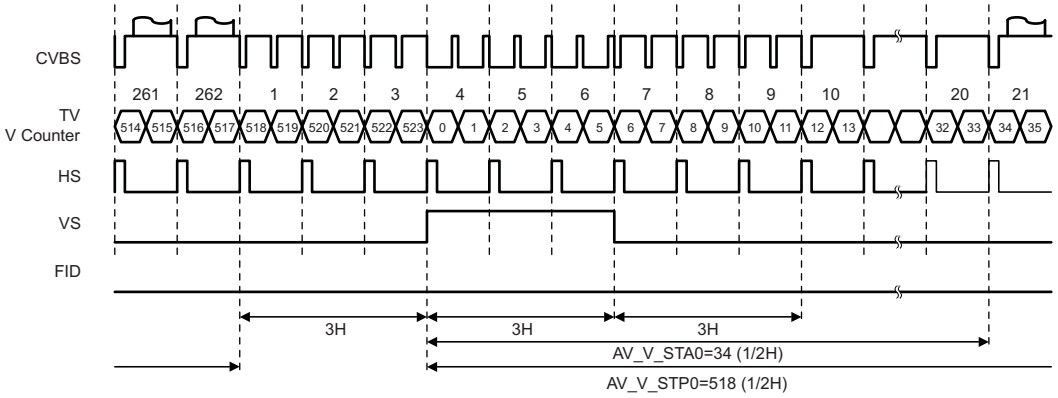



**Figure 12-56. 312p (Non-Interlaced PAL) Vertical Timing**

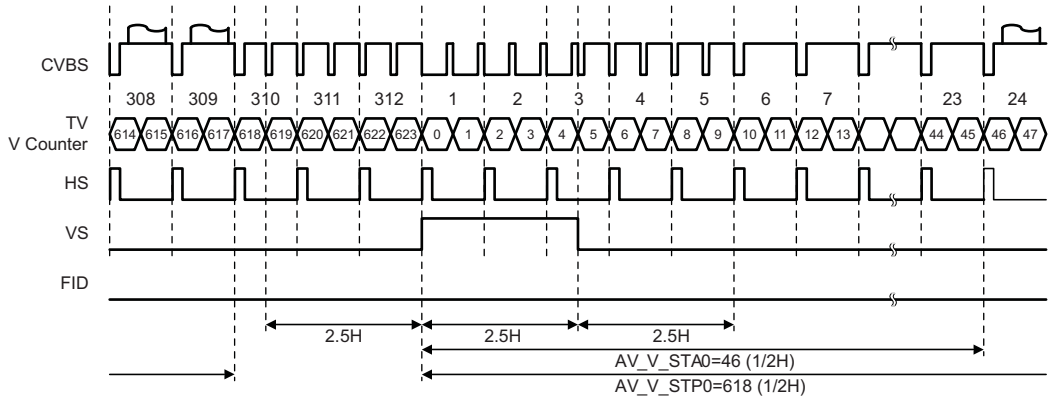



**Table 12-34. Non-Interlaced Format**

| FMT | HITV | VITV | ITLC | UEL | Derived Format |
|---|---|---|---|---|---|
| 0 | 1716 | 525 | 1 | 0 | 525i |
| | | 524 | | | 262H Non-Interlace |
| | | 526 | | | 263H Non-Interlace |
| | | 525 | | 1 | 262H, 263H Non-Interlace |
| 1 | 1728 | 625 | 1 | 0 | 625i |
| | | 624 | | | 312H Non-Interlace |
| | | 626 | | | 313H Non-Interlace |
| | | 625 | | 1 | 312H, 313H Non-Interlace |

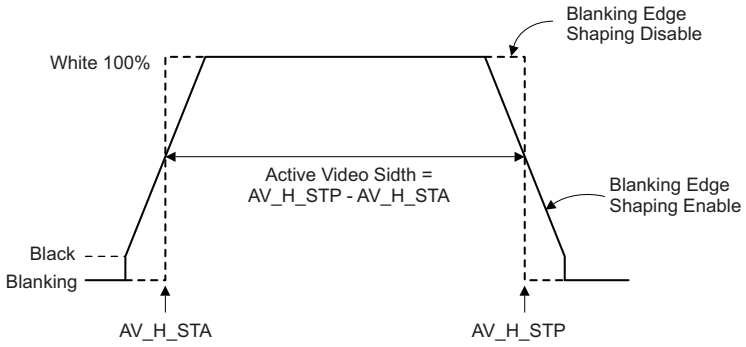

Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.3.6 Horizontal Blanking Edge Shaping

Horizontal video blanking edge is shaped so that the output video has the proper blanking transition.

This feature is enabled by default but can be disabled by setting 1 to the BLS register. Figure 12-57 shows the waveforms when blanking edge shaping is enabled and disabled.

**Figure 12-57. Horizontal Blanking Shaping**

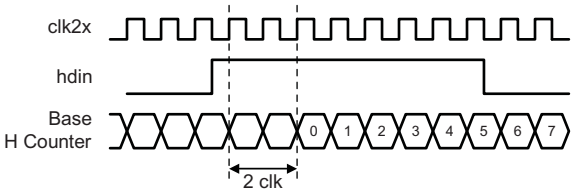



#### 12.2.6.2.3.7 Slave Mode Timings

##### 12.2.6.2.3.7.1 Slave Mode Horizontal Timing

Figure 12-58 shows slave mode horizontal timing. The hdin acts as a horizontal sync input. The internal base horizontal counter is reset after two clk2x cycles upon detecting a rising edge of hdin. The polarity of hdin can be inverted by the HIP register.

**Figure 12-58. Slave Mode Horizontal Timing**




##### 12.2.6.2.3.7.2 Slave Mode Vertical Timings

Figure 12-59, Figure 12-60, and Figure 12-61 show the slave mode vertical timing. The vdin acts as a vertical sync input. The internal base vertical counter is reset when vdin rise transition is detected at hdin rising edge or 0.5H position for the interlaced mode. If vdin is behind hdin assertion. Vertical reset is suspended until the next hdin rise edge or the next 0.5H for the interlaced mode.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-59. Interlaced Slave Vertical Timing (FMD = 0,1)**

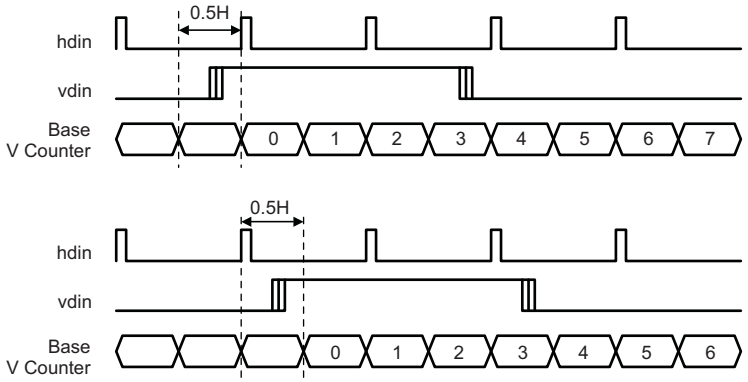
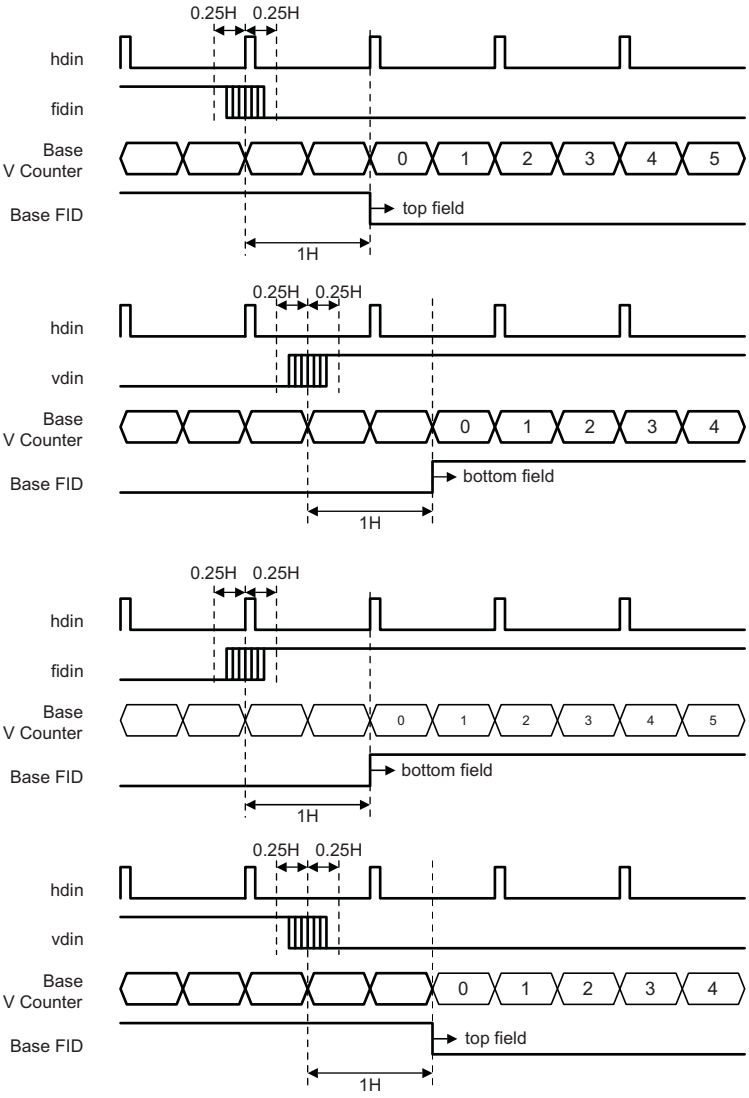


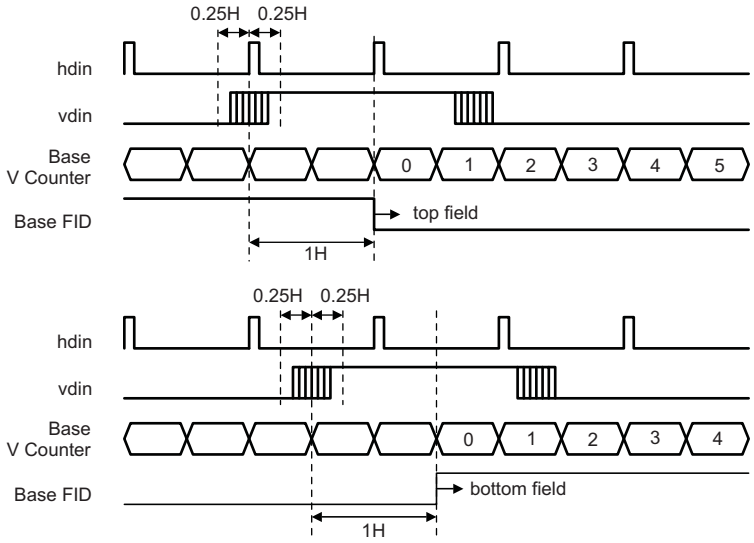

**Figure 12-60. Interlaced Slave Vertical Timing (FMD = 2)**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-61. Interlaced Slave Vertical Timing (FMD = 3)**




#### *12.2.6.2.3.7.3 Slave Mode Field Detection*

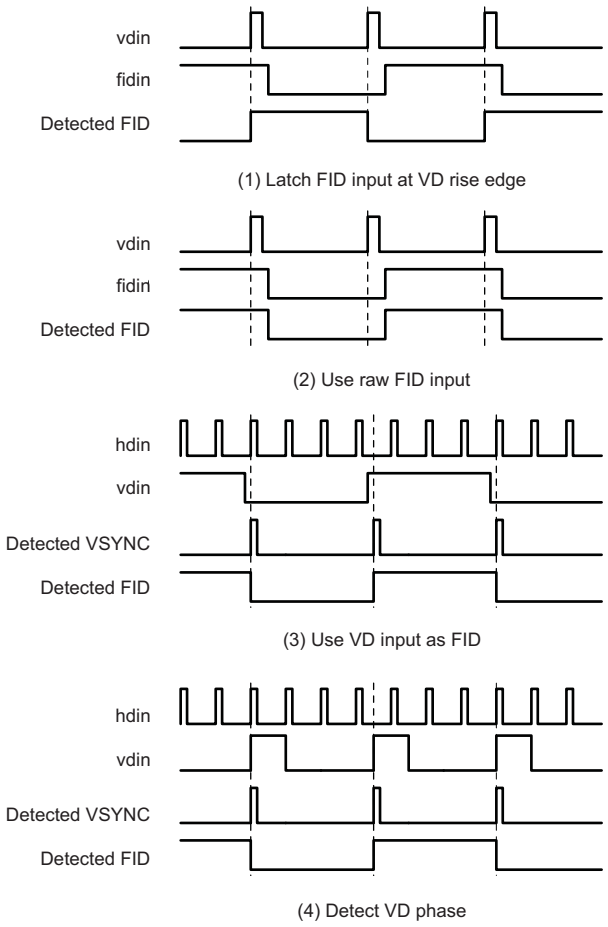
For slave interface mode, the field ID is detected by one of the following methods:

1. Latch FID input at VD rise edge
2. Use raw FID input
3. Use VD input as FID
4. Detect VD phase

These four modes can be selected by the FMD register. Figure 12-62 shows the timing of each mode.

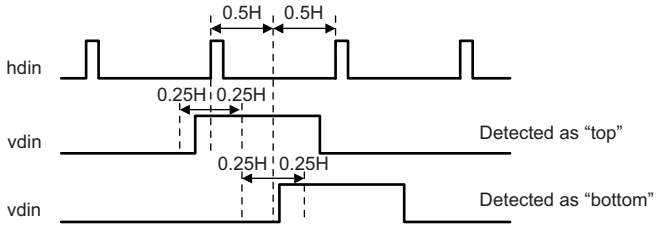

Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-62. Field Detection Mode**




In option4 (Detect VD phase), the timing generator detects VD asserted position in a line. When vdin occurs within 0.25H around hdin, the fid is detected as “top.” Otherwise, the fid is detected as “bottom.” Figure 12-63 shows this detection scheme. When in non-standard mode, field id is always detected as “top” in option4.

**Figure 12-63. Field Detection by VD Phase (FMD=3)**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.4 Color Bar Insertion

When the CBAR register is set to 1, the input pixel data is replaced by the predefined color bar data. The data format of the inserted color bar is 8-bit RGB as shown in Table 12-35. Although the color bar data is generated internally, its timing is based on the AVID signal so it is required to configure the DTV_AVID_STA and DTV_AVID_STP registers appropriately.

When color bar mode is used, the color space conversion matrix should be appropriately programmed so that it can convert the full-range RGB to the desired color format.

**Table 12-35. Color Bar Table**

| Color | G | B | R |
|---|---|---|---|
| White | 255 | 255 | 255 |
| Yellow | 255 | 0 | 255 |
| Cyan | 255 | 255 | 0 |
| Green | 255 | 0 | 0 |
| Magenta | 0 | 255 | 255 |
| Red | 0 | 0 | 255 |
| Blue | 0 | 255 | 0 |
| Black | 0 | 0 | 0 |

#### 12.2.6.2.5 2× Interpolation

Three pixel components, YG, UB and VR, are applied to 2x interpolators. The interpolation is intended to interpolate the SD input pixel which is 1x pixel rate (13.5 MHz) to the 2× clock rate (27 MHz). The interpolator consists of 27 tap symmetric FIR with programmable coefficients. The coefficient register format is s(0.7). The transfer function is expressed as:

$$H(z) = \sum_{n=0}^{26} h(n)\, z^{-n}$$

**Table 12-36. 2× Interpolation Filter Coefficients**

| Coefficients | Value | Default |
|---|---|---|
| h(0), h(26) | UPFC0 | 0 |
| h(1), h(25) | 0 | 0 |
| h(2), h(24) | UPFC1 | 0 |
| h(3), h(23) | 0 | 0 |
| h(4), h(22) | UPFC2 | 0 |
| h(5), h(21) | 0 | 0 |
| h(6), h(20) | UPFC3 | 0 |
| h(7), h(19) | 0 | 0 |
| h(8), h(18) | UPFC4 | 6/128 |
| h(9), h(17) | 0 | 0 |
| h(10), h(16) | UPFC5 | -20/128 |
| h(11), h(15) | 0 | 0 |
| h(12), h(14) | UPFC6 | 78/128 |
| h(13) | 1 | 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

The interpolation is disabled by default and can be enabled by the UPS register.

The interpolator takes 8-bit input (8.0) and generates 12-bit output that has 8-bit integer and 4-bit fractional parts (8.4). After the 2x interpolation, the data processing is done in 12-bit throughout the data pipe. When the interpolation is disabled, the zeros are inserted on 4 LSBs.

#### 12.2.6.2.6 Color Space Converter

The 2x interpolated data then comes into a color space converter (CSC) to be converted into the desired color format. Two independent color space converters are available for CVBS and S-Video paths.

The CSC is implemented based on following equation:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \frac{1}{64} \begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{bmatrix} \begin{bmatrix} X_1 + D_1 \\ X_2 + D_2 \\ X_3 + D_3 \end{bmatrix} + \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix}$$

The input is X (8.4) and the output is Y s(12.0). The 3x3 matrix coefficients A, B and C are 13 bit s(6.6). The input offset D is 13 bit s(8.4) and the output offset E is 13 bit s(12.0). The parameters in the equation, A, B, C, D and E, are programmable.

Since the CSC output directly contributes the output video amplitude, the coefficients should be determined considering the final video amplitude. In other words, the CSC can also perform amplification for the video output. The output gain should be included in the matrix coefficients. The input offset D should be applied when input data has an offset like BT601. The output offset E should be used to put pedestal level which is needed for NTSC-M.

Following the above CSC conversion, a limiter is available to constrain the output data to the upper and lower bounds.

The programming example of the CSC will be described later in the sections for each targeted system.

#### 12.2.6.2.7 CVBS Encoding

##### 12.2.6.2.7.1 Color Systems

Targeted color modulation system of SDTV is specified by the CCM register. Table 12-37 shows the supported color systems. Quadrature AM modulation is used for NTSC or PAL systems while FM modulation is used for SECAM. Independent programmability of the timing and color systems enables to generate variant TV formats such as PAL-M, PAL-60 and so on.

**Table 12-37. Color Systems**

| CCM | Color System |
|---|---|
| 0 | NTSC |
| 1 | PAL |
| 2 | SECAM |
| 3 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.7.2 NTSC/PAL Encoding

##### 12.2.6.2.7.2.1 YUV Generation

First, YUV data should be generated by an appropriate color space conversion. The input color format could be YCbCr or RGB. The CSC can convert either format to YUV.

The fundamental YUV formula is defined in the standard as follows:

$$E'_Y = 0.587E'_G + 0.114E'_B + 0.299E'_R$$
$$E'_U = K_b\left(E'_B - E'_Y\right)$$
$$E'_V = K_r\left(E'_R - E'_Y\right)$$

where $K_b$ and $K_r$ are kell factors which have the following values:

$$K_b = \frac{1}{3}\sqrt{\frac{209556997}{96146491}} = 0.492111$$
$$K_r = \sqrt{\frac{221990474}{288439473}} = 0.877283$$

$E_B{}' - E_Y{}'$ and $E_R{}' - E_Y{}'$ are derived using the following equation:

$$\begin{bmatrix} E'_Y \\ E'_B - E'_Y \\ E'_R - E'_Y \end{bmatrix} = \begin{bmatrix} 0.587 & 0.114 & 0.299 \\ -0.587 & 0.886 & -0.299 \\ -0.587 & -0.114 & 0.701 \end{bmatrix} \begin{bmatrix} E'_G \\ E'_B \\ E'_R \end{bmatrix} \equiv \mathbf{M_0} \begin{bmatrix} E'_G \\ E'_B \\ E'_R \end{bmatrix}$$

Using $M_0$, the fundamental YUV formula can be rewritten as follows.

$$\begin{bmatrix} E'_Y \\ E'_U \\ E'_V \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & K_b & 0 \\ 0 & 0 & K_r \end{bmatrix} \bullet \mathbf{M_0} \bullet \begin{bmatrix} E'_G \\ E'_B \\ E'_R \end{bmatrix} \equiv \mathbf{M} \begin{bmatrix} E'_G \\ E'_B \\ E'_R \end{bmatrix}$$

The following are examples of CSC programming for various input and output combinations.


Copyright © 2011–2016, Texas Instruments Incorporated

***Example 12-1. Full-range RGB [0…255] to YUV 10:4 Picture Sync Ratio with 7.5% Setup***

In this example, it is assumed that DAC is configured to have maximum amplitude of 1400 mV.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = A \bullet \mathbf{M} \bullet \frac{1}{255} \begin{bmatrix} G \\ B \\ R \end{bmatrix} + \begin{bmatrix} S_Y \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{1}{64} \begin{bmatrix} 285 & 55 & 145 \\ -140 & 211 & -71 \\ -250 & -48 & 298 \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix} + \begin{bmatrix} 157 \\ 0 \\ 0 \end{bmatrix}$$

$$A = \frac{10}{14} \times 1000(mV) \times 92.5\% \times \frac{4095(LSB)}{1400(mV)} = 1932.6$$

$$S_Y = \frac{10}{14} \times 1000(mV) \times 7.5\% \times \frac{4095(LSB)}{1400(mV)} = 157$$

$$Sync = \frac{4}{14} \times 1000(mV) \times \frac{4095(LSB)}{1400(mV)} = 836$$

This coefficient set is preset by default for the CVBS CSC registers.

***Example 12-2. Full-range RGB to YUV 10:4 Picture Sync Ratio with Zero Setup***

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = A \bullet \mathbf{M} \bullet \frac{1}{255} \begin{bmatrix} G \\ B \\ R \end{bmatrix} + \begin{bmatrix} S_Y \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{1}{64} \begin{bmatrix} 308 & 60 & 157 \\ -152 & 229 & -77 \\ -270 & -52 & 322 \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix}$$

$$A = \frac{10}{14} \times 1000(mV) \times \frac{4095(LSB)}{1400(mV)} = 2089$$

$$S_Y = 0$$


Copyright © 2011–2016, Texas Instruments Incorporated

***Example 12-3. Studio RGB [0…219] to YUV 10:4 Picture Sync Ratio with 7.5% Setup***

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = A \bullet \mathbf{M} \bullet \frac{1}{219}\begin{bmatrix} G \\ B \\ R \end{bmatrix} + \begin{bmatrix} S_Y \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{1}{64}\begin{bmatrix} 332 & 64 & 169 \\ -163 & 246 & -83 \\ -291 & -56 & 347 \end{bmatrix}\begin{bmatrix} G \\ B \\ R \end{bmatrix} + \begin{bmatrix} 157 \\ 0 \\ 0 \end{bmatrix}$$

$$A = \frac{10}{14} \times 1000(mV) \times 92.5\% \times \frac{4095(LSB)}{1400(mV)} = 1932.6$$

$$S_Y = \frac{10}{14} \times 1000(mV) \times 7.5\% \times \frac{4095(LSB)}{1400(mV)} = 157$$

***Example 12-4. Studio RGB [0…219] to YUV 10:4 Picture Sync Ratio with Zero Setup***

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = A \bullet \mathbf{M} \bullet \frac{1}{219}\begin{bmatrix} G \\ B \\ R \end{bmatrix} + \begin{bmatrix} S_Y \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{1}{64}\begin{bmatrix} 358 & 70 & 183 \\ -176 & 266 & -90 \\ -314 & -61 & 375 \end{bmatrix}\begin{bmatrix} G \\ B \\ R \end{bmatrix}$$

$$A = \frac{10}{14} \times 1000(mV) \times \frac{4095(LSB)}{1400(mV)} = 2089$$

$$S_Y = 0$$


Copyright © 2011–2016, Texas Instruments Incorporated

***Example 12-5. Full-range RGB [0…255] to YUV 7:3 Picture Sync Ratio with Zero Setup***

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = A \bullet \mathbf{M} \bullet \frac{1}{255} \begin{bmatrix} G \\ B \\ R \end{bmatrix} + \begin{bmatrix} S_Y \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{1}{64} \begin{bmatrix} 351 & 68 & 179 \\ -173 & 261 & -88 \\ -308 & -60 & 368 \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix}$$

$$A = \frac{7}{10} \times 1000(mV) \times \frac{4095(LSB)}{1400(mV)} = 2047.5$$

$$S_Y = 0$$

$$Sync = \frac{3}{10} \times 1000(mV) \times \frac{4095(LSB)}{1400(mV)} = 878$$

***Example 12-6. ITU-R BT.601 YCbCr to YUV 10:4 Picture Sync Ratio with 7.5% Setup***

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = A \begin{bmatrix} 1 & 0 & 0 \\ 0 & K_b & 0 \\ 0 & 0 & K_r \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.886 & 0 \\ 0 & 0 & 0.701 \end{bmatrix} \begin{bmatrix} 1/219 & 0 & 0 \\ 0 & 1/112 & 0 \\ 0 & 0 & 1/112 \end{bmatrix} \begin{bmatrix} Y - 16 \\ C_b - 128 \\ C_r - 128 \end{bmatrix} + \begin{bmatrix} S_Y \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{1}{64} \begin{bmatrix} 565 & 0 & 0 \\ 0 & 482 & 0 \\ 0 & 0 & 679 \end{bmatrix} \begin{bmatrix} Y - 16 \\ C_b - 128 \\ C_r - 128 \end{bmatrix} + \begin{bmatrix} 157 \\ 0 \\ 0 \end{bmatrix}$$

$$A = \frac{10}{14} \times 1000(mV) \times 92.5\% \times \frac{4095(LSB)}{1400(mV)} = 1932.6$$

$$S_Y = \frac{10}{14} \times 1000(mV) \times 7.5\% \times \frac{4095(LSB)}{1400(mV)} = 157$$


Copyright © 2011–2016, Texas Instruments Incorporated

***Example 12-7. Full-range RGB to YUV 10:4 Picture Sync Ratio with Zero Setup (DAC Full-swing)***

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = A \bullet \mathbf{M} \bullet \frac{1}{255} \begin{bmatrix} G \\ B \\ R \end{bmatrix} + \begin{bmatrix} S_Y \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{1}{64} \begin{bmatrix} 326 & 63 & 166 \\ -160 & 242 & -82 \\ -286 & -56 & 342 \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix} + \begin{bmatrix} 180 \\ 0 \\ 0 \end{bmatrix}$$

Maximum excursion occurs at yellow of the 100% color bar whose amplitude is calculated as follows:

$$\begin{bmatrix} E'_{Yyellow} \\ E'_{Uyellow} \\ E'_{Vyellow} \end{bmatrix} = \mathbf{M} \begin{bmatrix} E'_G = 100 \\ E'_B = 0 \\ E'_R = 100 \end{bmatrix}$$

$$= \begin{bmatrix} 88.6 \\ -88.6Kb \\ 11.4Kr \end{bmatrix}$$

$$MaxAmp_{yellow} = 0.925\left(Y_{yellow} + \sqrt{U_{yellow}{}^2 + V_{yellow}{}^2}\right) + 7.5$$

$$= 130.83(IRE)$$

$$A = \frac{100(IRE)}{40(IRE) + 131(IRE)} \times 0.925 \times 4095(LSB) = 2215$$

$$S_Y = \frac{100(IRE)}{40(IRE) + 131(IRE)} \times 0.075 \times 4095(LSB) = 180$$

$$Sync = \frac{40(IRE)}{40(IRE) + 131(IRE)} \times 4095(LSB) = 958$$


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.7.2.2 LPF

The YUV data from color space converter is applied to the LPF. The LPF is used to bandlimit the signal to conform to the standard. The LPF comprises of 11 tap symmetric FIR with the programmable coefficients. Independent coefficients are available for luma and chroma. The luma LPF takes the input from CSC in s(12.0) and outputs s(12.0). The chroma LPF takes s(11.0) from the CSC (MSB thrown away) and output s(11.0). The coefficient register format is s(0.7). The sampling rate of the FIR filter is half of clk2x clock. The LPFs are disabled by default and can be activated by YLPF and CLPF registers. LPF is available only for the CVBS generation.

**Table 12-38. Luma LPF Filter Coefficient**

| Coefficients | | Value | Default |
|---|---|---|---|
| h(0) | h(10) | YLPFC0 | 0 |
| h(1) | h(9) | YLPFC1 | 0 |
| h(2) | h(8) | YLPFC2 | 0 |
| h(3) | h(7) | YLPFC3 | 0 |
| h(4) | h(6) | YLPFC4 | 32/128 |
| h(5) | | YLPFC5 | 64/128 |

**Table 12-39. Chroma LPF Filter Coefficient**

| Coefficients | | Value | Default |
|---|---|---|---|
| h(0) | h(10) | CLPFC0 | 0 |
| h(1) | h(9) | CLPFC1 | 0 |
| h(2) | h(8) | CLPFC2 | 0 |
| h(3) | h(7) | CLPFC3 | 12/128 |
| h(4) | h(6) | CLPFC4 | 32/128 |
| h(5) | | CLPFC5 | 40/128 |

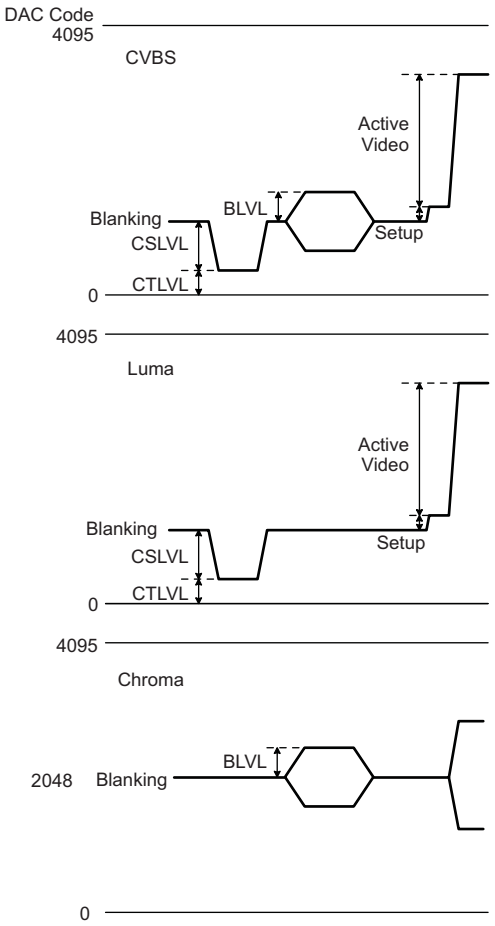

Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.7.2.3 Sync/Burst Insertion

Following LPF, sync waveform is inserted onto the luma. The sync tip level and the sync amplitude are programmable as shown in Figure 12-64. The color burst is inserted onto the chroma. The amplitude of the color burst is specified by CBLVL register as follows:

$$ColorBurst = \begin{cases} CBLVL & \ldots \quad NTSC\ (CCM = 0) \\ \sqrt{2} \times CBLVL & \ldots \quad PAL\ (CCM = 1) \end{cases}$$

**Figure 12-64. CVBS Video Amplitude**




Closed caption and WSS signals are also inserted onto the appropriate lines of luma. The pulse amplitude of CC/WSS bit stream is automatically calculated based on the sync amplitude and the picture sync ratio setting. White 100% level is calculated by the following equation:

$$White_{100\%} = \begin{cases} CSLVL \times 10/4 & \ldots \quad CPSR = 0 \\ CSLVL \times 7/3 & \ldots \quad CPSR = 1 \end{cases}$$

The pulse amplitude of CC/WSS is calculated using the ratio of the white level defined in each standard.

The generic VBI-data is also inserted when used. The detail of the VBI-data insertion is described in Section 12.2.6.2.11.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.7.2.4 QAM Modulation

The VENC generates the sub-carrier by internal direct digital synthesizer (DDS). The frequency of sub-carrier is determined by three increment parameters, SCP0, SCP1 and SCP2 as follows:

$$f_{sc} = \frac{1}{1024} \times \left( SCP0 + \frac{SCP1}{SCP2} \right) \times 27MHz$$

Typical settings of these parameters are shown in Table 12-40.

**Table 12-40. Sub-carrier Increment Parameters**

| TV Format | SCP0 | SCP1 | SCP2 | Frequency |
|---|---|---|---|---|
| NTSC-M/J | 135 | 25 | 33 | 3.579545454… MHz |
| NTSC-4.43 | 168 | 2516 | 16875 | 4.433618749… MHz |
| PAL-B/D/G/H/K/I | 168 | 2516 | 16875 | 4.433618749… MHz |
| PAL-M | 135 | 87 | 143 | 3.575611888… MHz |
| PAL-N | 168 | 2516 | 16875 | 4.433618749… MHz |
| PAL-Nc | 135 | 14391 | 16875 | 3.58205625 MHz |
| PAL-60 | 168 | 2516 | 16875 | 4.433618749… MHz |

The sub-carrier to horizontal (SC-H) phase can be controlled by the user. Writing the SCSD register automatically initializes the sub-carrier phase as the value specified in SCSD. The initialization is occurred at the line 4 in the color field 1 for NTSC and the line 1 in the color field 1 for PAL. The phase resolution of DDS is (1/1024)*360°.

Chroma modulation is performed as expressed by:

$$C = \begin{cases} U \sin(2\pi f_{sc}) + V \cos(2\pi f_{sc}) & \ldots \quad NTSC \\ U \sin(2\pi f_{sc}) \pm V \cos(2\pi f_{sc}) & \ldots \quad PAL \end{cases}$$


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.7.3 SECAM Encoding

##### 12.2.6.2.7.3.1 DbDr Generation

First, SECAM YDbDr need to be generated using the CSC. In this example, the input is assumed to be full-range RGB [0…255] and the output is assumed to have 7:3 picture sync ratio with zero setup.

The fundamental SECAM YDbDr equation is expressed as follows:

$$\begin{bmatrix} E'_Y \\ D'_b \\ D'_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1.505 & 0 \\ 0 & 0 & -1.902 \end{bmatrix} \bullet \mathbf{M_0} \bullet \begin{bmatrix} E'_G \\ E'_B \\ E'_R \end{bmatrix}$$

Where $D'_b$ has a range of 0 to ±1.33343 and $D'_r$ has a range of 0 to ±1.333302 and needs to be scaled so that the resulted values have the corresponding deviation for the FM modulation.

The deviation calculation is explained in Section 12.2.6.2.7.3.3.

$$\begin{bmatrix} Y \\ D_b \\ D_r \end{bmatrix} = \begin{bmatrix} A_Y & 0 & 0 \\ 0 & A_{D_b} & 0 \\ 0 & 0 & A_{D_r} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1.505 & 0 \\ 0 & 0 & -1.902 \end{bmatrix} \bullet \mathbf{M_0} \bullet \frac{1}{255} \begin{bmatrix} G \\ B \\ R \end{bmatrix}$$

$$= \frac{1}{64} \begin{bmatrix} 302 & 59 & 154 \\ -186 & 280 & -94 \\ 286 & 55 & -341 \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix}$$

$$A_Y = \frac{7}{10} \times 1000(mV) \times \frac{4095(LSB)}{1400(mV)} = 2047.5$$

$$A_{D_b} = \frac{0.230(MHz)}{27(MHz)} \times 1024 \times 128 \times \frac{1}{1.33343} = 837.3$$

$$A_{D_r} = \frac{0.280(MHz)}{27(MHz)} \times 1024 \times 128 \times \frac{1}{1.333302} = 1019.4$$

$$Sync = \frac{3}{10} \times 1000(mV) \times \frac{4095(LSB)}{1400(mV)} = 878$$

Where $D_b$ has a range of 0 to ±1117 and $D_r$ has a range of 0 to ±1359.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.7.3.2 Low-Frequency Pre-Emphasis

The YDbDr from CSC is applied to the LPF for a proper attenuation. The LPF function is already described in Section 12.2.6.2.7.2.2. The low-pass filtered Dr and Db are then applied to low-frequency pre-emphasis. The pre-emphasis is implemented by an IIR filter whose transfer function is expressed by:

$$H(z) = \frac{2.75 - 2.5z^{-6}}{1 - 0.75z^{-6}}$$

Then, a limiter clips the DbDr to conform to the maximum deviation requirement specified in SECAM standard. Figure 12-65 shows an example of the limiter configuration.

**Figure 12-65. Limiter Configuration**

$$-350kHz \le \Delta f_{OB} \le 506kHz$$

$$\text{CULCLP} = -\frac{0.35(MHz)}{27(MHz)} \times 1024 \times 128 = -1699$$

$$\text{CUUCLP} = +\frac{0.506(MHz)}{27(MHz)} \times 1024 \times 128 = +2456$$

$$-506kHz \le \Delta f_{OR} \le 350kHz$$

$$\text{CVLCLP} = -\frac{0.506(MHz)}{27(MHz)} \times 1024 \times 128 = -2456$$

$$\text{CVUCLP} = +\frac{0.35(MHz)}{27(MHz)} \times 1024 \times 128 = +1699$$


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.7.3.3 FM Modulation

The low-frequency, pre-emphasized DbDr is then applied to FM modulation. The frequency of the FM modulated sub-carrier is determined by the following equation:

$$f_{FM} = \frac{1}{1024} \times \frac{D_x}{128} \times 27MHz$$

Prior to the modulation, the offsets corresponding to the nominal subcarrier frequency of Db and Dr are added to the low-frequency, pre-emphasized DbDr. The Db and Dr offsets are set by the sub-carrier parameter register SCP1 and SCP2, respectively, as shown in Figure 12-66.

**Figure 12-66. Sub-carrier Parameter Registers**

$$f_{OB} = 4.25MHz$$

$$\text{SCP1} = \frac{4.25(MHz)}{27(MHz)} \times 1024 \times 128 = 20632$$

$$f_{OR} = 4.40625MHz$$

$$\text{SCP2} = \frac{4.40625(MHz)}{27(MHz)} \times 1024 \times 128 = 21390$$

The amplitude of the FM modulated sub-carrier is specified by the CBLVL register. While the CBLVL register specifies the burst amplitude for NTSC/PAL mode, it works differently in SECAM as the sub-carrier amplitude.

For color synchronization, the VENC does not support field identification, but line identification.

The phase of the sub-carrier is reset every line.

#### 12.2.6.2.7.3.4 Bell Filter

A high-frequency pre-emphasis is performed on the FM modulated sub-carrier. The filter has a bell characteristic expressed by:

$$H(z) = \frac{8 - 2.25z^{-4} - 5.5z^{-6}}{1 - 0.5z^{-2}}$$

Finally, SECAM chroma is available from the bell filter output.

#### 12.2.6.2.7.4 Luma Delay

The Y signal delay on the CVBS path can be adjusted by the CYDLY register. It adjusts the Y delay right before mixing Y and C to generate CVBS. The adjustable range is from –8 to 7 cycle of clk2x. S-Video Y is not affected.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.8 WSS

The VENC can insert 14-bit video information (WSS) on line 23 of every frame to conform to the ITU-R BT.1119-2 (ETSI EN 300 294) Wide Screen Signaling specification for the 625i system. The 14-bit transmitted data is specified in the 20-bit WSS_DATA register. The WSS_EN register enables attribute insertion. Figure 12-67 shows the WSS_DATA register usage.

**Figure 12-67. WSS_Data Register Usage**

| 19 14 | 13 11 | 10 8 | 7 4 | 3 0 |
|---|---|---|---|---|
| Unused | GROUP4 | GROUP3 | GROUP2 | GROUP1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Transmission order is LSB first, that is, WSS_DATA[0] is transmitted first then WSS_DATA[1] follows.

The amplitude of the WSS symbol is automatically determined as five-seventh (5/7) of white 100%. As described earlier, the white level is calculated by the sync amplitude and picture-sync ratio registers (CSLVL and CPSR for CVBS).

#### 12.2.6.2.9 CGMS

The VENC can insert 14-bit CGMS data on line 20 and line 283 to conform to the EIAJ CPR-1024 for 525i. The 14-bit VID data and the corresponding 6-bit CRC are set in the 20-bit WSS_DATA register. The CRC should be calculated by user based on the following equation.

$G(x) = x^6 + x + 1$, where $x^6$, x are preset to 1

The WSS_EN register enables attribute insertion. Figure 12-68 shows the WSS_DATA register usage.

**Figure 12-68. WSS_Data Register Usage**

| 19 14 | 13 6 | 5 2 | 1 0 |
|---|---|---|---|
| CRC | WORD2 | WORD1 | WORD0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Transmission order is LSB first, that is, WSS_DATA[0] is transmitted first and then WSS_DATA[1] follows. WSS_DATA and WSS_EN registers are exclusively used for WSS and CGMS insertion depending on the timing format.

The amplitude of CGMS data is automatically determined as 70% of white (70IRE).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.10 Closed Caption

The VENC supports closed caption encoding. Closed caption data is transmitted on the line 21 of the odd (top) field and the line 284 of the even (bottom) field in 525i mode. It is possible to specify the fields on which closed captioning is enabled by L21EN register.

Closed caption encoding for 625i system is also supported. Closed caption is inserted on line 22 of the odd field and line 335 of the even field for 625i systems.

The data should be written to the L21DO or L21DE register for odd or even fields, respectively. When data is written to L21DO, CAOST read status bit is changed to 1. When data is written to L21DE, CAEST becomes 1. These bits are automatically cleared to 0 when a caption data transmission is completed on the line 21 in odd field or the line 284 in even field (line 22 and 335 for 625i).

When the caption data register (L21DO or L21DE) is not written before the caption data transmission timing of the corresponding field, an ASCII code specified by L21DF register is used for transmission. A null character should be typically set to L21DF register.

The width of each data register is 8 bits, which consists of 7-bit data and the parity bit. It is your responsibility to calculate the parity bit. The amplitude of closed caption pulse is automatically determined as 50% of white (50IRE).

#### 12.2.6.2.11 Raw VBI-data Insertion Mode

The VBI waveform which is not supported by VENC hardware can be inserted using raw VBI-data insertion mode. In this mode, the VBI waveform is synthesized by software according to the waveform requirements of the selected data type. The VBI_IF module sends VBI waveform to the VENC at the specified VBI lines.

The raw VBI-data insertion mode is enabled VBIEN register is set to 1. Otherwise, the VBI_IF module is basically in a reset state.

The VENC asserts venc_vbi_req pulses to the VBI_IF module during VBI period. The horizontal timing of venc_vbi_req can be configured by VBI_H_STA and VBI_H_STP registers. The reference timing of these registers is the internal TV counter. The vertical timing is determined by the AV window position which is specified by AV_V_STA and AV_V_STP registers. The timing is shown in Figure 12-69 and Figure 12-70.

The VBI_IF module keeps tracks of VBI line numbers by counting the number of venc_vbi_req pulses. On the specified VBI line number, the VBI_IF will send the samples from the payload to the VENC with vbi_venc_val set to HIGH. The data width of vbi_venc_data is only 8bit so zeros are padded to the 4 LSBs when inserted.

Note that CC/WSS and Macrovision have the priority over the VBI data when VBI data insertion is activated on the lines which have these pulses. The vbi_venc_val is disregarded at the lines CC/WSS and Macrovision are enabled.

#### 12.2.6.2.12 Macrovision

The VENC supports analog copy protection Macrovision Rev7.1 to the SDTV CVBS and S-Video output. Due to strict security control by Macrovision, any information such as Macrovision register map, usage, and so on, are not directly distributed to the customers. These information are supposed to be given by Macrovision upon a certain certification.

#### 12.2.6.2.13 Register Setting for Various Video Standards

Table 12-41 is a list of the typical register setting for various video standards.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-69. VBI I/F Horizontal Timing**

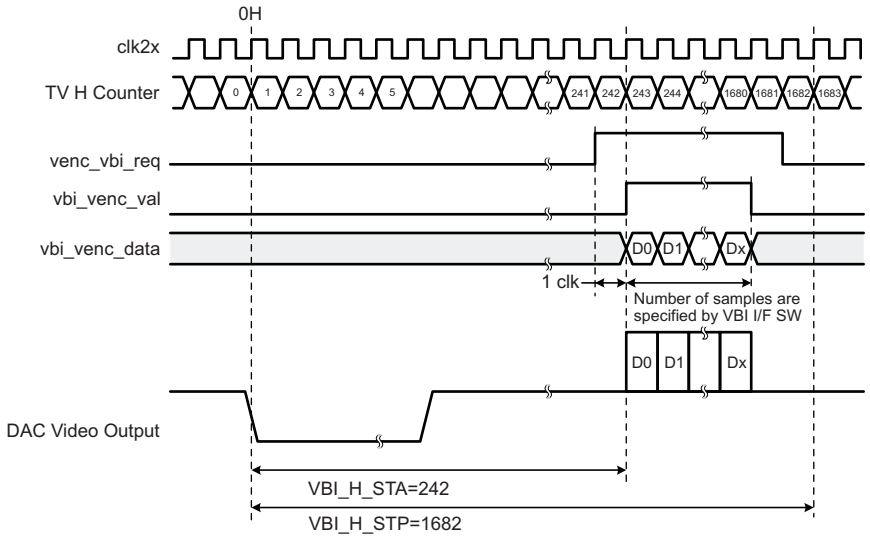
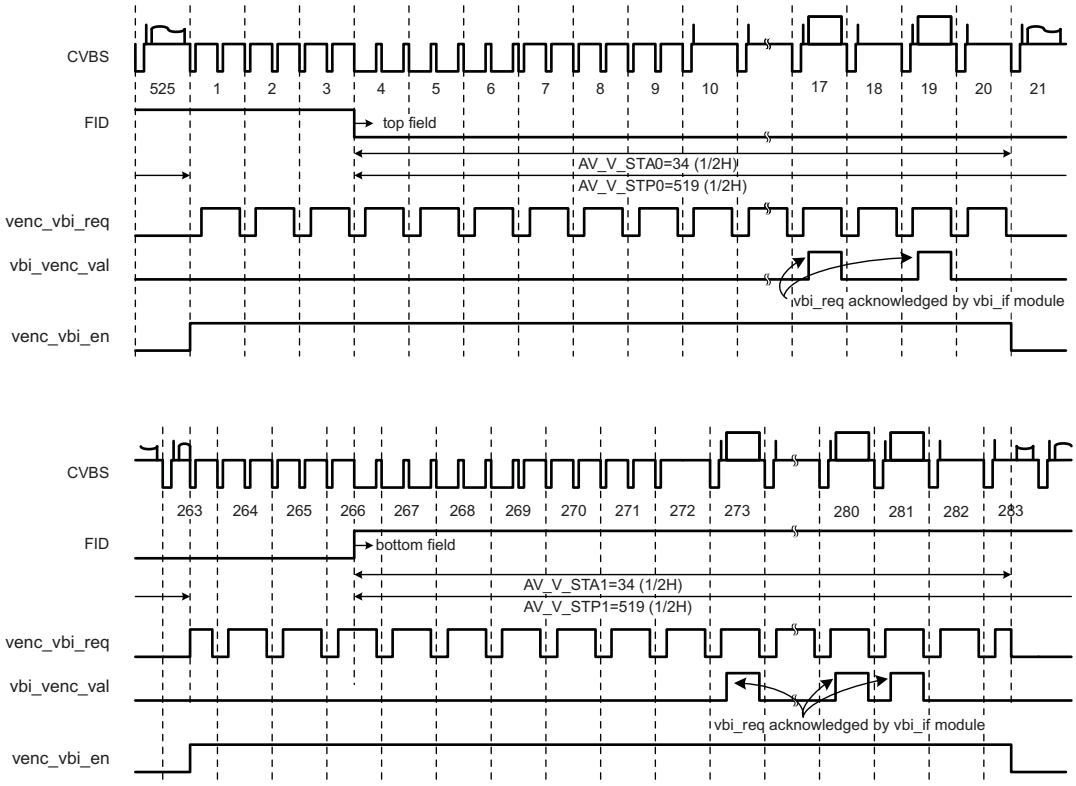



**Figure 12-70. VBI I/F Vertical Timing**





Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-41. Typical Register Setting for Various Video Standards**

| Parameter | Register Name | Location | SDTV | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | NTSC-M | NTSC-4.43 | PAL | PAL-M | PAL-N | PAL-Nc | PAL-60 | SECAM |
| Horizontal Interval | HITV | 0x12[12:0] | 1716 | 1716 | 1728 | 1716 | 1728 | 1728 | 1716 | 1728 |
| Vertical Interval | VITV | 0X12[28:16] | 525 | 525 | 625 | 525 | 625 | 625 | 525 | 625 |
| Scan Mode | ITLC | 0x4[2] | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Scan Format | FMT | 0x84[11:8] | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| SDTV Vertical Sync Width | SVSW | 0x84[3] | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| Horizontal Start of Active Video | AV_H_STA | 0x88[12:0] | 244 | 244 | 264 | 244 | 265 | 265 | 244 | 265 |
| Horizontal Stop of Active Video | AV_H_STP | 0x88[28:16] | 1684 | 1684 | 1704 | 1684 | 1704 | 1704 | 1684 | 1704 |
| Vertical Start Line of Active Video | AV_V_STA0 | 0x8c[12:0] | 34 | 34 | 45 | 34 | 45 | 45 | 34 | 45 |
| Vertical Stop Line of Active Video | AV_V_STP0 | 0x8c[28:16] | 519 | 519 | 620 | 519 | 620 | 620 | 519 | 620 |
| Vertical Start Line of Active Video | AV_V_STA1 | 0x90[12:0] | 34 | 34 | 45 | 34 | 45 | 45 | 34 | 45 |
| Vertical Stop Line of Active Video | AV_V_STP1 | 0x90[28:16] | 519 | 519 | 620 | 519 | 620 | 620 | 519 | 620 |
| Horizontal Burst Start | BST_H_STA | 0x94[8:0] | 143 | 143 | 151 | 156 | 151 | 151 | 143 | 157 |
| Horizontal Burst Stop | BST_H_STP | 0x94[24:16] | 211 | 211 | 212 | 225 | 212 | 219 | 211 | 264 |
| CVBS Sync Tip Level | CTLVL | 0x9c[11:0] | 188 | 188 | 148 | 188 | 188 | 148 | 148 | 148 |
| CVBS Sync Amplitude | CSLVL | 0x9c[27:16] | 836 | 836 | 878 | 836 | 836 | 878 | 878 | 878 |
| CVBS Picture Sync Ratio | CPSR | 0xa0[0] | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| Color Modulation Mode | CCM | 0xa0[5:4] | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 2 |
| Burst Level | CBLVL | 0xa0[27:16] | 418 | 418 | 310 | 295 | 295 | 310 | 310 | 480 |
| Sub-carrier Increment Integer | SCP0 | 0x148[23:16] | 135 | 168 | 168 | 135 | 168 | 135 | 168 | - |
| Sub-carrier Increment Numerator | SCP1 | 0x14C[15:0] | 25 | 2516 | 2516 | 87 | 2516 | 14391 | 2516 | 20632 |
| Sub-carrier Increment Denominator | SCP2 | 0x14C[31:16] | 33 | 16875 | 16875 | 143 | 16875 | 16875 | 16875 | 21390 |



Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.6.2.14 DAC Output

##### 12.2.6.2.14.1 DAC Output Level

The VENC has two channel 12-bit digital outputs for DAC input. The DAC output code can be optionally inverted by the DAIV register.

##### 12.2.6.2.14.2 DAC Output Configuration

It is your choice to specify which DAC outputs which video signal. DA0S-DA1S registers specify the DAC output signal as shown in Table 12-42.

**Table 12-42. DAC Output Select**

| DAxS | DAC Output |
|---|---|
| 0 | CVBS |
| 1 | S-Video Y |
| 2 | S-Video C |
| 3-15 | Reserved |

##### 12.2.6.2.14.3 DAC Oversampling

The VENC is capable of 2x oversampling for the final DAC output. The 2x DAC oversampling eases the external analog filter cost as it can eliminate the unwanted image around the sampling frequency of clk2x. Setting DAUPS enables 2x oversampling. The oversampling filter is comprised of a 27 tap symmetric FIR filter with the programmable coefficients. The transfer function is expressed as:

$$H(z) = \sum_{n=0}^{26} h(n)\ z^{-n}$$

The clk4x is used in the oversampler as a DAC sampling clock. Its frequency should be double of clk2x if the hardware is configured to support 2x oversampling. Otherwise, the clk4x should be identical to clk2x. Whether or not to support 2x oversampling can be configured by INC_DUPS parameter in the Verilog HDL.

**Table 12-43. DAC 2x Oversampling Filter Coefficients**

| Coefficients | Value | Default |
|---|---|---|
| h(0), h(26) | DUPFC0 | 0 |
| h(1), h(25) | 0 | 0 |
| h(2), h(24) | DUPFC1 | 0 |
| h(3), h(23) | 0 | 0 |
| h(4), h(22) | DUPFC2 | 0 |
| h(5), h(21) | 0 | 0 |
| h(6), h(20) | DUPFC3 | 0 |
| h(7), h(19) | 0 | 0 |
| h(8), h(18) | DUPFC4 | 6/128 |
| h(9), h(17) | 0 | 0 |
| h(10), h(16) | DUPFC5 | -20/128 |
| h(11), h(15) | 0 | 0 |
| h(12), h(14) | DUPFC6 | 78/128 |
| h(13) | 1 | 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

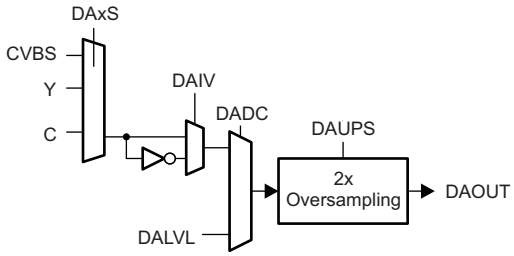
#### 12.2.6.2.14.4 DAC Power Down

DAC power down is managed by the DA0E-DA1E registers. The corresponding DAC should be enabled by setting these registers to 1. Otherwise, DAC should be in power down state. Both DACs can be power down independently. By default, these registers are set to 0 to disable all DACs.

#### 12.2.6.2.14.5 DAC DC Output Mode

A test mode to output the specified DC level on the DAC output pins is available. Setting DADC register to 1 switches DAC output from normal video signal to DALVL register. In this mode, both DACs have same DC output of DALVL D/A converted value.

**Figure 12-71. DAC I/F Logic**





Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.7 High-Definition Video Encoder (HD_VENC)

The HD VENC module encodes the active video data from COMP module and sends it to 30-bit wide DVO (Digital Video Output) port by generating timing signals to control the data flow.

#### 12.2.7.1 Features Supported

- The DVO port will support the output data rate up to 1080p formats (148.5 MHz × 30bits)
- The DVO support 10/20/30-bit wide output with embedded sync or discrete sync
- Fully programmable DVO timing generation (VS,HS,FID) to support any format flat display panel
- Fully programmable internal timing generation - VS, HS, FID, HBI, VBI, and ACT_VID for internal video compositor
- Programmable Color Space Converter: 10-bit data path.
- Gamma correction table

#### 12.2.7.2 Functional Overview

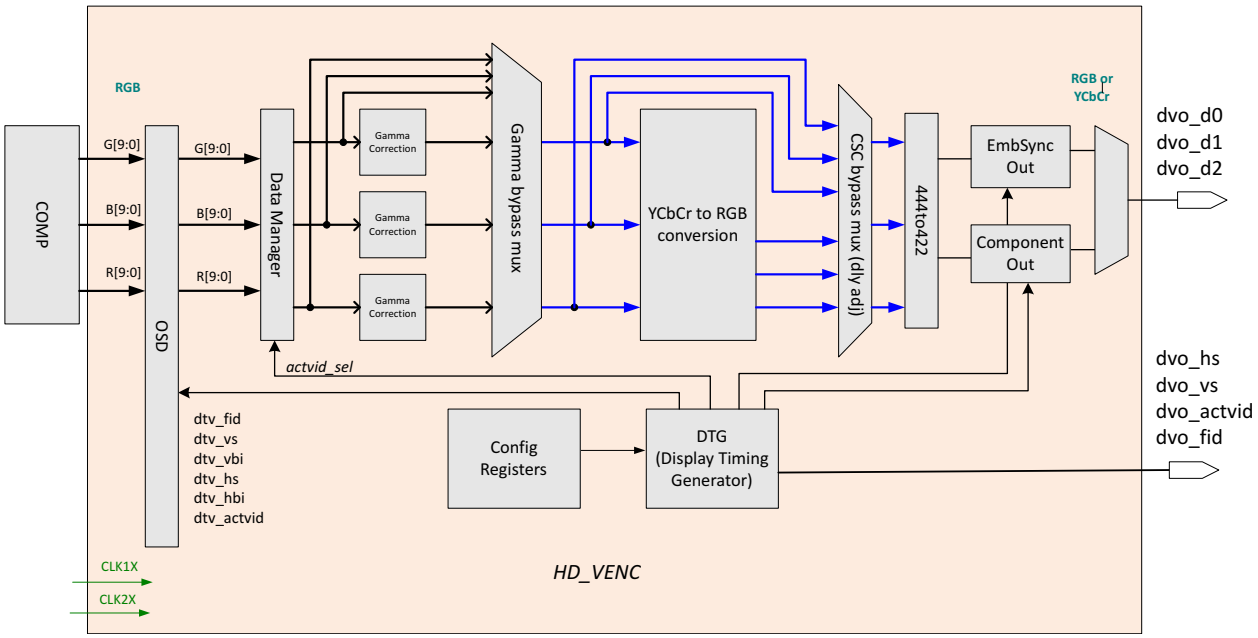
HD VENC primarily has two blocks: OSD and encoder. The OSD module is used to get the data from the COMP module using the sync signals generated by encoder module. In the device, HD VENC supports either digital (HD_VENC_G [HDMI/DVO1] or HD_VENC_D[DVO2]).

Table 12-44 lists the possible display ports and corresponding VENC names. The basic block diagram of HD VENC module is shown in Figure 12-72.

**Table 12-44. Display Ports and VENC Names**

| Display Port Name | Type | HDVENC Used | Port Name for PINMUX | Features Supported |
|---|---|---|---|---|
| HDMI/DVO1 | Digital | HD_VENC_D | VOUT1 | All features in digital data flow path |
| DVO2 | Digital | HD_VENC_D | VOUT0 | All features in digital data flow path |
| SDVENC | Analog | SD_VENC | Not Applicable | Please check SD VENC userguide for details. |

**Figure 12-72. HD VENC Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.7.3 Video Data Interface to OSD

#### *12.2.7.3.1 OSD Interface Overview*

The interface between OSD and video display encoder are consisting of a 30-bit data bus, a pixel clock and five sync signals. By default 30-bit RGB video data will be passed from OSD to encoders on every rising edge of the pixel clock during active video period. The maximum frequency of the pixel clock is 148.5 MHz.

Table 12-45 lists all the control signals of this interface.

**Table 12-45. OSD Interface Signals**

| Signal Names | Descriptions |
|---|---|
| dtv_fid | Field ID signal. This signal will toggle between “1” and “0” on every field in interlace mode. It will toggle between “1” and “0” on every frame in progressive mode.<br>Programmability: controlled by register CFG13 (END_F1) and CFG10 (LINES). |
| dtv_vs | Vertical sync signal. This signal is a one-line long pulse. This pulse indicates the first line of each field in interlace mode. In progressive mode, it indicates the first line of each frame.<br>Programmability: location of this signal is controlled V_BLANK1, VBLANK2, and FD1. |
| dtv_vbi | Vertical blank interval signal. This signal goes to “1” during non-active video period; it stays at “0” during active video period.<br>Programmability: controlled by register CFG16/17. (For OSD interface only) |
| dtv_hs | Horizontal sync signal. This is the horizontal sync signal.<br>Programmability; the start location and width of this signal are both programmable. They are controlled by CFG17. (For OSD interface only) |
| dtv_hbi | This is a four-pixels-wide and active-high signal. It appears once every video line.<br>Programmability: The location of this pulse is programmable. (For OSD interface only) |
| dtv_actvid | This is active-video qualification signal. When this signal is high, encoder is expecting active video data output from OSD after one clock delay.<br>Programmability: The fully programmable. (For OSD interface only) |

**Note:** The dtv_vs, dtv_vbi, dtv_hs and dtv_hbi signals are generated by the encoder but are not used by the HDVPSS design. Therefore, all parameters related to these signals do not need to be specified.


Copyright © 2011–2016, Texas Instruments Incorporated

The following figures illustrate the protocol of this interface. Figure 12-73 illustrates all the signals that are related to frame/field sync.

**Figure 12-73. Vertical Sync Signals for OSD**

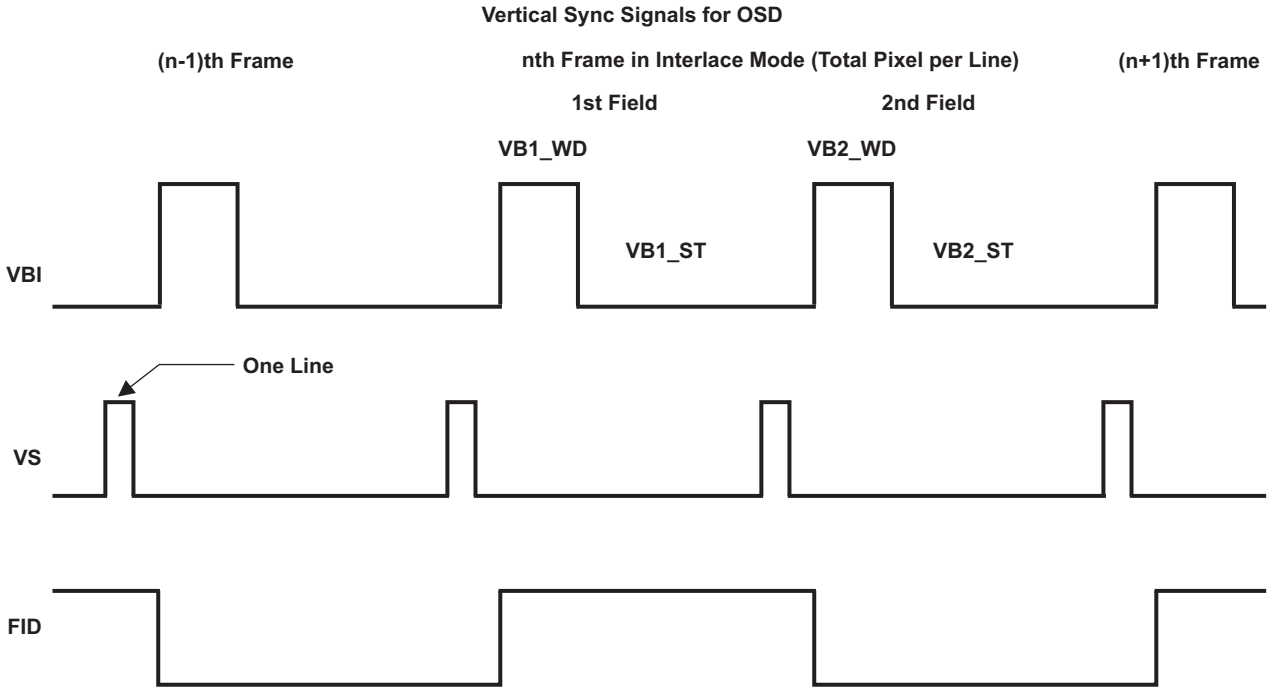
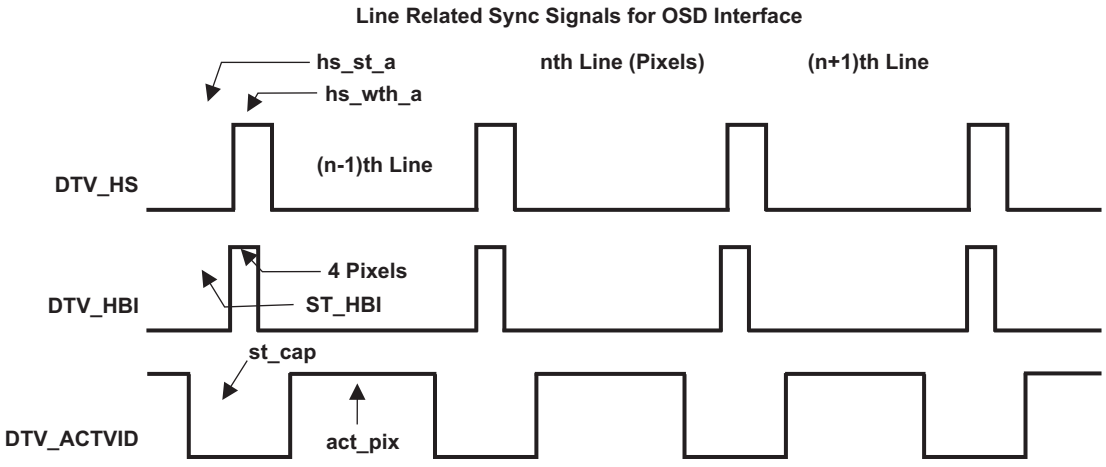



Figure 12-74 illustrates all the signals that are related to line sync. The programmable parameters with corresponding register names are indicated in the diagram.

**Figure 12-74. Horizontal Sync-related Signal for OSD**



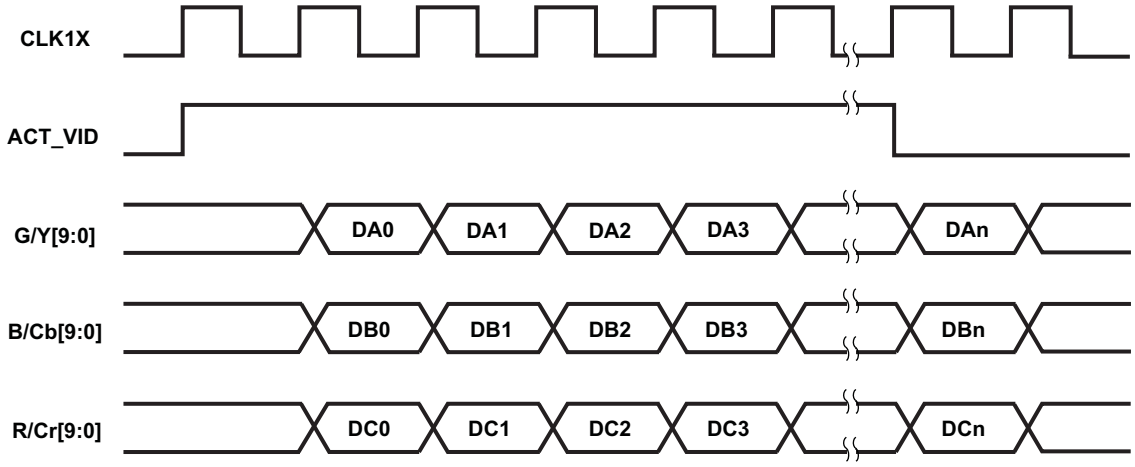


Copyright © 2011–2016, Texas Instruments Incorporated

Figure 12-75 is the OSD-to-encoder data bus timing diagram. ACT_VID is the active video data qualification signal. Encoder will capture data from OSD after one clock delay when the ACT_VID goes to “high”.

The input 30-bit video data can also be in YCbCr 444 format. On chip processor will set the register bit “Y_RGBn” in CFG0 register to select RGB or YCbCr color spaces.

**Figure 12-75. Video Data Interface Between OSD and Encoder**





Copyright © 2011–2016, Texas Instruments Incorporated

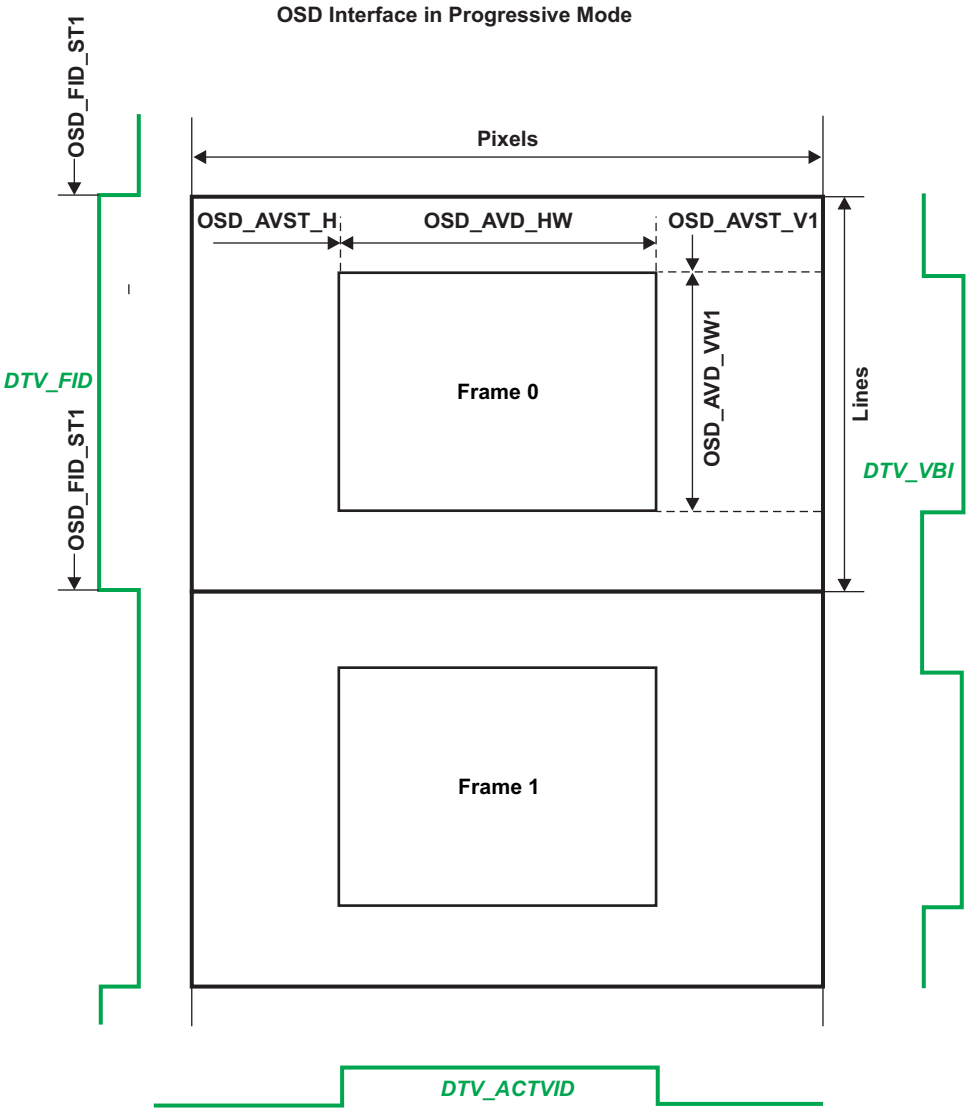
#### 12.2.7.3.2 OSD Interface Configuration for Progressive Display

Figure 12-76 illustrates the OSD interface in the configuration for progressive display mode.

The corresponding register settings are described in the register section.

The green color signals are the signals outputting to OSD.

**Figure 12-76. OSD Interface in Progressive Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

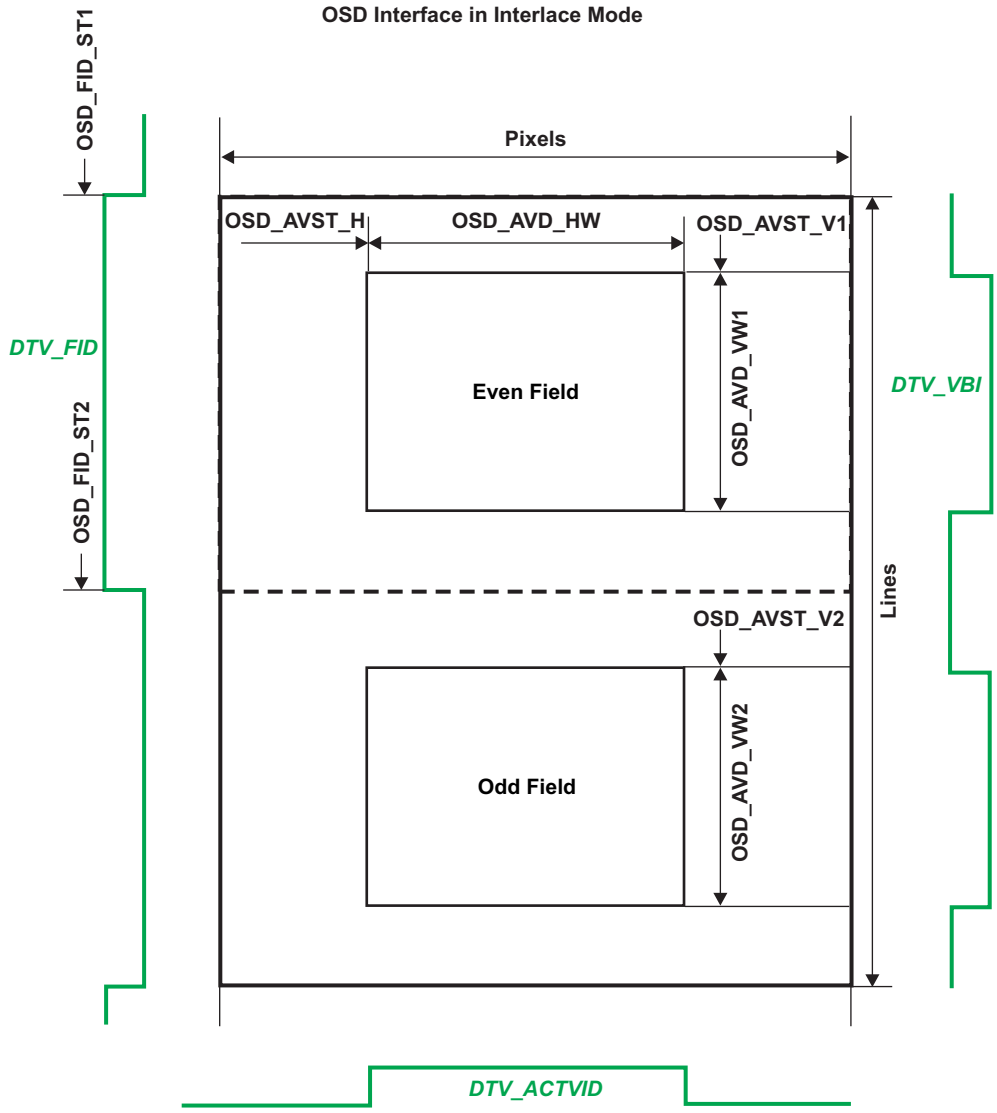
#### 12.2.7.3.3 OSD Interface Configuration for Interlace Display

Figure 12-77 illustrates the OSD interface in the configuration for interlace display mode.

The corresponding register settings are described in the register section.

The green color signals are the signals outputting to OSD.

**Figure 12-77. OSD Interface in Interlace Mode**






Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.7.4 Display Timing Generation (DTG)

The DTG (Display Timing Generator) block will generate all the sync and display timing signals for both OSD interface and DVO interface.

Due to the fast development of display panel technology and lack of the interface standard, the sync timing for DVO will be designed highly flexible. This will make sure that the device can interface to the most panels used by digital TV industry; meanwhile, maintains the simplicity for the software programming.

The details of OSD interface and DVO interface are described in the corresponding sections.

### 12.2.7.5 DVO (Digital Video Output)

#### *12.2.7.5.1 Output Formats of DVO*

Table 12-46 lists all formats that the DVO supports.

**Table 12-46. DVO Formats**

| Formats | DVO_D0 | DVO_D1 | DVO_D2 |
|---|---|---|---|
| Single-stream 656 | YCbCr | (Unused) | (Unused) |
| Dual-stream 656 | Y | CbCr | (Unused) |
| Tri-stream 656 | Y/G | Cb/B | Cr/R |
| YUV422 20bit output with discrete sync | Y | CbCr | (Unused) |
| 30bit output with discrete sync | Y/G | Cb/B | Cr/R |

During the two discrete sync modes, following are the sync signals that DVO outputs:

1. DVO_HS
2. DVO_VS
3. DVO_ACTVID
4. DVO_FID

#### *12.2.7.5.2 Embedded Sync Word Definitions of DVO*

DVO of the device supports single/dual/tri stream with embedded sync (SAV/EAV).

The active video data in single/dual stream formats are in 4:2:2 YCbCr format.

DVO of the device also supports 10-bit video, the embedded sync SAV/EAV in 10bit format as: 3FF, 000, 000, XXX. All the possible values for XXX are listed in Table 12-47.

**Table 12-47. Definition of SAV and EAV words**

| BITS | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | SAV/EAV | Hex |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Function | 1 | F | V | H | P3 | P2 | P1 | P0 | x | x | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SAV | 200 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | EAV | 276 |
| 2 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | SAV | 2AC |
| 3 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | EAV | 2D8 |
| 4 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | SAV | 31C |
| 5 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | EAV | 368 |
| 6 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | SAV | 3B0 |
| 7 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | EAV | 3C4 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.7.5.3 DVO Interface Configuration

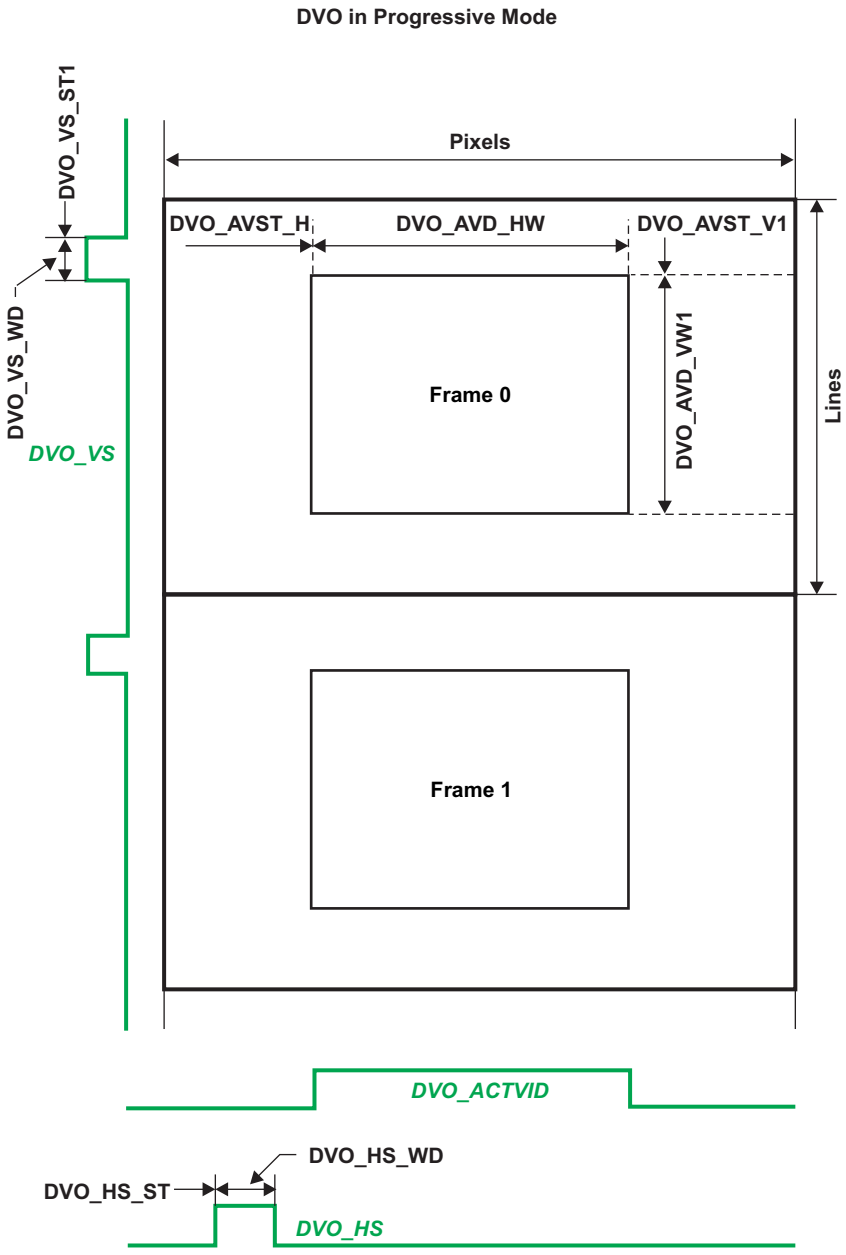
The design goal of the DVO of the device is to interface to all standard and non-standard panels seamlessly. To archive this goal, the DVO of the device has been refined with ultimate flexibility; meanwhile maintaining simplicity for the software programming. Following subsections present all the interface signals for both progressive and interface display devices.

##### 12.2.7.5.3.1 DVO Interface for Progressive Display

The progressive display format is the display format for most flat panel display devices, such LCD, Plasma, and DLP. Figure 12-78 illustrates the progressive configuration of DVO port. The configuration register settings are described in the register section.

The green color signals are the signals outputting from the DVO port.

**Figure 12-78. Progressive Configuration of DVO**





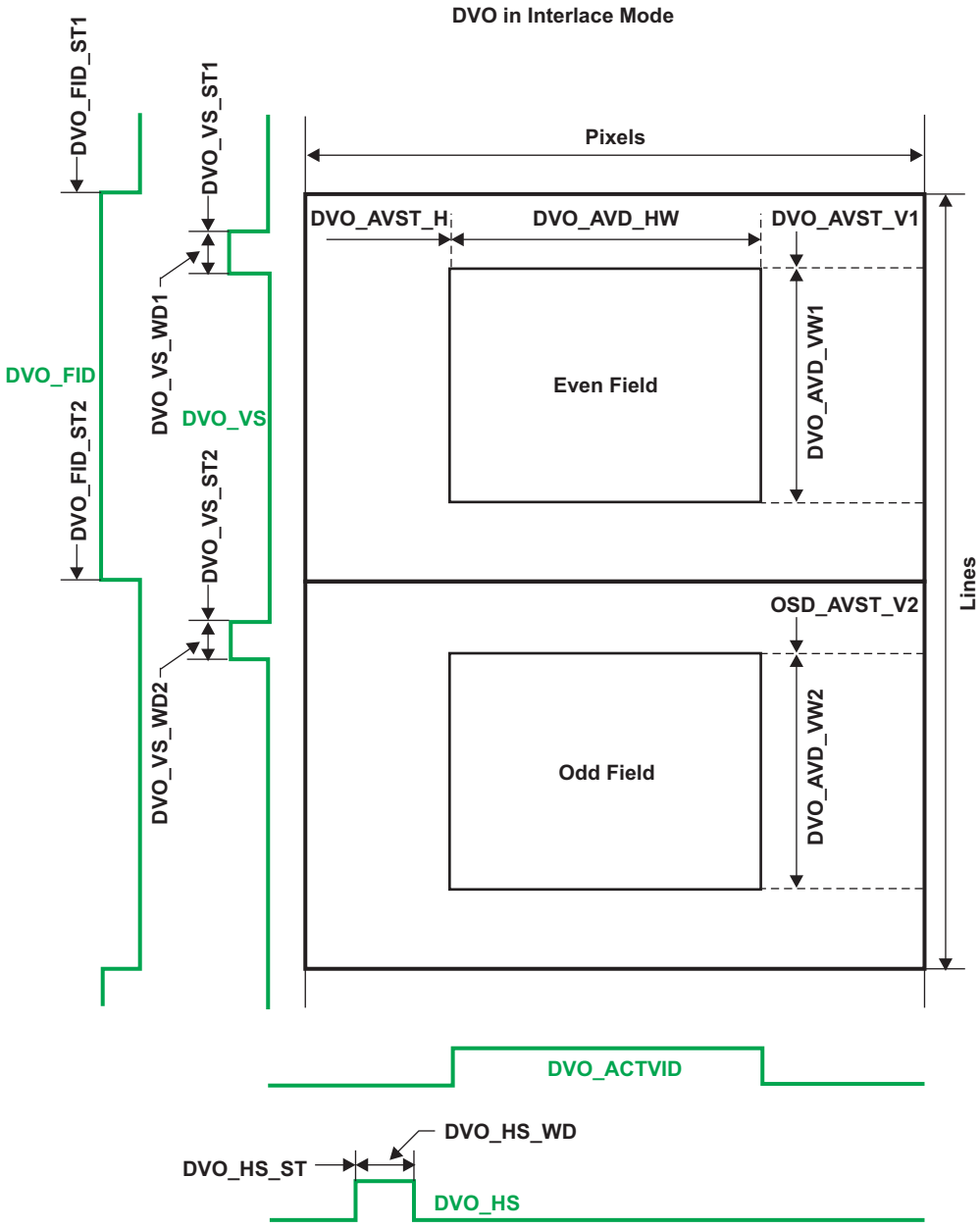
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.7.5.3.2 DVO Interface for Interlace Display

Figure 12-79 illustrates the interlace configuration of DVO port. The configuration register settings are described in the register section.

The green color signals are the signals outputting from the DVO port.

**Figure 12-79. Interlace Configuration of DVO**





Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.7.6 JEIDA Format of DVO

A new output format of DVO has been added and a corresponding control bit “JED” has also been added.

This bit is used to make DVO interfacing to certain LCD panels seamlessly. Table 12-48 illustrates the JEIDA and VESA (the normal/default format we use) format.

**Table 12-48. JEIDA and VESA Format**

| VESA Mode | | JEIDA Mode | |
|---|---|---|---|
| **RGB Data** | **DVO Port** | **RGB Data** | **DVO Port** |
| G/B/R[9] | DV_OUT0/1/2[9] | G/B/R[3] | DV_OUT0/1/2[9] |
| G/B/R[8] | DV_OUT0/1/2[8] | G/B/R[2] | DV_OUT0/1/2[8] |
| G/B/R[7] | DV_OUT0/1/2[7] | G/B/R[9] | DV_OUT0/1/2[7] |
| G/B/R[6] | DV_OUT0/1/2[6] | G/B/R[8] | DV_OUT0/1/2[6] |
| G/B/R[5] | DV_OUT0/1/2[5] | G/B/R[7] | DV_OUT0/1/2[5] |
| G/B/R[4] | DV_OUT0/1/2[4] | G/B/R[6] | DV_OUT0/1/2[4] |
| G/B/R[3] | DV_OUT0/1/2[3] | G/B/R[5] | DV_OUT0/1/2[3] |
| G/B/R[2] | DV_OUT0/1/2[2] | G/B/R[4] | DV_OUT0/1/2[2] |
| G/B/R[1] | DV_OUT0/1/2[1] | G/B/R[1] | DV_OUT0/1/2[1] |
| G/B/R[0] | DV_OUT0/1/2[0] | G/B/R[0] | DV_OUT0/1/2[0] |

The JEIDA mode is only used in three-channel RGB with discrete sync output format.

### 12.2.7.7 VBI Data Service

Following are the commonly used VBI-data-services on analog component video interface, these functions are supported by the hardware inside the encoder.

- Type-A and Type-B data packets based on CEA-805-A standard, which include CGMS-A/APS(Analog Protection System)/AFD(Active Format Description)/RCI(Re-distribution Control Information)
- International VBI data support based on the international standard IEC-62375, which is mainly for the applications outside of north-American.

A typical VBI data package usually consists of three segments, they are illustrated in Figure 12-80.

**Figure 12-80. Typical VBI Data Package**

Each segment has its-own programmable symbol. Number of symbol for each segment is also programmable. The minimum element of a symbol is one pixel clock. For different display formats, different pixel clocks have been used. All of these require the VBI data encoder can be used for different display interface formats.

Two programmable VBI data encoders have been designed to service all the above functional requirement. These VBI data line encoders are symbol based, it can accommodate different symbol modulations.

- The first VBI line will be used for type-A data packet of CEA-805-A standard or IEC-62375 standard.
- The second VBI line will be used for type-B data packet of CEA-805-A standard


Copyright © 2011–2016, Texas Instruments Incorporated

Table 12-49 shows the first line encoder's programmable capacity. The payload symbol can be programed in the phase-modulation representation. It can be used for type-A VBI data service of CEA-805-A or IEC-62375.

**Table 12-49. Programmable Capacity of First Line Encoder**

| Start Symbol | Header Symbol | Payload Symbol |
|---|---|---|
| Max. = 64 | Max. = 64 | Max. = 128 |

Table 12-50 shows the second line encoder's programmable capacity, it can be used for type-B VBI data service of CEA-805-A. The payload has been increased to accommodate large data packets.

**Table 12-50. Programmable Capacity of Second Line Encoder**

| Start Symbol | Header Symbol | Payload Symbol |
|---|---|---|
| Max. = 64 | Max. = 64 | Max. = 128 |

Although these two line encoders are targeting to some specific VBI applications, but they are flexible enough to be used by some other applications. For example, both encoder data-lines can be transmitted on any line in a frame or a field (by set the line number to register VBI_L1).

These VBI data line encoders can support up to two VBI lines in a frame or a field. If the host-CPU updates the VBI data registers at the frame rate (or field rate), the encoders are able to send out two VBI lines per frame or per field.

### 12.2.7.8 Processing Blocks

The main processing blocks in encoder module are:

- Gamma Correction (Gamm)
- Color Space Conversion (CSC)

#### *12.2.7.8.1 Gamma Correction (Gamm)*

The gamma correction modules perform a gamma or color correction independently for each color in RGB color space by using configurable LUT (Look-Up Table) in RAM. Each Gamm module reads 10-bit input and outputs 10-bit value from its corresponding Look-up Table entry. The RAM has 1024 32-bit words and contains three LUTs for each color as shown in Table 12-51.

**Table 12-51. Gamma Correction LUT**

| | bit[9:0] | bit[19:10] | bit[29:20] | bit[31:30] |
|---|---|---|---|---|
| Address at 0x0000 | LUT_G[0] | LUT_B[0] | LUT_R[0] | reserved, return 0 during read |
| Address at 0x0004 | LUT_G[1] | LUT_B[1] | LUT_R[1] | reserved, return 0 during read |
| Address at 0x0008 | LUT_G[2] | LUT_B[2] | LUT_R[2] | reserved, return 0 during read |
| ... | ... | ... | ... | ... |
| Address at 0x0FFC | LUT_G[1023] | LUT_B[1023] | LUT_R[1023] | reserved, return 0 during read |

- If CFG0.RD_MEM = '1', processor can read and write these LUTs as regular memory.
- If CFG0.RD_MEM = '0', processor can only write to these LUTs as regular memory.


Copyright © 2011–2016, Texas Instruments Incorporated

Table 12-52 shows the memory map address of configuration register and the LUT memories.

**Table 12-52. Memory Map Addresses for Configuration Register and LUT Memories**

| Descriptions | Start Address[12:0] | End Address[12:0] |
|---|---|---|
| HD VENC MMR | 0x0000 | 0x0FFF |
| LUT RAM | 0x1000 | 0x1FFF |

Gamm module can be bypassed by setting CFG0.BYPS_GC bit-field to 1.

#### *12.2.7.8.2 Color Space Conversion (CSC)*

The color space conversion (CSC) module is used to convert the input data from one color space to another by using a programmable 3 × 3 matrix. See the *Color Space Converter* section for more details. The CSC module can be bypassed by setting CFG0.BYPS_CS bit-field to 1.


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.8 VIP Parser

The VIP Parser module is used to capture the video data into the HDVPSS module.

#### 12.2.8.1 Features

The device has two identical instances of the VIP subsystem. Each VIP instance has one VIP parser.

For one VIP Parser instance, the video capture requirements include:

- Two Pixel Clock Input Domains are supported (Port A and Port B)
  - Each Pixel Clock Input Domain has separate clock and framing signals.
  - Each Pixel Clock Input Domain can support embedded (BT.656 style) or discrete (BT.601 style) sync.
  - Pixel Clock Input Domain Port A supports up to one 24-bit input data bus, including BT.1120 style embedded sync for 16 and 24 bit data.
  - Pixel Clock Input Domain Port B supports one 8-bit input data bus. At the SOC level, the same device pins may be shared between Ports A and B. One 24 bit chip level set of device pins is shared. Port A has access to all 24 bits. Port B only has access to 8 bits.
- Embedded Sync data interface mode supports single or multiplexed sources
- Discrete Sync data interface mode supports only single source input
- The two Pixel Clock Input Domains can be individually configured in any combination of Embedded or Discrete Sync
- Vertical ancillary data is supported for each input source. For discrete sync modes where vertical ancillary data is not specified using signals, vertical ancillary data will appear in active video buffer.
- A maximum of 8 + 1 (8 normal line sources + 1 split-line source) multiplexed sources are supported for a single Pixel Clock Input Domain using TI Line Mux Mode
- Multiplexed data can only appear in embedded sync mode
- Where possible, blanking pixels that may contain embedded vertical ancillary data will be stored in a dedicated buffer per each video source
- Optional selection of channel (Luma or Chroma or both) from which Vertical Ancillary data is extracted for YUV422 source
- For RGB source, Vertical Ancillary data can be found in one of the R, G, or B channels. The VIP Parser can select the channel from which Vertical Ancillary data is extracted.
- Ancillary Data can appear in the Horizontal Blanking as well as the Vertical Blanking. Typically, only Vertical Blanking Ancillary Data is captured. However, Horizontal Blanking Ancillary Data can be captured as well using HSYNC style discrete sync capture mode.
- Video up to WUXGA (1920 × 1200) can be supported using Port A in 16- or 24-bit mode.

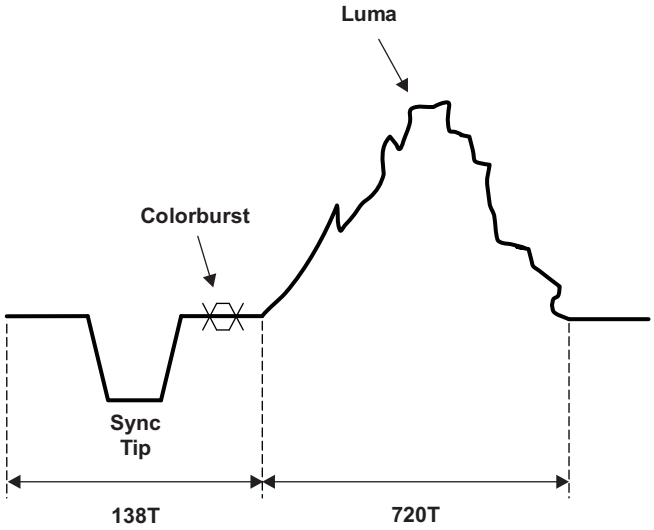
#### 12.2.8.2 Functional Description

##### 12.2.8.2.1 Analog Video

A digital interface stream is based on analog video. The waveform for a line of NTSC analog video is shown in Figure 12-81.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-81. NTSC Analog Video Waveform for One Horizontal Line**



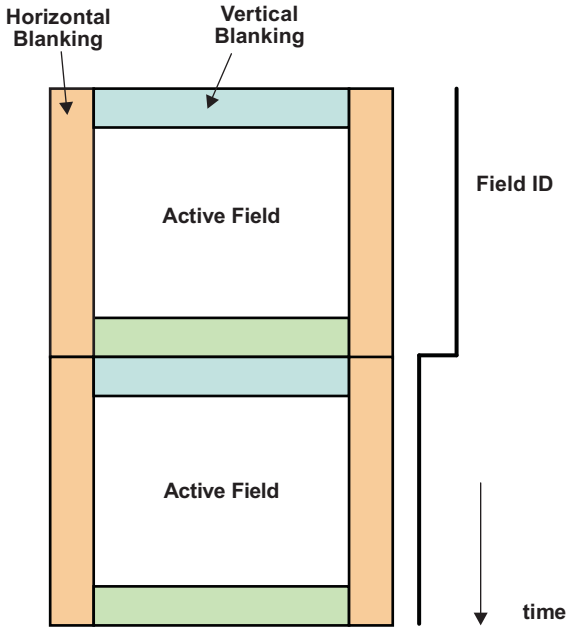

T is a time constant. For NTSC, T=1/13.5 MHz = 74ns.

#### *12.2.8.2.2 Digitized Video*

Digitized video is based on scan lines in found in analog video. BT.601 uses various sync signals to specify when a new field and a new line starts. BT.656 and BT.1120 uses sync words embedded in the data stream to specify start of field and start of line.

An image can be digitized into regions shown in Figure 12-82.

**Figure 12-82. Digitized Video**




With the capability to encode sync words inside the data stream, there is more flexibility for adding non-video related data, called Ancillary Data. Also, code words embedded in the digital stream can be used as a type of identifier for multiplexing several sources of video into one data stream. Such a multi-camera multiplex is useful in the digital security markets.


Copyright © 2011–2016, Texas Instruments Incorporated

Figure 12-83 shows End-of-Active-Video (EAV) and Start-of-Active-Video (SAV) code words added to a video transmission. The period between the EAV and SAV is equivalent to Horizontal Blanking. The period between the SAV to the next EAV is active video or vertical blanking.

In the BT.656 or BT.1120 embedded code word scheme, three bits of the EAV/SAV code word are important: F (field), H (horizontal blanking), and V (vertical blanking).

**Figure 12-83. Code Word Embedded Video Format**

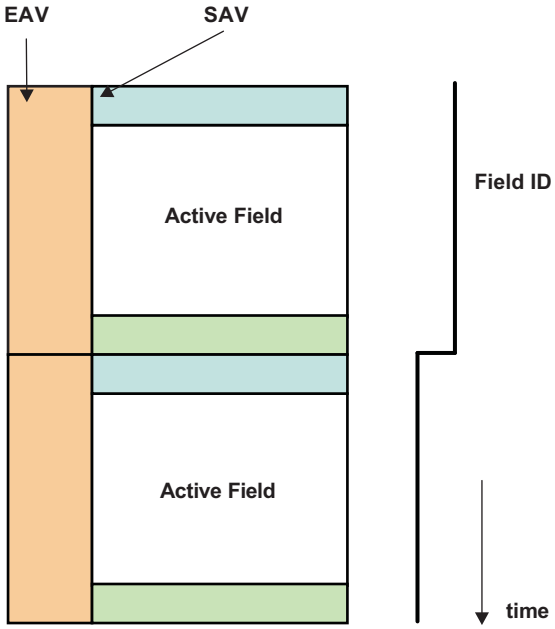
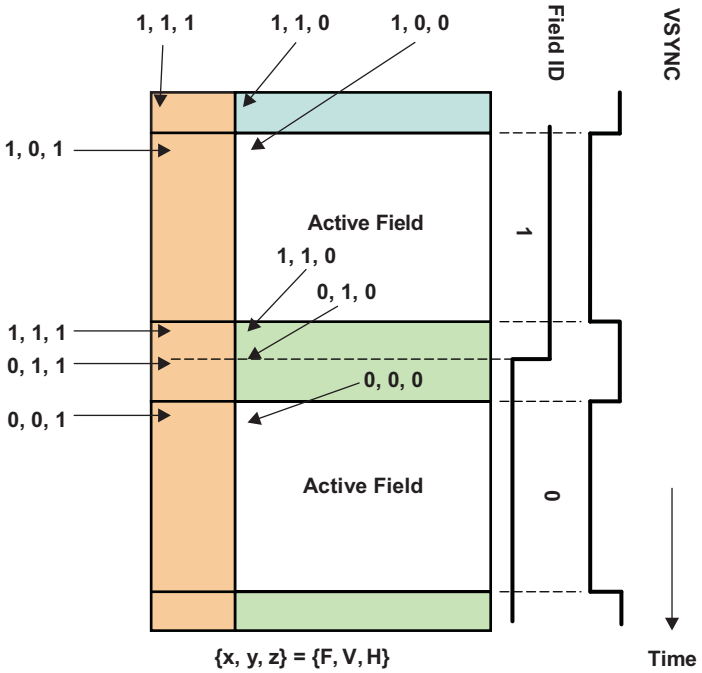



Figure 12-84 shows the values of F, V, and H flags at different locations in the picture. The Field flag represents the state of the Field ID for the picture. For progressive frames, F is always '0.' The V flag specifies vertical blanking areas. The H flag specifies horizontal blanking portions of the picture.

**Figure 12-84. Digitized Video with F, V, and H Flags in EAV/SAV**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.3 Frame Buffers

The VIP/VPDMA supports Frame Buffers in DDR for Active Video and Ancillary Data.

4:2:2 data is always saved in packed pixel buffers.

4:2:0 data is saved in Planar Luma buffers and Planar CbCr pair buffers.

A Luma Frame Buffer is a Planar storage area. Each line is the width in pixels (1Byte/pixel) of the output picture size format. The frame buffer contains the number of active video lines in the output picture size format.

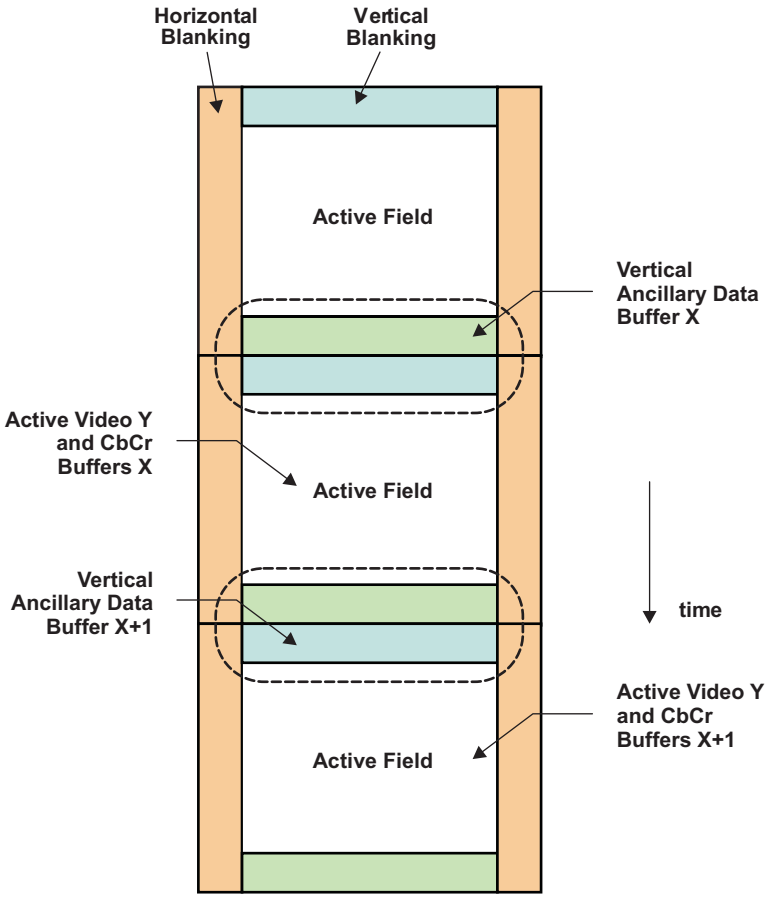
A Chroma Pair Frame Buffer is Planar storage of CbCr pixel pairs, with each pixel being a byte. For 4:2:0 storage, N lines in the output picture active video results in N/2 lines of CbCr pairs being stored.

The Ancillary Data buffer is different than the Active Video Frame Buffers. The Ancillary Data buffer only stores Vertical Blanking Ancillary Data. The number of lines in the Ancillary Data buffer is the same as the number of Vertical Blanking lines. Typically, only one channel is extracted from the Vertical Blanking data, so the width of the Ancillary Data buffer is the same as the width of the Luma Buffer.

In 8-bit input mode, it is possible for both Luma and Chroma sites to be extracted for Vertical Ancillary data. Each color component is strobed on separate input clock cycles. In this case, the line width of the Ancillary data is twice the Luma line width of the picture. Both Luma and Chroma sites cannot be extracted for 16-bit input mode because both Luma and Chroma are sent on the same input clock cycle and the Ancillary port to the VPDMA VPI is only 8 bits wide.

Figure 12-85 shows how the planar data regions are stored in DDR. The vertical blanking data is stored in a set of Planar Buffers. Note that the bottom of the Vertical Ancillary Data from the previous field or frame is stored in the same buffer as the top of the Vertical Ancillary Data from the current field or frame.

**Figure 12-85. Planar Buffer Storage Description**





Copyright © 2011–2016, Texas Instruments Incorporated

The Luma representing Active Video is stored in a set of Planar Buffers. The CbCr Chroma Pairs are stored in a set of Planar Buffers.

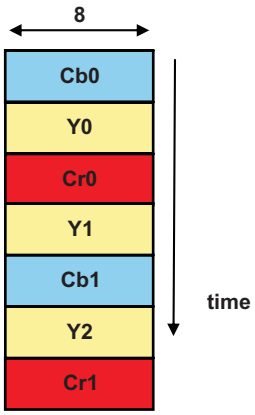
#### 12.2.8.2.4 Input Data Interface

This section describes how the data (luma and chroma data for YUV422 format capture & R,G and B data for RGB888 format capture) is muxed for the interface modes described below.

##### 12.2.8.2.4.1 8b Interface Mode

In 8b data interface mode, the input pixels are multiplexed according to Figure 12-86. The Chroma Format is 4:2:2. Sites with Cb/Cr pixels are known as Chroma sites. Those sites with Y pixels are known as Luma sites.

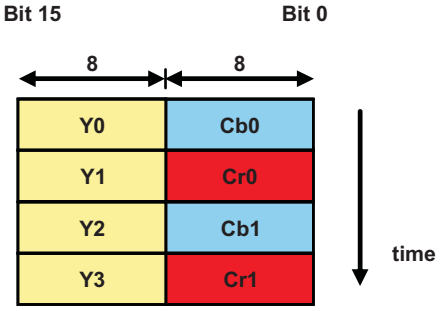
**Figure 12-86. 8b Interface Discrete Sync Pixel Multiplexing**




##### 12.2.8.2.4.2 16b Interface Mode

In 16b interface mode, Luma is on 8 bits of the data bus and Cb/Cr chroma pixels alternate on the other 8 bits of the data bus as shown in Figure 12-87.

**Figure 12-87. 16b Interface Discrete Sync Pixel Multiplexing**





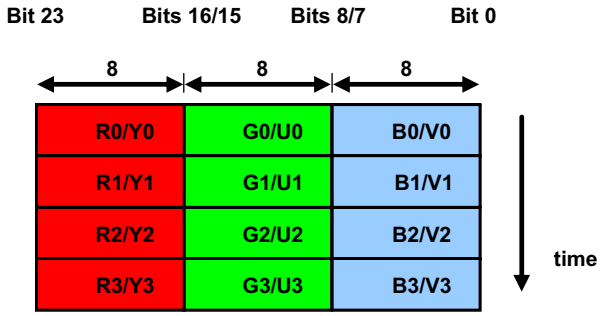
Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.4.3 24b Interface Mode

RGB or YUV444 data is sent in 24b Interface Mode. The three components are packed into the data bus and sent to the VPDMA. The 24-bit Luma VPI client to the VPDMA carries all three components. This data is saved to the DDR in packed mode. That is, the three components are not separated by hardware. The 24-bit data bus is shown in Figure 12-88.

Ancillary data is saved in the Ancillary Data buffer. The ctrl_chan_sel configuration register is used to select whether the ancillary data is from the R/Y, G/U, or B/V channels.

**Figure 12-88. 24b Interface RGB or YUV444 Discrete Sync**

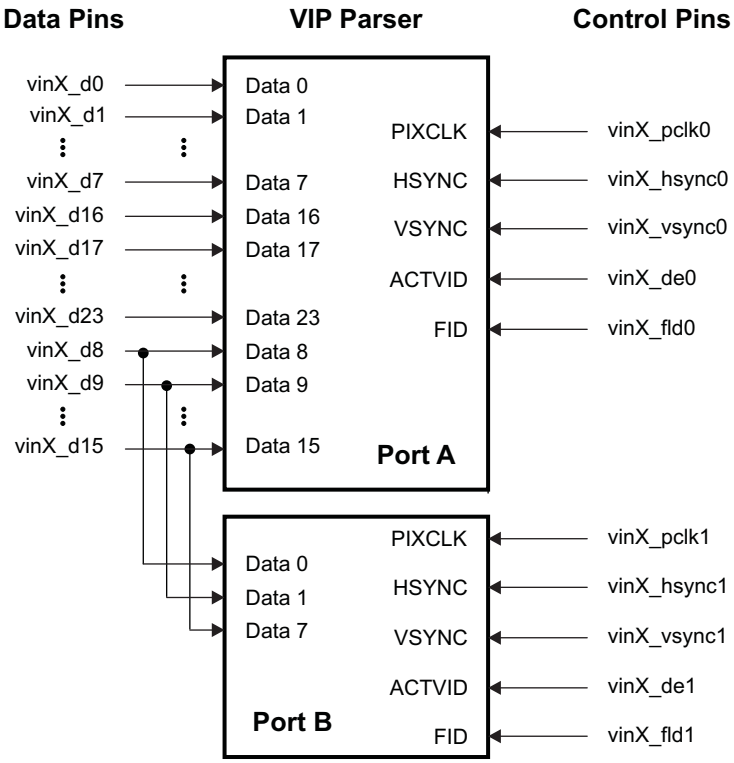



#### 12.2.8.2.5 Input Ports and Sharing of the 24 Bit Data Bus

The VIP Parser supports two independent Pixel Clock Input Domains, which shall be called Port A and Port B. Port A supports a single 24-bit data bus at the instance level and Port B supports a single 8-bit data bus at the instance level.

For the device, a single set of 24 device pins are used to drive both Ports A and B, as shown in the following figure.

**Figure 12-89. 24 Device Pins for Ports A and B**





Copyright © 2011–2016, Texas Instruments Incorporated

The configuration for each device input port is described in Table 12-53.

**Table 12-53. Valid Input Port Configurations**

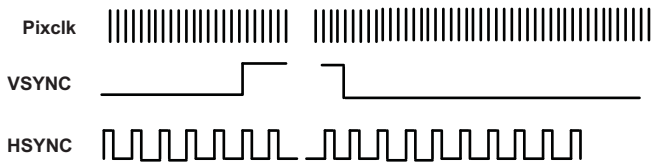
| Port A | Port B |
|---|---|
| 8 Bit | Off |
| 16 Bit | Off |
| 24 Bit | Off |
| 8 Bit | 8 Bit |
| 16 Bit | 8 Bit |
| Off | 8 Bit |

Each Port can individually be configured as Discrete Sync or Embedded Sync.

#### 12.2.8.2.5.1 Signal Relationships

A digital representation of video can be realized by using HSYNC and VSYNC signals to identify frame start and line start. Suppose HSYNC and VSYNC are active high, Figure 12-90 shows the general relationship of these signals.

**Figure 12-90. Discrete Sync Signals**




Every Pixclk cycle carries either an active pixel or a blanking pixel. VSYNC pulses between two fields (or frames, in the case of progressive video). HSYNC pulses to signify the beginning of every line. An ACTVID signal can be used as a data valid to specify active video.

Discrete Sync cannot be used with any multi-camera multiplexed stream inputs. In the device, if Port A is configured for 24 bit discrete sync, then Port B must be disabled since there are no more data input pins left over for Port B.

If Port A is not 24 bits, then the 8-bit Port B can be configured and enabled for either discrete or embedded sync.

Discrete sync basic mode is used to determine the type of vertical blanking signal. DISCRETE_BASIC_MODE='0' means that the vertical blanking interval is specified by the signaling. In the cases where there is no awareness of vertical blanking start and end, vertical blanking data is saved in the Active Video Buffer (DISCRETE_BASIC_MODE='1').

#### 12.2.8.2.5.2 General 5 Pin Interfaces

Discrete Sync signal handling varies among different sending devices. The information that must be conveyed includes the pixel data value, field ID, horizontal blanking, and vertical blanking. Many devices can be configured to adjust the timing of the signals relative to each other.

In this section, DATA will be depicted as 8 bits. However, discrete sync does optionally support 16b and 24b data input. Type 1 is named after a generic five pin interface between the sending and receiving devices.

In Figure 12-91, P0 represents the first pixel in the horizontal blanking interval following the last vertical blanking line of the previous field or frame. HSYNC specifies the horizontal blanking region and VSYNC specifies that the P0 pixel is in the vertical sync area. HSYNC can be a strobe that is active one or more cycles and can deassert before the actual end of horizontal blanking or HSYNC may be active for the full duration of horizontal blanking.

Likewise, VSYNC can be a strobe that is active one or more cycles and can deassert before the actual end of vertical blanking or VSYNC may be active for the full duration of vertical blanking.


Copyright © 2011–2016, Texas Instruments Incorporated

FID can change at this pixel or it may change later. For interlaced source, though, the FID will be inverted for this pixel at the same time point in the next field. So, it does not really matter when FID is captured. Many sending devices allow the location of FID changes to be programmable.

In this diagram and all others in this document, the active polarities of the interface signals can be either high or low. For the sake of uniformity in this document, all polarities are drawn active high. Also, different vendors have different datasheet names for the interface signals.

**Figure 12-91. Type 1, First Horizontal Blanking Pixel**

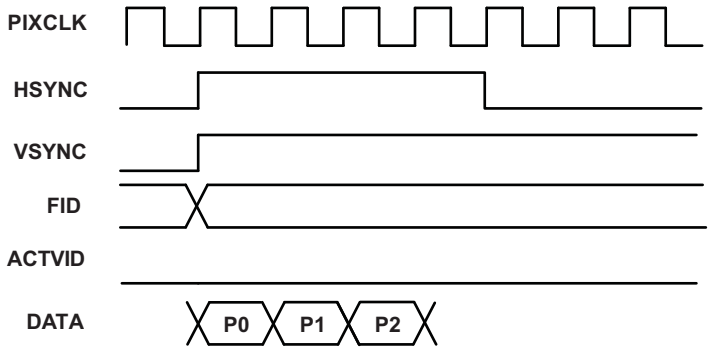
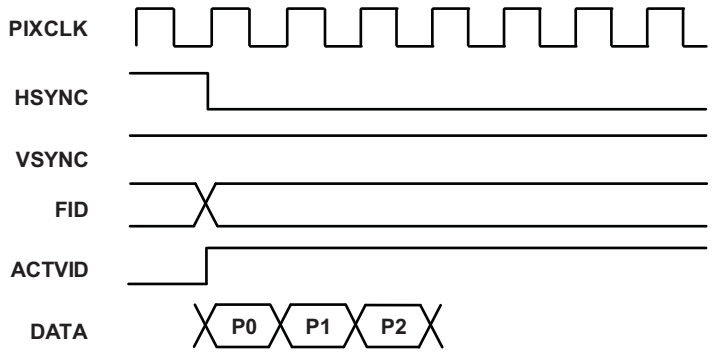



Figure 12-92 shows the P0 pixel being the first Chroma Channel data value in the Vertical Ancillary Data region. HSYNC is definitely de-asserted by now since P0 is no longer in horizontal blanking. ACTVID may or may not be active for Vertical Ancillary Data. Some devices consider these pixels to be Active (as in non-horizontal blanking). Other devices consider only video to be ACTIVE Video.

**Figure 12-92. Type 1, First Vertical Ancillary Data Pixel**

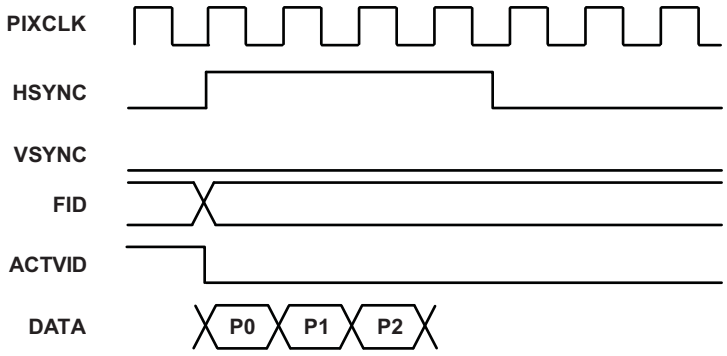




Copyright © 2011–2016, Texas Instruments Incorporated

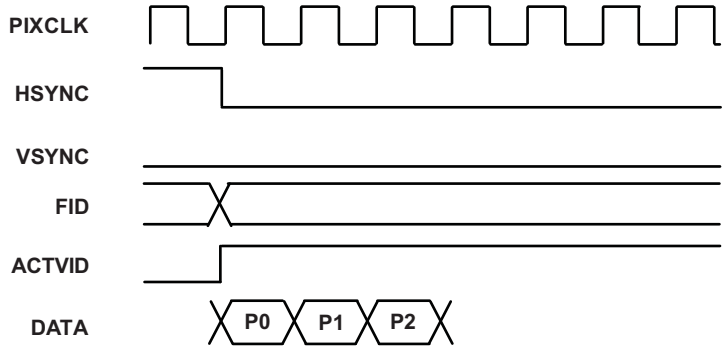
Following the vertical blanking region, the video portion of the field or frame starts. Figure 12-93 shows the horizontal blanking area in this video portion of the field or frame. P0 is the first pixel in the horizontal blanking. HSYNC is active for one or more pixel clocks. VSYNC is inactive in this video area. FID can change here. ACTVID is low since P0 is horizontal blanking.

**Figure 12-93. Type 1, Horizontal Blanking in Video Region**




In Figure 12-94, P0 represents the first Chroma pixel in the Active video line. HSYNC is inactive, since P0 is in the active video region. Likewise, VSYNC is inactive. FID may or may not change here. ACTVID is high to signal capturing of video pixels.

**Figure 12-94. Type 1, First Video Pixel**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.5.3 Signal Subsets—4 Pin VSYNC, ACTVID, and FID

A sending device may use only a subset of the signals described in Section 12.2.8.2.5.1. The sending device just needs to convey important signals required to capture the field or frame. It can be shown that various selections of four pins can be used to satisfy all Type 1 conditions.

Three pins, VSYNC, ACTVID, and FID, plus a pixel clock can be used to support discrete sync. VSYNC would bump the capture buffer. An inactive to active level of ACTVID specifies a line of data to capture. FID determines the field ID polarity. The scenario in which the sending device wants the receiving end to capture Vertical Ancillary Data using 4-pin signaling is shown in Figure 12-95.

**Figure 12-95. 4-Pin Reduced ACTVID Signaling with Vertical Ancillary Data**

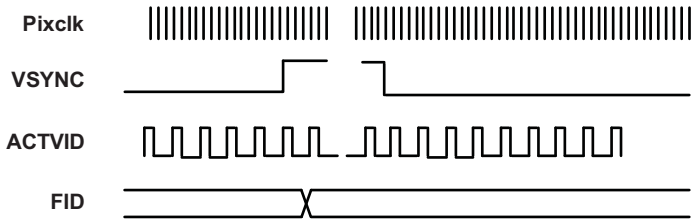
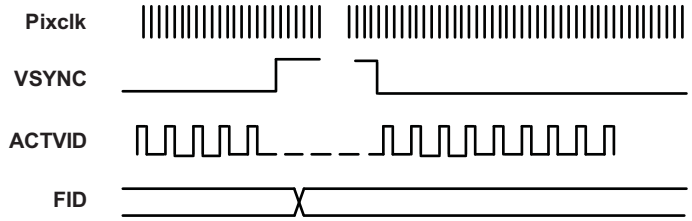



Figure 12-96 describes the case using the 4-pin interface in which the sending device does not send Vertical Ancillary Data.

**Figure 12-96. 4-Pin Reduced ACTVID Signaling with No Vertical Ancillary Data**

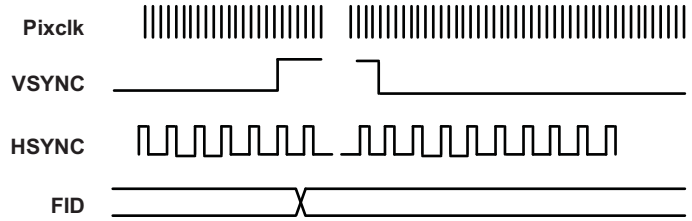



#### 12.2.8.2.5.4 Signal Subsets—4 Pin VSYNC, HSYNC, and FID

In this style of Discrete Sync, as shown in Figure 12-97, four pins are used including the Pixel Clock. HSYNC signals the beginning of the line. All data in the line is captured, including Horizontal Blanking Data. In fact, this signaling mode is the only one that allows Horizontal Blanking Data to be captured.

Of course, by capturing the horizontal blanking pixels in the frame buffers, there is no way to be certain exactly where the blanking ends and the active video starts. One would have to rely solely on video format specs to find the active video inside the frame buffer.

**Figure 12-97. 4-Pin Reduced HSYNC Signaling with Vertical Ancillary Data**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.5.5 Vertical Sync

Vertical Sync is used to indicate lines that are in the vertical blanking interval. The VSYNC also separates fields or frames. To be spec compliant, VSYNC should be active for a few lines at the bottom of the picture. The exact number of vertical blanking lines at the bottom depends on the specification for the picture format (480i, 480p, 720p, 1080i, 1080p).

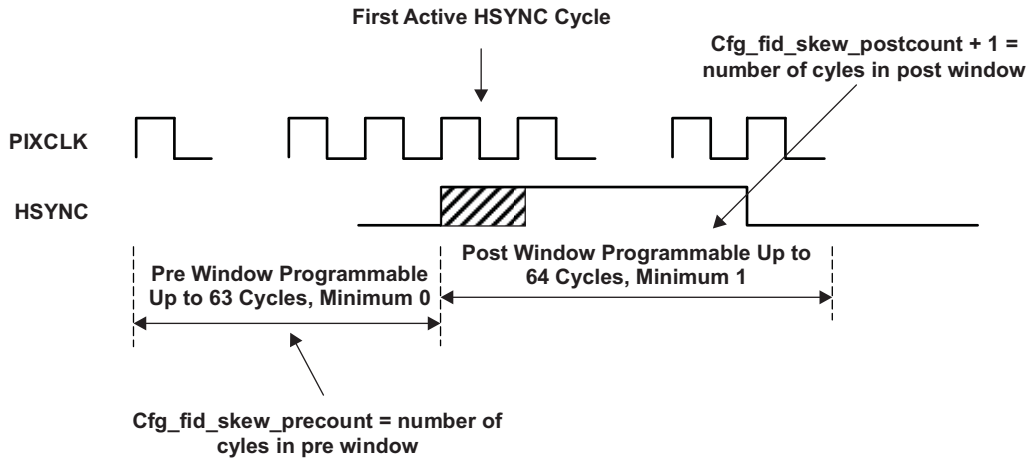
Likewise, the number of vertical blanking lines at the top of the picture depends on the video format specification corresponding to the incoming picture.

In the VIP parser, lines associated with an Active VSYNC are stored in the vertical ancillary data buffers using the ANC VPI port to the VPDMA. Lines without an Active VSYNC are stored in the active video Luma and Chroma-pair buffers using the Y and UV VPI ports, respectively, to the VPDMA.

When using HSYNC signaling instead of ACTVID, the VSYNC signal may be derived from an analog source such as an NTSC/PAL decoder. In this case, VSYNC may not transition on the exact cycle as HSYNC. Thus, the VIP Parser supports a window region around HSYNC in which VSYNC transitions will be detected. A VSYNC transition occurring within the window is identified the same way as if VSYNC transitioned on the same cycle as HSYNC going active.

The window is defined by a pre-window, which is determined by the cfg_fid_skew_precount register. There is also a post-window that is defined by cfg_fid_skew_postcount. Note that although the configuration registers are named fid_skew, they are also used for defining the VSYNC transition window. The window region definition is shown in Figure 12-98.

**Figure 12-98. VSYNC Pre and Post Window**




The results of VSYNC behavior in the transition window are shown in Figure 12-99. A low to high transition in the window is equivalent to VSYNC going low to high on the active HSYNC cycle.

Likewise, a high to low transition in the window is equivalent to VSYNC going high to low on the active HSYNC cycle.

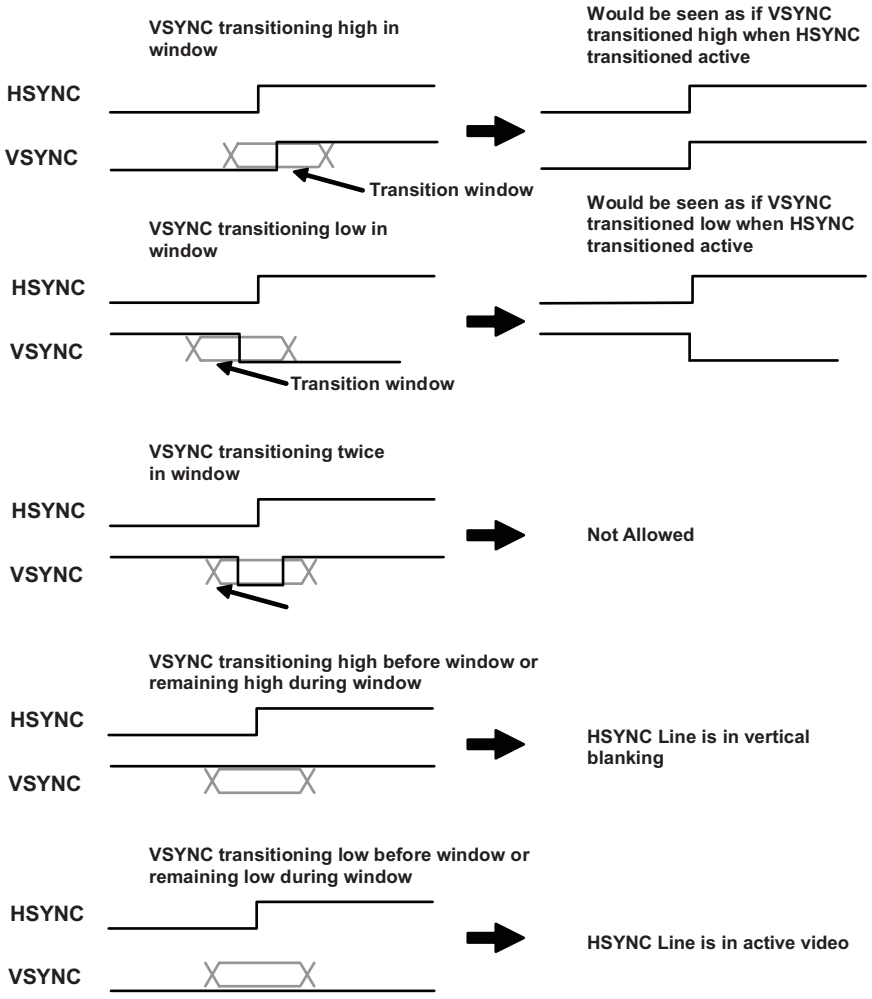
Two transitions of VSYNC within the VSYNC window is not allow and is undefined behavior.

If VSYNC is high throughout the transition window, then the HSYNC line is in vertical blanking.

If VSYNC is low throughout the transition window, then the HSYNC line is in active video.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-99. VSYNC Equivalence When Using Transition Window**




Note that VSYNC skew generally only applies to input signals that have been sampled from an analog source, as in a NTSC/PAL decoder. If the VSYNC is a VBLANK-type signal or if the sending device is another all-digital IC such as another device or Davinci part, then the VSYNC signal does not have a skew since VSYNC will be aligned to HSYNC. In this case, setting cfg_fid_skew_precount = ‘0’ and cfg_fid_skew_postcount = ‘0’ defines a minimum size window which will capture the value of VSYNC on the same cycle that HSYNC goes active.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.5.6 Field ID Determination Using Dedicated Signal

For Progressive Source, FIELD ID is always '0.'

For Interlace Source, we need to extract FIELD ID consistently.

In some cases, vertical sync is active on the first pixel of a line in the vertical blanking period and it stays active until the last line in the vertical blanking period.

However, the pixel where the FIELD ID signal transitions can be quite variable and depends on the external chip driving the VIP Parser. Many parts that generate digitized raw video have a programmable feature to specify when FIELD ID changes. From our standpoint, the good news is that FIELD ID is valid at the same point for every field. That is, if FIELD ID is read at one particular place in a field, the polarity of the signal will be reversed at the same location in the next field. So, FIELD ID can be corrected with a programmable polarity configuration bit that is XORed with the captured value.

For discrete sync mode, FIELD ID will be registered on the first active pixel capture cycle of each line in both styles of HSYNC and ACTVID usage as specified in Figure 12-100 and Figure 12-101.

**Figure 12-100. FID Registering When Using HSYNC**

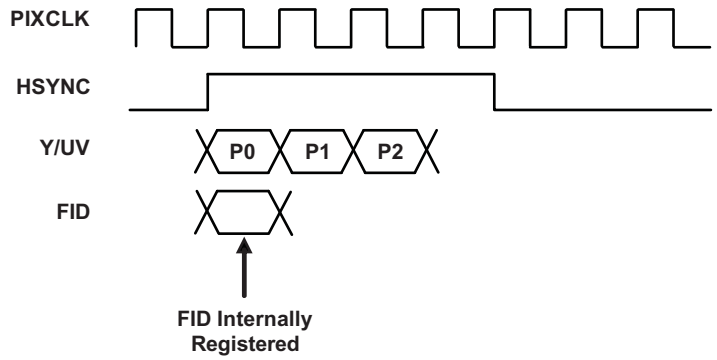
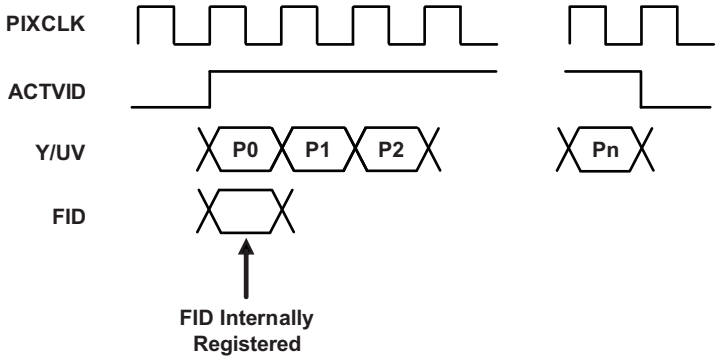



**Figure 12-101. FID Registering When Using ACTVID**





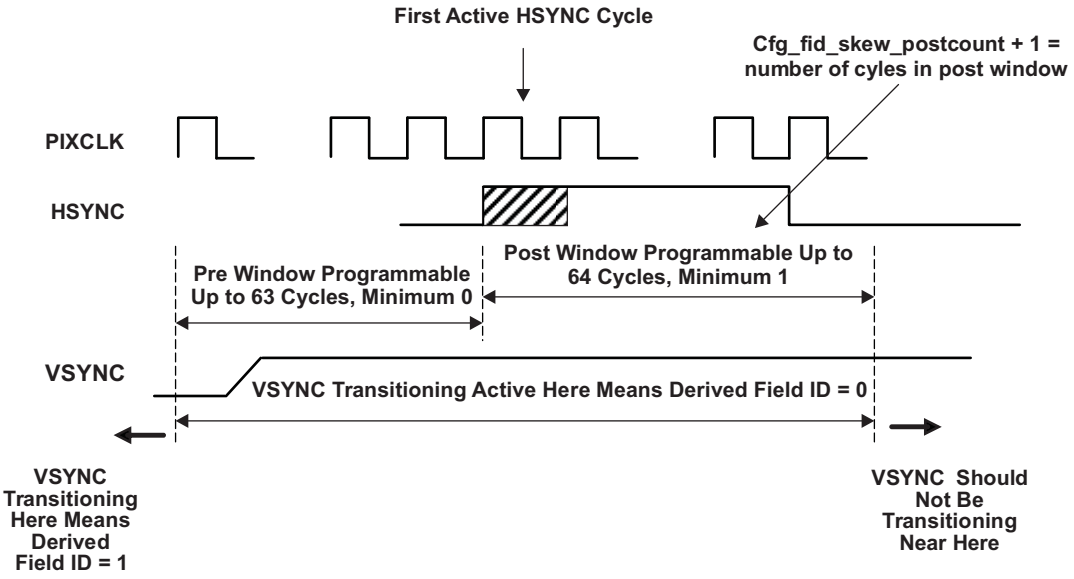
Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.5.7 Field ID Determination Using VSYNC Skew

In order to save a device pin, there is a case where a skew may be inserted into VSYNC (with respect to HSYNC) when HSYNC is used as a start of line indicator as described in Figure 12-97. In this case, no FIELD ID signal is sent by the source chip. A description of Field ID determination by VSYNC skew is shown in Figure 12-102.

The active polarity of VSYNC falling within n pixel clock cycles of the first active cycle of HSYNC indicates the field id. If VSYNC is active before this time window, then the field_id = '1' for the next picture. If VSYNC becomes active within this window, then field_id = '0' for the next picture.

**Figure 12-102. Field ID Determination By VSYNC Skew**




When using FID determination by VSYNC skew, the value for VSYNC is also determined by transitions in the window as shown in Figure 12-99.

The VIP Parser supports a configuration FID polarity bit. For FID determination by VSYNC Skew, the fid determination functions as described in Table 12-54.

**Table 12-54. Polarity Table for FID Determination By VSYNC Skew**

| Cfg_fid_polarity | Transition in Pre/Post Range | FID Determination |
|---|---|---|
| 0 | No | 1 |
| 0 | Yes | 0 |
| 1 | No | 0 |
| 1 | Yes | 1 |

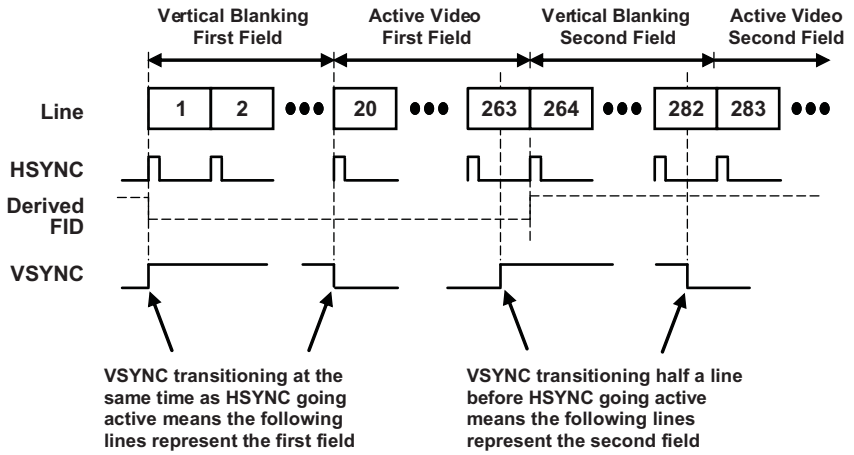

Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.5.8 Rationale for FID Determination By VSYNC Skew

FID determination by VSYNC skew is a method for field ID determination derived from the analog NTSC and PAL interlaced specifications. Under this method, the sending device will not be providing a FID signal. NTSC has 525 total lines split between two fields. PAL has 625 total lines split between two fields. Each of these interlaced standards support an odd number of lines.

Let's consider just the 525 active line NTSC signal. For the sake of consistency, let's call Line 1 the first line of the 2-field pair and Line 525 the last line of the 2-field pair.

**Figure 12-103. Example of 525-line FID Determination By VSYNC Skew**




A waveform is shown in Figure 12-103. VSYNC is defined to go active at the same time as HSYNC for the first line of the first field in a two-field picture pair. For this first field, VSYNC will go inactive after Line 20.

For the second field, VSYNC will go active in the middle of Line 263 to signal that Line 264 is the start of a vertical blanking interval. When HSYNC for Line 264 arrives, coinciding with the vertical blanking interval for the beginning of the second field, VSYNC has already been active for half of Line 263. For the second field, VSYNC will go inactive midway through Line 282 to indicate that Line 283 is active video. When HSYNC for Line 283 appears, VSYNC has already been inactive for half a line.

By seeing whether VSYNC transitions at the beginning of a line or whether it transitions at the midway point of a line, one can determine whether the upcoming group of lines represents the first field or the second field. The derived FID is shown in dashed lines.

The analog NTSC specification defines the field ID changing part way into the vertical blanking. That is, the first few lines of vertical blanking belong to the previous field and the next several lines of vertical blanking belong to the upcoming field. The VIP Parser saves one channel of the entire vertical blanking interval between two active video fields into a single buffer. The hardware does not discriminate between whether the vertical blanking lines belong to the bottom of the previous field or those belonging to the start of the next field. This usage model is consistent with vertical ancillary data capture for embedded sync mode of operation.

Obviously, FID Determination by VSYNC skew cannot be used when framing does not use the VSYNC signal but rather relies on the ACTVID signal instead.

#### 12.2.8.2.5.9 ACTVID Framing

Instead of an HSYNC signal, the VIP Parser can use ACTVID framing as described in Figure 12-96. Under ACTVID Framing, VSYNC is used to separate vertical blanking lines from active video lines.

FID determination by VSYNC Skew is not allowed for ACTVID framing because there is no HSYNC input signal in this mode. Also, the VSYNC transition window is not employed. VSYNC is captured at the first pixel of each ACTVID grouping of pixels. Lines are separated by ACTVID transitioning inactive.

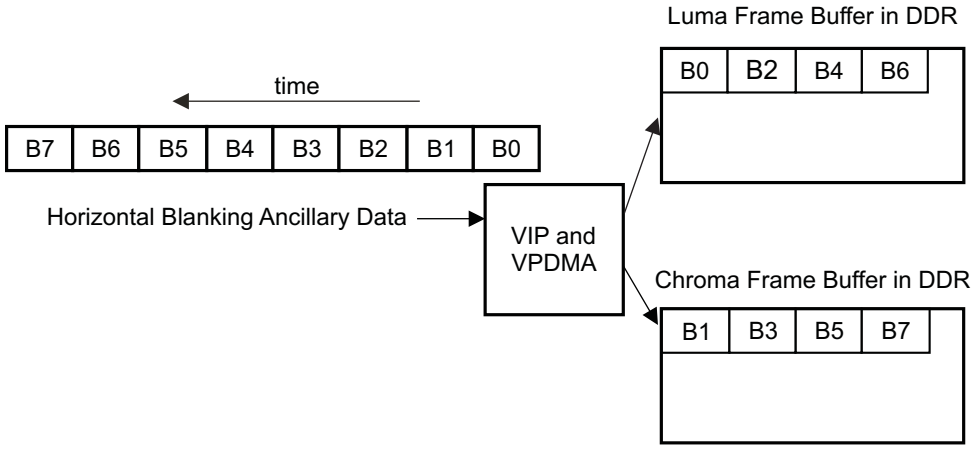

Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.5.10 Ancillary Data Storage in Descrete Sync Mode

Ancillary data appearing in horizontal blanking is called Horizontal Blanking Ancillary Data. Ancillary Data in vertical blanking is called Vertical Blanking Ancillary Data.

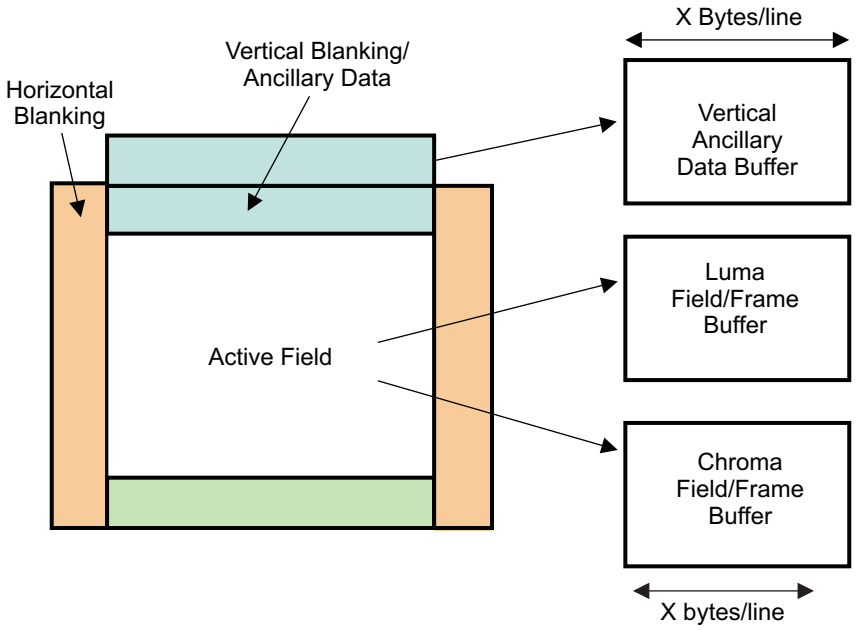
Horizontal Blanking Ancillary Data is not commonly used. For the ACTVID data valid mode described in Figure 12-95, there is no way to capture Horizontal Blanking Ancillary Data. Using the HSYNC mode in Figure 12-97, all blanking pixels are captured. However, the horizontal ancillary data is byte-by-byte distributed between the Luma and Chroma frame buffers. Chroma sited bytes are saved in the Chroma frame buffer and Luma sited bytes are saved in the Luma frame buffer. Some CPU effort would be needed in order to extract ancillary data from the desired Luma or Chroma channel frame buffers. Also, the video on each line starts after the horizontal blanking period. This situation is shown in Figure 12-104.

**Figure 12-104. Horizontal Ancillary Data Packing When HSYNC Used as Sync Signal**




With Interlaced source material, Vertical Blanking Ancillary data will be stored in a separate Vertical Ancillary Data Buffer as shown in Figure 12-105. The Channel from which vertical ancillary data is extracted is a configuration option. For an input image of x active pixels per line, each line of Vertical Blanking Ancillary Data will have x bytes. Unlike the horizontal case, the CPU parsing this Ancillary Data will see a contiguous section of Vertical Ancillary Data that is not intermixed with Video data.

**Figure 12-105. Interlaced Field Vertical Blanking Ancillary Data Storage**

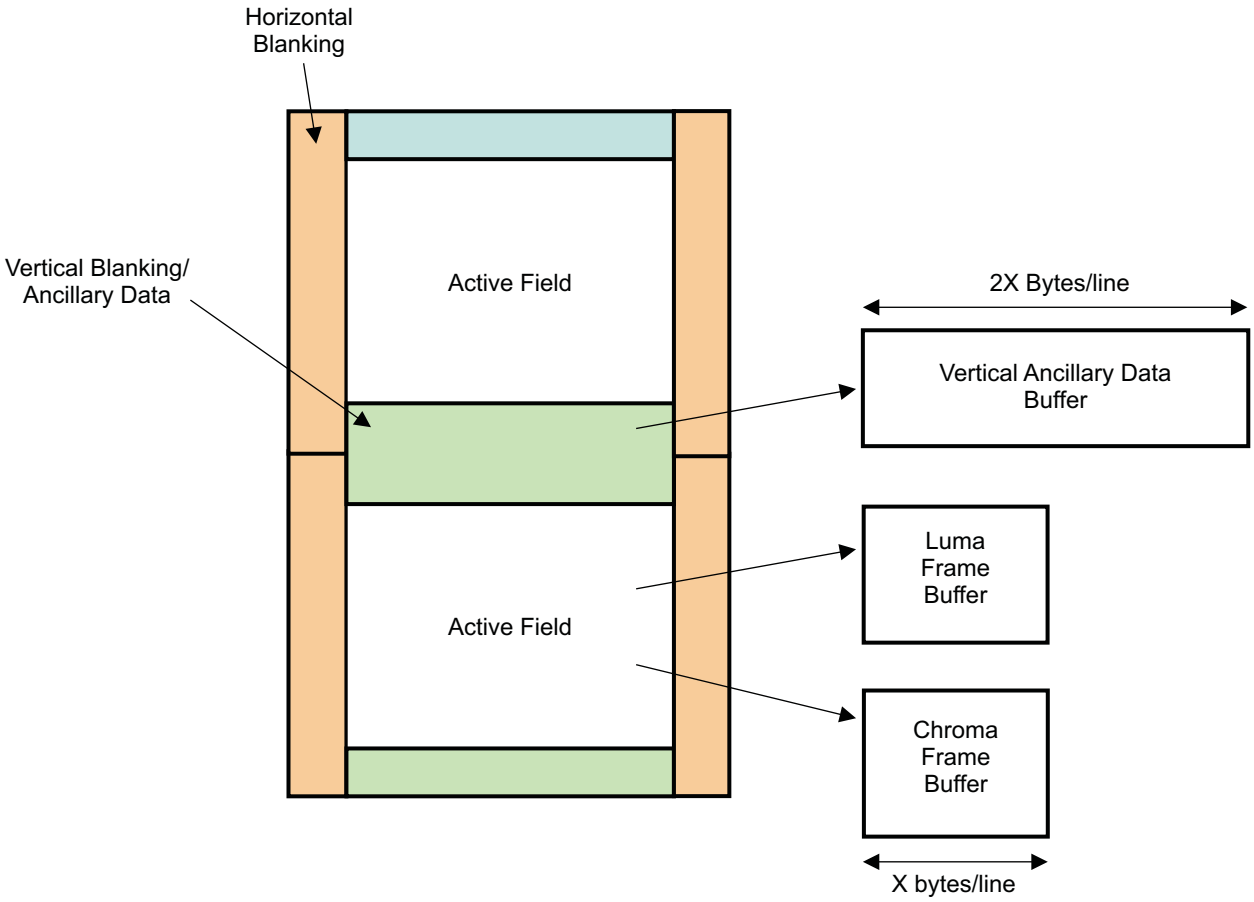




Copyright © 2011–2016, Texas Instruments Incorporated

For Progressive source video, the FIELD ID does not change. So, the Vertical Ancillary Data Buffer will contain all the information beginning from the vertical blanking of the previous frame. This situation is shown in Figure 12-106.

**Figure 12-106. Progressive Frame Vertical Blanking Ancillary Data Storage**

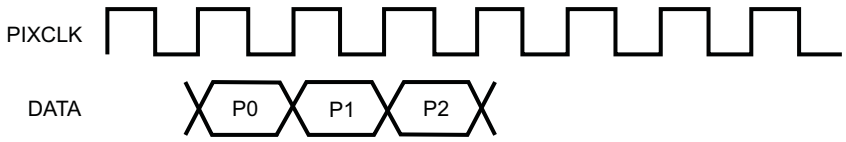



### 12.2.8.2.6 BT.656 Style Embedded Sync

#### 12.2.8.2.6.1 Data Input

Like Discrete Sync Input, Embedded Sync mode takes data from the 24b input bus. Input data can be 8, 16, or 24 bits wide. A sample is retrieved each and every Pixel Clock cycle. There is no valid signal gating data entry. Figure 12-107 shows a valid data sample each Pixel Clock period.

**Figure 12-107. Embedded Sync Data Entry**




#### 12.2.8.2.6.2 Sync Words

In embedded sync mode, code words are inserted into the stream at pixel clock rates. For external devices that send out 10 bits (single pixel interface) or 20 bits (parallel Y-Cb/Cr interface) of data, only the 8 (single pixel interface) or 16 (parallel 8bY-8bCb/Cr interface) most significant bits of each pixel are used.


Copyright © 2011–2016, Texas Instruments Incorporated

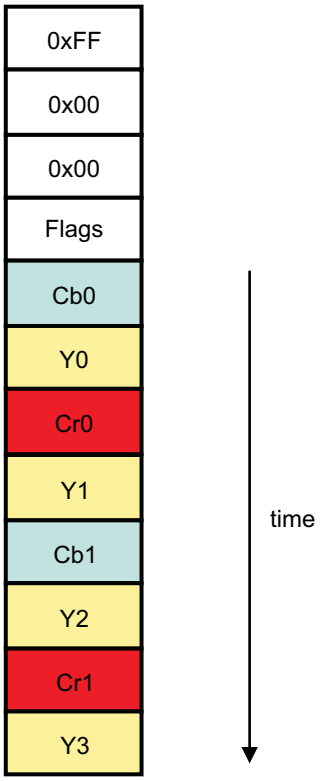
The key code words are Start of Active Video (SAV) and End of Active Video (EAV). Three flags are found in these code words: F (field), V (vertical sync), and H (horizontal sync). These flags signify the position in the frame corresponding to the data immediately following the codeword. The flags determine whether the code is EAV or SAV and where they lie in the picture. The first byte of the code word is 0xFF. The second and third bytes are 0x00. The bit ordering of the fourth byte is detailed in Table 12-55.

**Table 12-55. Fourth Byte of EAV/SAV Code Word**

| 7 (fixed) | 6 (F) | 5 (V) | 4 (H) | 3 (P3=V^H) | 2 (P2=F^H) | 1 (P1=F^V) | 0 (P0=F^V^H) | Description |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SAV, Field 0, Active Video |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | EAV. Field 0, Horizontal Blanking |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | SAV, Field 0, Vertical Blanking |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | EAV, Field 0, Horizontal Blanking in Vertical Blanking Region |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | SAV, Field 1, Active Video |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | EAV, Field 1, Horizontal Blanking |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | SAV, Field 1, Vertical Blanking |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | EAV, Field 1, Horizontal Blanking in Vertical Blanking Region |

An example of the input ordering of the embedded code word, followed by active video, is shown in Figure 12-108. The input data mode is 8 bits for the example.

**Figure 12-108. Code Word Format Example Followed by Video Data**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.6.3 Error Correction

The FVH flags are sent with four protection bits to support double error detection, single error correction. A non-correctable detected error is simply ignored. An option exists for the protection bits to correct a single bit error in the FVH flags. The correction table is shown in Table 12-56. n/c means that the error condition is detected, but it is non-correctable.

**Table 12-56. Error Correction Matrix**

| P3, P2, P1, P0 | F, V, and H Flags | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 0000 | 000 | 000 | 000 | n/c | 000 | n/c | n/c | 111 |
| 0001 | 000 | n/c | n/c | 111 | n/c | 111 | 111 | 111 |
| 0010 | 000 | n/c | n/c | 011 | n/c | 101 | n/c | n/c |
| 0011 | n/c | n/c | 010 | n/c | 100 | n/c | n/c | 111 |
| 0100 | 000 | n/c | n/c | 011 | n/c | n/c | 110 | n/c |
| 0101 | n/c | 001 | n/c | n/c | 100 | n/c | n/c | 111 |
| 0110 | n/c | 011 | 011 | 011 | 100 | n/c | n/c | 011 |
| 0111 | 100 | n/c | n/c | 011 | 100 | 100 | 100 | n/c |
| 1000 | 000 | n/c | n/c | n/c | n/c | 101 | 110 | n/c |
| 1001 | n/c | 001 | 010 | n/c | n/c | n/c | n/c | 111 |
| 1010 | n/c | 101 | 010 | n/c | 101 | 101 | n/c | 101 |
| 1011 | 010 | n/c | 010 | 010 | n/c | 101 | 010 | n/c |
| 1100 | n/c | 001 | 110 | n/c | 110 | n/c | 110 | 110 |
| 1101 | 001 | 001 | n/c | 001 | n/c | 001 | 110 | n/c |
| 1110 | n/c | n/c | n/c | 011 | n/c | 101 | 110 | n/c |
| 1111 | n/c | 001 | 010 | n/c | 100 | n/c | n/c | n/c |


Copyright © 2011–2016, Texas Instruments Incorporated

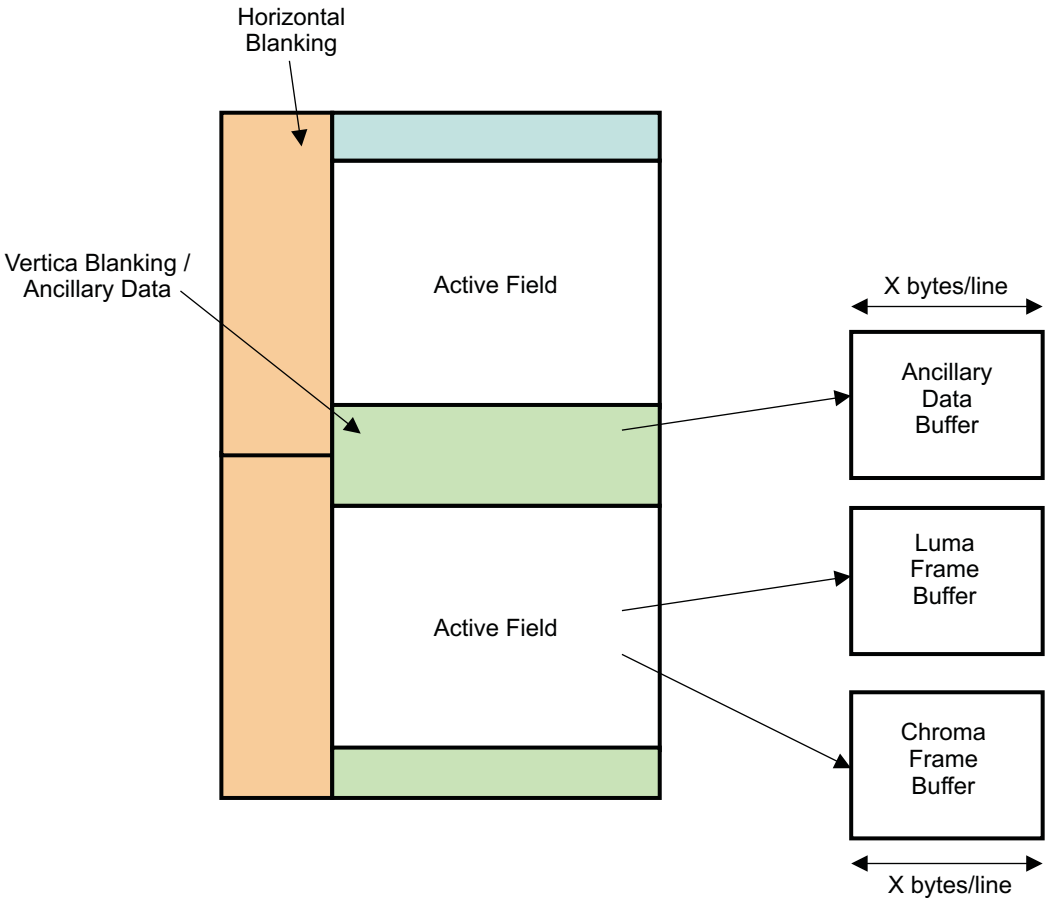
#### 12.2.8.2.6.4 Embedded Sync Ancillary Data

With Embedded Sync streams, only Vertical Ancillary Data can be extracted. The Vertical Ancillary Data buffer is the same width as the corresponding Luma and Chroma buffers. The channel from which Vertical Ancillary Data is extracted is a configuration option.

Horizontal Ancillary data cannot be extracted using embedded sync mode.

The Vertical Ancillary Data is captured starting from the end of the previous active video. See Figure 12-109 for a more detailed description of embedded sync packing.

**Figure 12-109. Embedded Sync Packing**





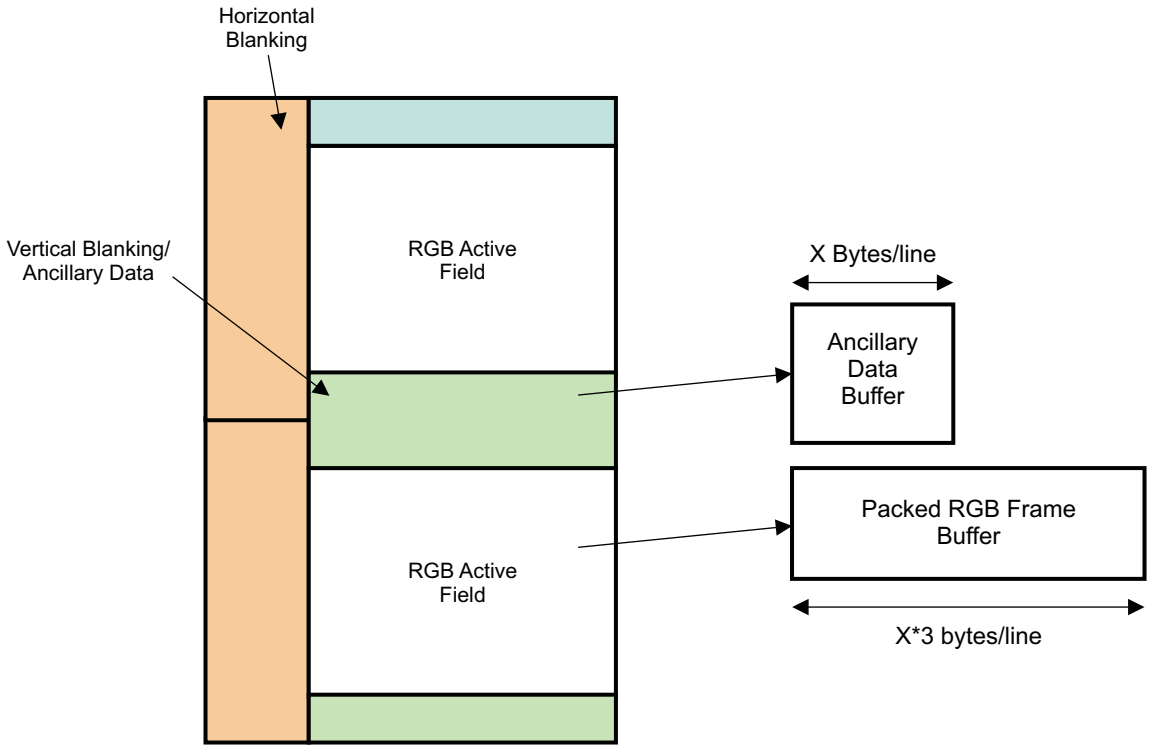
Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.6.5 Embedded Sync RGB 24-bit Data

YUV streams are separated into a planar Luma buffer, a planar Chroma Pair (CbCr) buffer, and a planar Vertical Ancillary Data buffer. RGB streams, on the other hand, are stored in a packed R-G-B format, as shown in Figure 12-110.

The BT.1120 standard defines a method of carrying RGB streams. After the SAV code, 8 bits of R, 8 bits of G, and 8 bits of B data are clocked in one cycle. A 24 bit data bus is required. The channel in which to search for the embedded sync codes and the FVH control code is determined by a configuration selection. Only vertical ancillary data from one channel (R, G, or B), which happens to be the channel where control codes are found, are captured. Vertical ancillary data in the other two channels are ignored.

**Figure 12-110. RGB Frame Storage**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.7 Source Multiplexing

##### 12.2.8.2.7.1 Multiplexing Scenarios

Some applications require multiple camera sources to be used at the same time. For example, video surveillance systems can record and display multiple camera sources. For this type of device, one solution would be to support N 8b or 16b data interfaces for each of N cameras. However, this solution does not efficiently minimize pin count. One set of 8b or 16b interfaces has the bandwidth to support more than one video source, depending on the resolution of the video. Table 12-57 is explanatory only and shows the number of sources that can be multiplexed in one VIP for 8b and 16b interface modes. Note that it does not reflect the capabilities of the VIP Parser design. In addition, the interface pixel clock rates are shown. The VPDMA limits 16 camera sources to be saved to DDR per Pixel Clock Input Domain.

**Table 12-57. Multiplexing Configurations and Pixel Clock Rates**

| | Max Channels In Single 16b Data Interface Mode | Max Channels In Dual 8b Data Interface Mode - Interleaved Channels per Single 8b Port. One 16b VIP can be configured to support two such 8b ports. | Interface Clock Rate (MHz) |
|---|---|---|---|
| HD Interlaced | 2 | 1 | 148.5 |
| D1 Interlaced | 8 | 4 | 108.1 |
| CIF Interlaced | n/a | n/a | n/a |
| HD Progressive | 1 | n/a | 148.5 |
| D1 Progressive | 4 | 2 | 108.1 |
| CIF Progressive (1) | 32 | 16 | 162.2 |

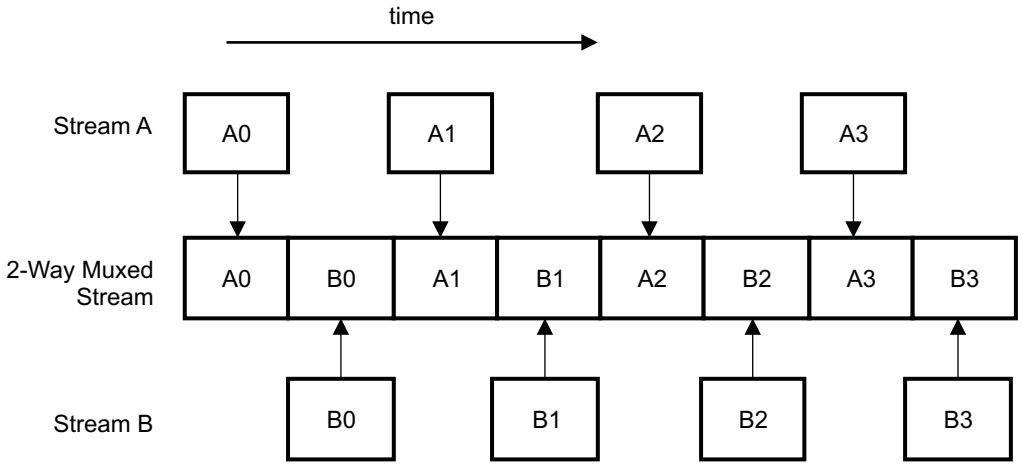
(1) Blanking pixels are not used in the CIF clock rate calculations. Addition of blanking pixels would require a slightly higher clock rate.

**NOTE:** These Channel Density values reflect 1 VIP subsystem.

##### 12.2.8.2.7.2 2-Way Multiplexing

For 2-Way Multiplexing, two embedded sync streams are interleaved a pixel at a time as shown in Figure 12-111.

**Figure 12-111. 2-Way Multiplexing**




The sync codeword, FF-00-00-XY, is replicated in both source streams. In 2-Way Multiplexing, the sizes of both camera sources must be the same. Likewise, the Vertical Ancillary Data size for both sources must be identical. However, the two streams are not necessarily sending the same pixel site in adjacent clock cycles.

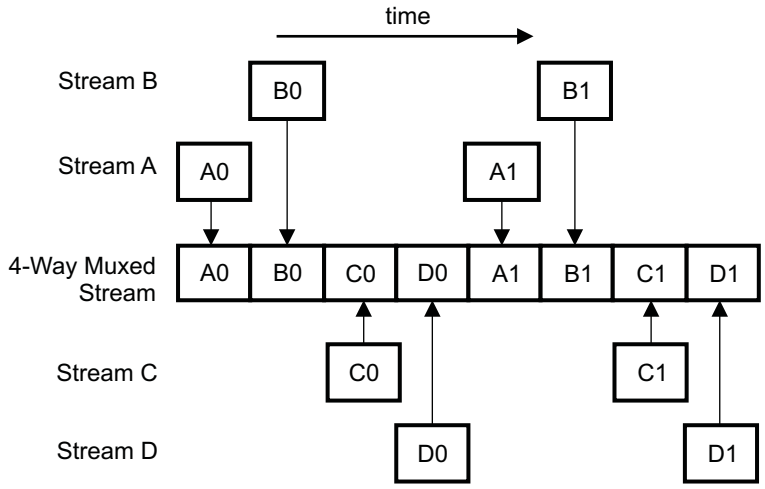

Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.7.3 4-Way Multiplexing

For 4-Way Multiplexing, four embedded sync streams are multiplexed into one as seen in Figure 12-112.

Again, the sync codeword is in all four sources. Like 2-Way Multiplexing, the sizes of the four camera sources are the same and the sizes of the Vertical Ancillary Data regions are the same. The four streams are not necessarily sending the same pixel site in adjacent clock cycles.

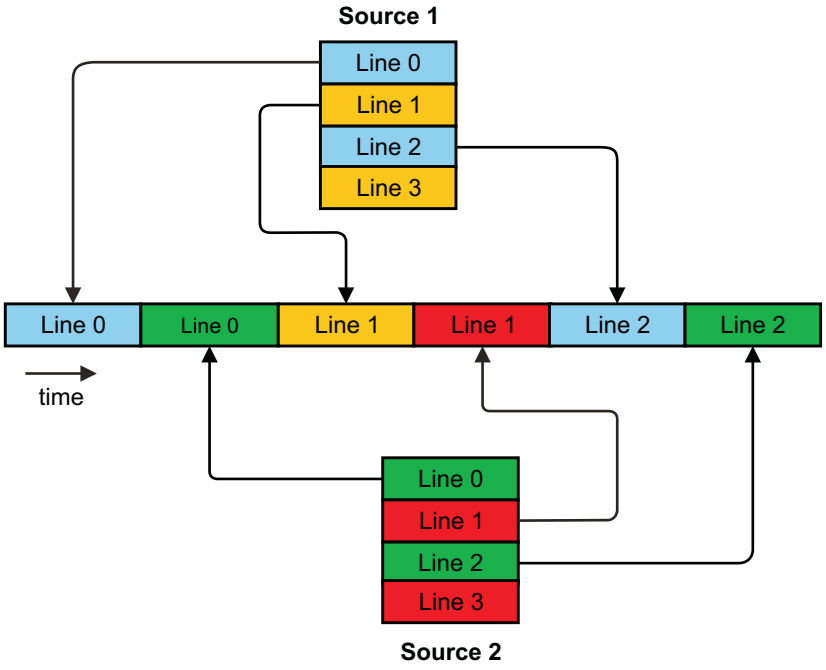
**Figure 12-112. Example of 4-Way Multiplexing**




#### 12.2.8.2.7.4 Line Multiplexing

In Line Multiplexing, n-different sources are sent into the VIP a complete line at a time using a modified version of embedded sync. An example of Line Multiplexing for two sources is shown in Figure 12-113.

**Figure 12-113. Example of Line Multiplexing**




The width and height of each source in Line Multiplexed data can be different. For instance, one source can be PAL while another one can be NTSC. A line is comprised of YUV422 pixels in repeating patterns of CbYCrY.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.7.5 Super Frame Concept in Line Multiplexing

Older TI DSPs do not natively support Line Multiplexing, which is thus far proprietary to the TVP5158 video decoder. When connected to older DSPs, the CPU must parse through the pixel data and in order to create frame buffers. Hence, the '5158 team came up with the concept of a super frame. Different camera sources are interleaved on a line-by-line basis. The beginning of each line carries a Metadata tag that provides key information about that line.

This Metadata tag is preceded by a SAV codeword in which F=0, V=0, and H=0. In other words, the older DSPs would see this line as a normal active video line.

At the end of the line, an EAV code is inserted with an appropriate number of padding pixels following it. This EAV code has F=0, V=0, and H=1.

In older DSPs, a set of like sized buffers are defined to store raw video frames. When a line's V bit changes, the current buffer is closed and an interrupt is given to the CPU. An empty buffer is used to store the incoming line's pixels. With the 5158's SAVs and EAVs having the V flag always set to 0, the older DSPs would never advance their buffer pointer.

Hence, there is a dummy vertical blanking line. The external device would insert an artificial vertical blanking line with a SAV having F=0, V=1, and H=0. It also inserts a dummy EAV with F=0, V=1, and H=1. This dummy line causes the older DSPs to change input buffer and signals the CPU to parse the captured data and build frame buffers.

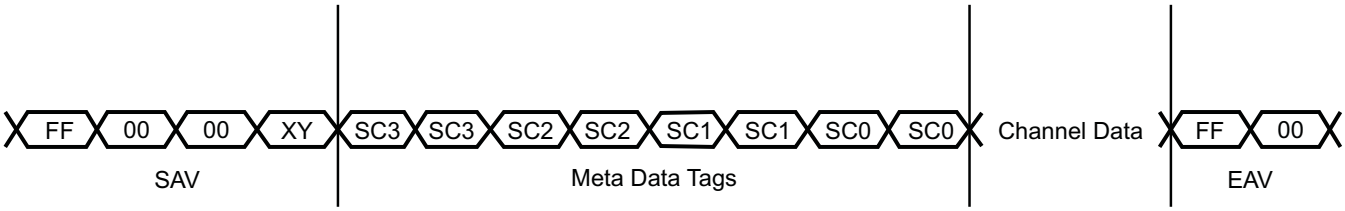
The startcodes used for the Metadata wrapped lines and the dummy lines form the Super Frame.

The dss_vip_parser has logic to parse out the super frame, analyze the Metadata tags, and frame buffer the line contents appropriately.

#### 12.2.8.2.7.6 8-bit Data Interface in Line Multiplexing

Figure 12-114 is an example of an 8-bit line multiplexing interface. Channel Data is the CbYCrY sequence representing a line. Preceding Channel Data is the four byte Meta Data tags. Note that the Meta Data bytes are replicated in both the Luma and the Chroma sites. The entire structure is bounded by a traditional SAV/EAV code in which the V flag is 0.

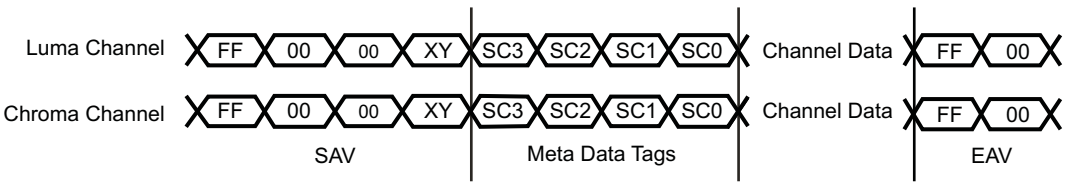
**Figure 12-114. 8-bit Line Mux Interface**




#### 12.2.8.2.7.7 16-bit Data Interface in Line Multiplexing

Figure 12-115 describes the 16-bit line multiplexing interface. Channel Data is the active line. All the Y pixels are in the Luma Channel. The CbCr pixels are in the Chroma Channel. Each cycle, a 16 bit value representing one Luma sample and one Chroma sample enters the VIP Parser. The Meta Data tags are replicated in both channels. Likewise, the SAV/EAV startcodes are found in both channels. The V flags for the SAV/EAV startcodes are always 0.

**Figure 12-115. 16-bit Line Mux Interface**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.7.8 Split Lines in Line Multiplex Mode

Suppose an external device is sending two dissimilar sources in Line Multiplex mode. One narrower source has X pixels per line. The wider source has 2X pixels per line. In an older DSP, the capture buffer for the Super Frame must be configured to be the width of the largest incoming line. So, lines from the narrower source would take up only half of the allocated space in the capture buffer resulting in wasted DDR memory space.

The Meta Data has provisions for the external device to split a line. The Beginning of Line (BOL) and End of Line (EOL) flags tag a split line as described in Table 12-58.

**Table 12-58. Split Line Table**

| BOL | EOL | Function |
|---|---|---|
| 0 | 0 | Undefined |
| 0 | 1 | Line Segment is the second half of a line |
| 1 | 0 | Line Segment is the first half of a line |
| 1 | 1 | Line has not been segmented into two. |

#### 12.2.8.2.7.9 Meta Data

Table 12-59 shows the bitfields in the Meta Data start codes.

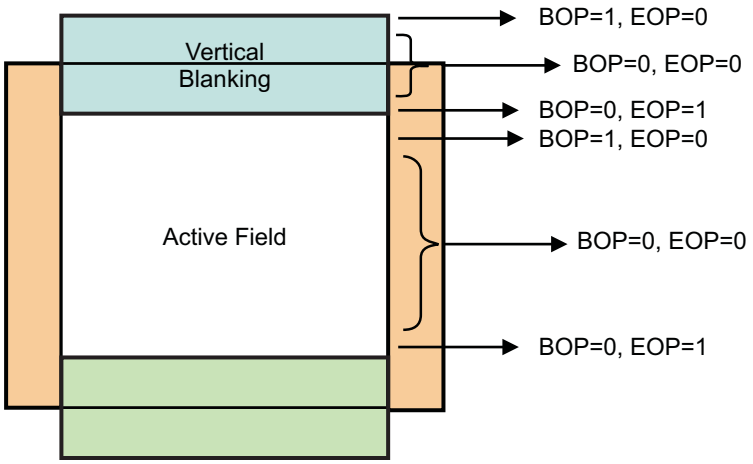
**Table 12-59. Meta Data Layout**

| Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SC3 | 1 | BOP | EOP | RSVD | CH_ID[3:0] | | | |
| SC3 | 1 | BOP | EOP | RSVD | CH_ID[3:0] | | | |
| SC2 | 0 | BOL | EOL | VDET | LINE_ID[10:7] | | | |
| SC1 | ~LINE_ID[6] | LINE_ID[6:0] | | | | | | |
| SC0 | PAD | F | V | H | P3 | P2 | P1 | P0 |

BOP tags the line as a startline in a period. A period is defined as the contiguous lines in a vertical blanking period or the contiguous lines in a field or frame. For the vertical blanking period case, the vertical blanking at the bottom of the previous field or frame combined with the vertical blanking at the top of the current field or frame is combined to create one period.

EOP tags the line as an endline in a period. Figure 12-116 shows the definition of the two types of Periods as outlined by BOP and EOP.

**Figure 12-116. BOP/EOP Definition of a Period**





Copyright © 2011–2016, Texas Instruments Incorporated

For the Split Line at the top of a Period, the BOP bit is set for both halves of the split line. Likewise, for the Split Line at the bottom of a Period, the EOP bit is set for both halves of the split line.

CH_ID is the channel ID, which tags the camera source that generated the incoming line. A maximum of 16 camera sources are support per Pixel Input Clock Domain.

LINE_ID is the line number, starting from 0 and incrementing by one for each subsequent line from the same source. The TVP5158 only supports SD sources or smaller, so only 9 bits of the line ID (bits 8:7) will be used by this external device.

PAD is a flag which tags the line as an artificially inserted padding line. When PAD is '0', the line should be discarded.

F, V, H, P3, P2, P1, and P0 are the bits representing the normal XY code. F is the Field ID associated with line, V signals when the line is in the vertical blanking, H specifies that the line is in the Horizontal Blanking, and P3:P0 are the protection bits.

Since only active video and vertical ancillary data lines are encapsulated in the Meta Data, the H bit in the SC0 byte should never be '1'.

#### *12.2.8.2.7.10 TI Line Mux Mode, Split Lines, and Channel ID Remapping*

The VIP Parser supports a maximum of 8 different Channel IDs per Port. The Channel IDs must be in the range {0:7} (3 bits). In the source multiplex, only one source can be a split line source and have the same Channel ID as one of the non-split line sources.

This scenario involves an external NTSC decoder which supports 8 D1 cameras. The external NTSC decoder will downscale the 8 sources to SIF format. However, one camera source will be sent in the multiplex as both the downscaled version and the original D1 sized version. They will both have the same Channel ID. However, the D1 version will be sent as a split-line. The source multiplex will thus have 9 maximum streams.

The dss_vip_parser, for TI Line Mux Mode only, will left shift the Channel ID by one. Bit 0 is used as an indicator whether the Source is a split-line source or a normal non-split line source. Only one of the nine inputs can be a split line source.

**Table 12-60. TI Line Mux Mode Channel ID Remapping**

| Source Input Channel ID | Channel ID Sent to VPDMA | |
|---|---|---|
| | Non-split Line | Split Line |
| 0x0 | 0x0 | 0x1 |
| 0x1 | 0x2 | 0x3 |
| 0x2 | 0x4 | 0x5 |
| 0x3 | 0x6 | 0x7 |
| 0x4 | 0x8 | 0x9 |
| 0x5 | 0xA | 0xB |
| 0x6 | 0xC | 0xD |
| 0x7 | 0xE | 0xF |

All subsequent references to the Camera Source, such as in a VPDMA return descriptor, will reference the remapped Channel ID.

The SrcIDx size status registers and the Source FID status registers reflect the remapped Channel ID when the port is in TI Line Mux mode.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.8 Channel ID Extraction for 2x/4x Multiplexed Source

##### 12.2.8.2.8.1 Channel ID Extraction Overview

For 2-way and 4-way multiplexed source, the Channel ID is either embedded in the four protection bits inside the EAV/SAV code words or in the horizontal blanking pixel data. A configuration setting determines where the VIP Parser would search for the Channel ID.

##### 12.2.8.2.8.2 Channel ID Embedded in Protection Bits for 2- and 4-Way Multiplexing

The four-bit channel ID is an identifier corresponding with the source number (camera) of the incoming video. As shown in Table 12-61, the Channel ID is placed in the code fourth byte of the EAV/SAV code words normally used for protection bits. With 4 bits, the maximum number of sources that can be defined in this range is 16. However, only Channel IDs in the range {0:7} are supported. Obviously, error correction cannot be performed on the FVH flags since the protection bits are no longer there.

**Table 12-61. Channel ID Embedded in EAV/SAV**

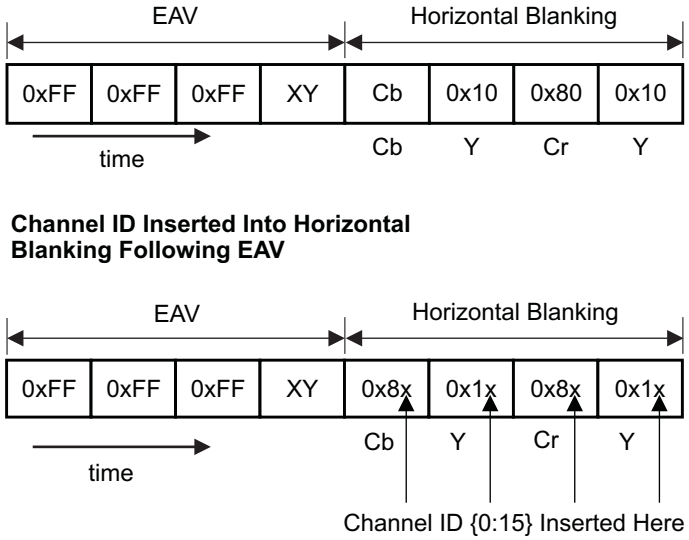
| 7 (fixed) | 6 (F) | 5 (V) | 4 (H) | 3 (ch_id[3]) | 2 (ch_id[2]) | 1 (ch_id[1]) | 0 (ch_id[0]) | Description |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | Ch_id = {0:15} | | | | SAV, Field 0, Active Video |
| 1 | 0 | 0 | 1 | Ch_id = {0:15} | | | | EAV, Field 0, Horizontal Blanking |
| 1 | 0 | 1 | 0 | Ch_id = {0:15} | | | | SAV, Field 0, Vertical Blanking |
| 1 | 0 | 1 | 1 | Ch_id = {0:15} | | | | EAV, Field 0, Horizontal Blanking in Vertical Blanking Region |
| 1 | 1 | 0 | 0 | Ch_id = {0:15} | | | | SAV, Field 1, Active Video |
| 1 | 1 | 0 | 1 | Ch_id = {0:15} | | | | EAV, Field 1, Horizontal Blanking |
| 1 | 1 | 1 | 0 | Ch_id = {0:15} | | | | SAV, Field 1, Vertical Blanking |
| 1 | 1 | 1 | 1 | Ch_id = {0:15} | | | | EAV, Field 1, Horizontal Blanking in Vertical Blanking Region |

##### 12.2.8.2.8.3 Channel ID Embedded in Horizontal Blanking Pixel Data for 2- and 4-Way Multiplexing

In Horizontal Blanking and Vertical Blanking, non-ancillary data pixels should be Y=0x10 and Cb=Cr=0x80. When the Channel ID is embedded in the Horizontal Blanking for 2 and 4-way multiplexing, the lower nibbles of all Luma and Chroma pixels are replaced by the 4 bit Channel ID. This scenario is shown in Figure 12-117. The maximum number of values defined by this 4-bit range is $2^4$ = 16. However, only Channel IDs in the range {0:7} are supported.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-117. Channel ID Inserted Into Horizontal Blanking**




#### 12.2.8.2.9 Embedded Sync Mux Modes and Data Bus Widths

Legal combinations of Embedded Sync Mux Modes and Data Bus Widths are described in Table 12-62.

**Table 12-62. Valid Embedded Sync Mux Mode and Data Bus Width Combinations**

| | 1x Mux | 2x Mux | 4x Mux | Line Mux |
|---|---|---|---|---|
| 8 Bit | √ | √ | √ | √ |
| 16 Bit | √ | n/a | n/a | √ |
| 24 Bit | √ | n/a | n/a | n/a |

#### 12.2.8.2.10 Interrupts

The VIP parser provides one interrupt line to the HDVPSS. The source for this interrupt can be one or more of 22 events.

When an interrupt occurs and is determined to be from the VIP parser, VIP parser level of masks, clears, and status registers must be checked and updated first.

The VIP Parser Interrupt Mask, Interrupt Clear, and Interrupt Status register layout is shown in Table 12-63.

**Table 12-63. Register Layout**

| 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|
| PrtB Disable Comple | PrtA Disable Complete | PrtB Anc ProtoVio | PrtB YUV ProtoVio | PrtA Anc ProtoVio | PrtA YUV ProtoVio | PrtB Src0Size | PrtA Src0Size | PrtB DisConn | PrtB Conn | PrtA Disconn |
| **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| PrtA Conn | OpPrtB Anc | | OpPrtB YUV | OpPrtA Anc | | OpPrtA YUV | InPrtB | InPrtA | PrtB Vdet | PrtAV det |


Copyright © 2011–2016, Texas Instruments Incorporated

Table 12-64 describes each of the interrupts supported by the VIP Parser.

**Table 12-64. VIP Parser Interrupts**

| Interrupt | Description |
|---|---|
| PrtBDisableComplete | When Port B is disabled in the middle of the frame, this interrupt is activated when complete frame is sent out of Port B following disable. |
| PrtADisableComplete | When Port A is disabled in the middle of the frame, this interrupt is activated when complete frame is sent out of Port A following disable. |
| PrtBANCProtocolVio | This interrupt is enabled when the protocol checker on the output of the dss_vip_parser encounters a violation on the Ancillary VPI of Port B. |
| PrtBYUVProtocolVio | This interrupt is enabled when the protocol checker on the output of the dss_vip_parser encounters a violation on the Active Video VPI of Port B. |
| PrtAANCProtocolVio | This interrupt is enabled when the protocol checker on the output of the dss_vip_parser encounters a violation on the Ancillary VPI of Port A. |
| PrtAYUVProtocolVio | This interrupt is enabled when the protocol checker on the output of the dss_vip_parser encounters a violation on the Active Video VPI of Port A. |
| PrtBSrc0Size | The output size for Srcnum=0 on Port B differs from the SRC0_NUMLINES and SRC0_NUMPIX register settings |
| PrtASrc0Size | The output size for Srcnum=0 on Port A differs from the SRC0_NUMLINES and SRC0_NUMPIX register settings |
| PrtBDisConn | Port B Link Disconnect for Srcnum 0 |
| PrtBConn | Port B Link Connect for Srcnum 0 |
| PrtADisConn | Port A Link Disconnect for Srcnum 0 |
| PrtAConn | Port A Link Connect for Srcnum 0 |
| OpPrtBAnc | Overflow at Ancillary Data VPDMA interface for the Port B |
| OpPrtBYUV | Overflow at Luma VPDMA interface for Port B |
| OpPrtAAnc | Overflow at Ancillary Data VPDMA interface for the Port A |
| OpPrtAYUV | Overflow at Luma VPDMA interface for Port A |
| InPrtB | Overflow at Input Async FIFO for Port B |
| InPrtA | Overflow at Input Async FIFO for Port A |
| PrtBVdet | Video Detect Interrupt for Port B |
| PrtAVdet | Video Detect Interrupt for Port A |

- A ‘1’ in the Status register associated with an Interrupt source shows that the interrupt source is pending. The Status register is read-only. To clear a bit in the Status register, the associated bit in the Clear register must be written with a ‘1.’
- A ’1’ the bit position of the Mask register associated with an Interrupt source ensures that the hardware interrupt will never be passed on to the DSS interrupt controller. A ‘0’ in the bit position of the Mask register associated with an Interrupt source will cause the DSS interrupt controller to see a VIP Parser interrupt in the event the hardware in the parser triggers it.
- A ‘1’ in the bit position of the Clear register associated with an Interrupt source clears the hardware interrupt status register until the next time the hardware triggers it. After a Clear, the CPU should set the bit back to a ‘0.’ Otherwise, the hardware would not be able to set any subsequent interrupts of the same type.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.11 VDET Interrupt

For Line Multiplexing Embedded Sync mode only, the Meta Data Header includes a Video Detect (VDET) flag. The TVP5158 sets this VDET flag whenever NTSC or PAL sync is found. Some other external devices using Line Multiplexing mode may not use VDET. However, when VDET changes, a VDET interrupt is issued to the DSS interrupt controller. Each Pixel Clock Input Domain has a separate VDET interrupt.

The VDET status register is comprised of 32 bits, each bit representing the value of the VDET flag found in the Meta Data of the Channel ID. Bit 0 is the VDET value from Channel ID 0, Bit 1is is VDET value from Channel ID 1, etc. There is a separate status register for each Pixel Input Clock Domain.

In Line Mux mode, the meta-data field defining the srcnum is 5 bits wide. The TVP5158 spec only defines the last three bits of this field and the upper two bits are reserved. When set to ‘1’, the cfg_tvp5158_chan_id_type register tells the VIP Parser to ignore the top two bits of the field any only use the lower three bits, giving a maximum of eight srcnums. This bit should always be set to '1' in TI Line Mux mode.

#### 12.2.8.2.12 Source Video Size

These status registers provide the size of the last active video frame in 16 different input sources per port. There is no interrupt activated on the change in the source size in any of the input sources. These read-only registers only inform the application of the width and the height of the last active field or frame associated with each channel ID.

#### 12.2.8.2.13 Clipping

The source for the clipping feature can be either embedded sync or discrete sync. Embedded sync streams use 2 reserved codewords as sync codes, 0x00 and 0xFF. These sync codes define different regions in the incoming data. Discrete sync streams have no limitation on the allowed quantization values. All 8 bits, from 0x00 to 0xFF are legal quantization values.

Suppose the input is discrete sync and the device converts the discrete sync raw video into a embedded sync stream (by adding appropriate embedded sync headers) and stores it on the network or hard disk. In this case, the processor checks every raw video byte to ensure that the reserved 0x00 and 0xFF codewords are not used in the raw video before the stream is saved. Otherwise, when the stored stream is read back by some device as an embedded sync stream, illegal codes are found.

The hardware has the capability to clip illegal codewords into legal ones. This clipping is only needed for discrete sync sourced inputs to the HDVPSS VIP module.

Embedded sync sourced input streams do not need to be clipped as they are presumably already following the rules for embedded sync.

For the embedded sync interface, input values 0x00 and 0xFF are reserved for the sync detection, so these values cannot be used in the rest of the system. The VIP Parser supports one configuration bit that, when enabled, changes all 0x00 to 0x01 and 0xFF to 0xFE in the vertical ancillary data. Another configuration bit, when enabled, changes all 0x00 to 0x01 and 0xFF to 0xFE in the active video portion of the input picture.

Note that if the clipping is enabled for ancillary data, the post processing software will never be able to find a data packet sync header, since the 00-FF-FF sequence will be changed to 01-FE-FE.

For 24- bit YUV, clipping is done on each 8 bit channel. If data[23:16]==0xFF, the clipped value will be 0xFE. If data[23:16]==0x00, the clipped value will be 0x01. Likewise, clipping is done for data bit ranges 15:8 and 7:0


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.14 Current and Last FID Value

The FID values for the current field or frame are reported in the Status Registers. When a new field or frame enters, the current FID values are saved into the previous FID status registers and the new FID value is loaded into the current FID register

Following a reset, the previous and current FID status registers are set to '1.' The first two fields or frames are ignored. On the third input field or frame after a reset, the previous FID is loaded with the current FID ('1'), and the current FID is loaded with the actual FID. By the fourth field or frame after a reset, both the previous and current FID values should represent the values found in the input stream.

The FID values are reported for each camera source in both Pixel Input Clock Domains.

#### 12.2.8.2.15 Disable Handling

When the VIP port is used in single-channel embedded or discrete sync interface, output of the VIP can be connected to the scalar or other modules. When connected with these modules, it may be required to work on frame boundary or it may cause lock up without reset. This feature makes sure that VIP is always disabled at the frame boundary.

In a scenario, assume that the VIP Parser has been processing a single input stream. Then, VIP port is disabled by setting ENABLE to '0'. The VIP Parser will continue to output data downstream until it sends out an end frame pixel and the downstream module accepts the end frame pixel.

#### 12.2.8.2.16 Picture Size Interrupt

Each VIP port can be set up to trigger an interrupt if the picture size varies from a pre-programmed expected picture size. This interrupt is supported only for the Active Video portion of the input video and not for the Vertical Ancillary portion. Also this interrupt is only support for source number 0 in multi channel capture.

The interrupts are named PrtASrc0Size and PrtBSrc0Size. They are described in Table 12-64.

For Port A, the expected active video picture size values are programmed in SRC0_NUMLINES and SRC0_NUMPIX in the VIP_PARSER_xtra_port_a register. For Port B, the expected active video picture size values are programmed in SRC0_NUMLINES and SRC0_NUMPIX in VIP_PARSER_xtra_port_b register.

Note that the Picture Size Interrupt reflects the Active Video size going out of the VIP port. If cropping is enabled for Srcnum=0, the Picture Size is the post-cropped size.

#### 12.2.8.2.17 Ancillary and Video Data Cropping

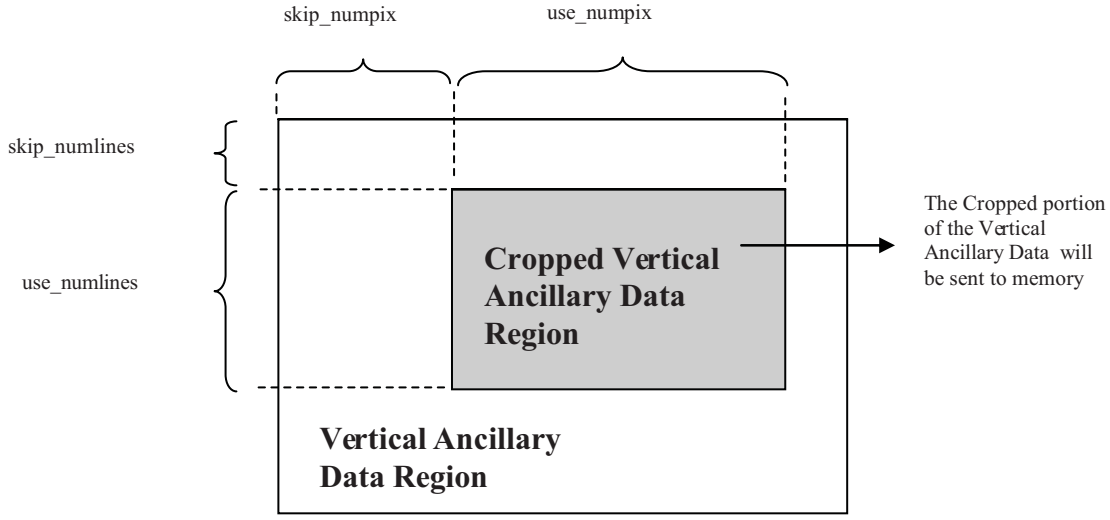
Each VIP port supports cropping of video and ancillary data. The size of the cropping window for video and ancillary data can be configured independently. This feature is supported for only one source number in a VIP port.

For the Vertical Ancillary Data from Port A, cropping is enabled by setting the ANC_BYPASS_N bit in VIP_PARSER_xtra2_port_a register. The Source Number from Port A that gets cropped is defined by the ANC_TARGET_SRCNUM in VIP_PARSER_xtra2_port_a register.

ANC_SKIP_NUMPIX, ANC_USE_NUMPIX, ANC_SKIP_NUMLINES, and ANC_USE_NUMLINES define the region of the selected Source Number that is cropped and sent to memory. The Vertical Ancillary Data Cropping region is described in Figure 12-118.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-118. Vertical Ancillary Data Cropping Region**

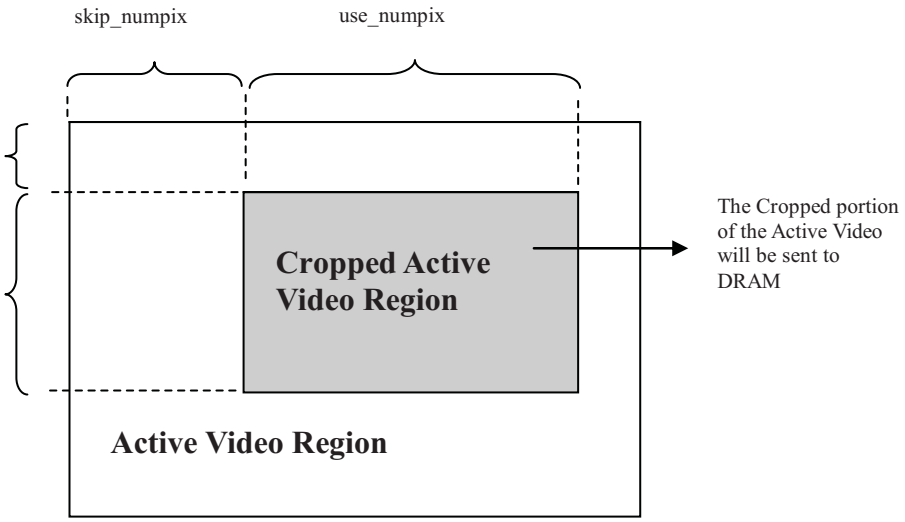



Note that when the VIP is configured to receive data over 8-bit interface, ancillary data from both Luma and Chroma can be captured by configuring ANC_CHAN_SEL_8b to '1x'. Thus, the number of data elements per line in this case is twice the equivalent number of Luma pixels per line. In other words, for this particular dual channel capture example, if there are 720 Luma pixels per line, then the total number of Vertical Ancillary Data Pixels in the source picture can be 2x720=1440 pixels.

For Active Video from Port A, cropping is enabled by setting the ACT_BYPASS_N bit in VIP_PARSER_xtra4_port_a register. The Source Number from Port A that gets cropped is defined by the ACT_TARGET_SRCNUM in VIP_PARSER_xtra2_port_b register.

ACT_SKIP_NUMPIX, ACT_USE_NUMPIX, ACT_SKIP_NUMLINES, and USE_NUMLINES define the region of the selected Source Number that is cropped and sent to memory. The Cropped Active Video region is described in Figure 12-119.

**Figure 12-119. Active Video Region**



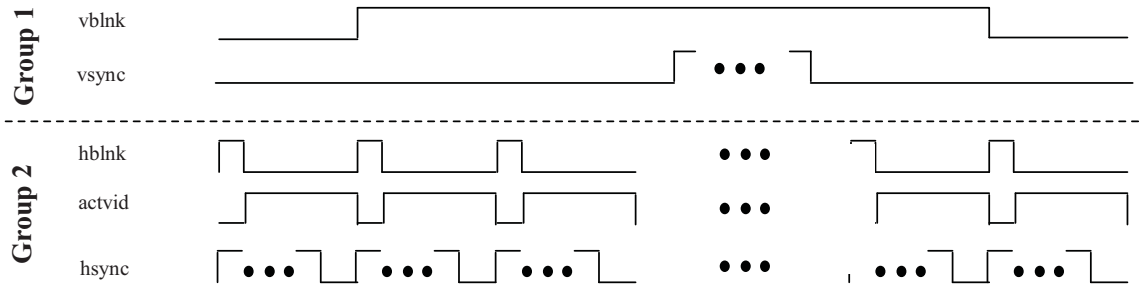

Cropping for Port B works in a similar way to Port A. Since picture data is in 4:2:2 format, SKIP_NUMPIX and USE_NUMPIX must be evenly divisible by 2. If the output of vip_parser is sent to a 4:2:2 to 4:2:0 converter, then USE_NUMLINES must also be evenly divisible by 2.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.18 Discrete Sync Signals

External ICs generally produce discrete sync interface signals seen in Figure 12-120.

**Figure 12-120. Discrete Sync Interface Signals**




**vbnk** represents the vertical blanking interval. Generally, vertical blanking is at the top of a NTSC/PAL field. In certain standards, the last few lines of a field or frame are in vertical blanking in addition to the beginning few lines in the following field or frame.

v**sync** is the vertical sync indicator. vsync is active during a portion of the vertical blanking. Generally, vsync is defined to transition from inactive to active sometime during vertical blanking. This signal then transitions to an inactive state before the end of vertical blanking. Certain standards define the line numbers where vsync transitions.

**hblank** is the horizontal blanking interval for each line. The horizontal blanking is the same number of pixels whether the line is in the active video region or in the vertical blanking region of the scan.

**actvid** is the region of a line that is active video. It is exactly inverse of the hblank signal. The number of pixels in the actvid region is the same for a line in vertical blanking as a line in active video.

**hsync** transitions from inactive to active for the first pixel of each line, which is a horizontal blanking pixel. hsync will transition to the inactive state before the end of the line.

**Group 1** signals define the vertical separation between fields or frames. Group 2 signals define the separation between lines. Note that the VIP module defines three discrete sync control signals: Actvid, Hsync, and Vsync. One of the Group 1 signals can be tied to the Vsync input. The Actvid signal from Group 2 is tied to the Actvid input. One of the other two Group 2 signals, hblank or hsync, can be tied to the Hsync input.

**USE_ACTVID_HSYNC_N** defines whether the line separation method uses the signal from the ACTVID or the HSYNC input of the VIP module. DISCRETE_BASIC_MODE determines whether discrete sync works as described in Section 12.2.8.2.5.1. In summary, choose one signal from Group 1 and one signal from Group 2. There should be a way to capture the external data.


Copyright © 2011–2016, Texas Instruments Incorporated

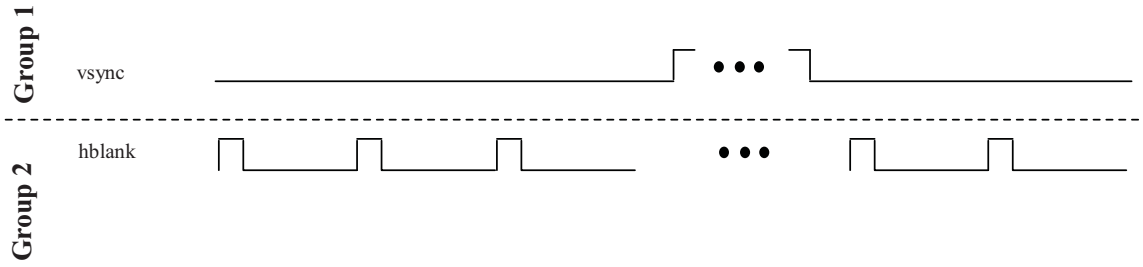
#### 12.2.8.2.18.1 vsync and hblank Input Signals

Figure 12-121 shows vsync from Group 1 and hblank from Group 2 being used. Set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='1'.

Also, since VIP is not aware of vertical blanking interval start and end, all lines, including both vertical ancillary and active video, will appear in the memory. Lines starting after an inactive to active transition on vsync will delineate a start of frame. Every data element on the Pixel clock's active edge will be stored in the Active Video Buffer.

Note that the vsync signal cannot transition active on the same cycle that hsync transitions active for fast pixel clock rates, since a system clock cycle is needed to insert a new frame indicator into the frame buffer.

**Figure 12-121. vsync and hblank Input Signals**

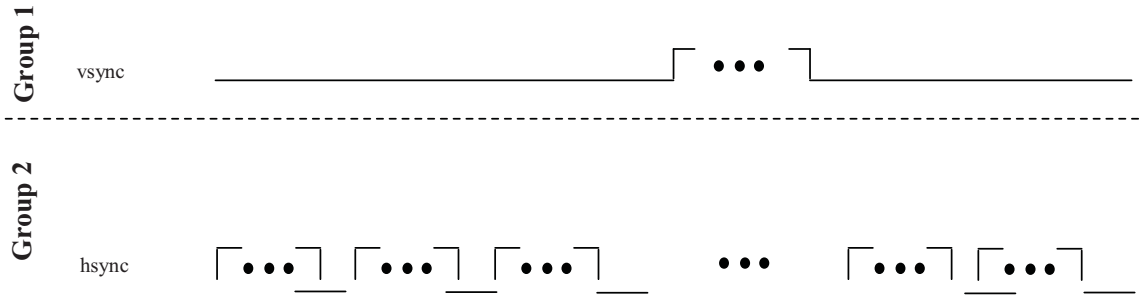



#### 12.2.8.2.18.2 vsync and hsync Input Signals

Figure 12-122 shows vsync from Group 1 and hsync from Group 2 being used. Set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='1'. Since the VIP is not aware of when VBI data starts and ends, all lines, including both vertical ancillary and active video, will appear in the memory. Lines starting after an inactive to active transition on vsync will delineate a start of frame. Every data element on the Pixel clock's active edge will be stored in the Active Video Buffer. Note that this scenario is the same as the vsync and hblank.

Like the vsync and hblank case, it also holds true that the vsync signal cannot transition active on the same cycle that hsync transitions active for fast pixelclock rates since a system clock cycle is needed to insert a new frame indicator into the framebuffer.

**Figure 12-122. vsync and hsync Input Signals**





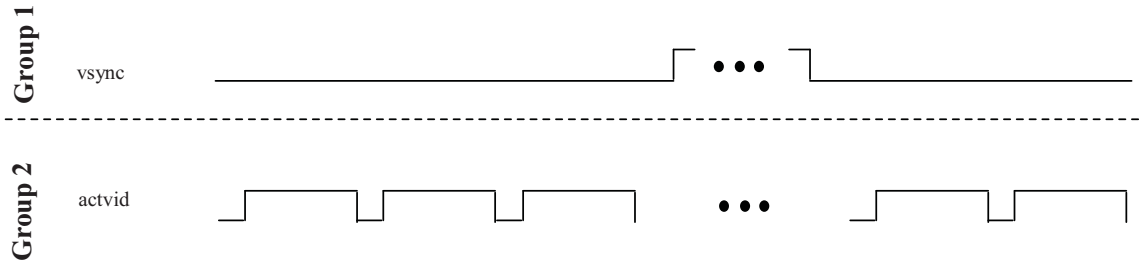
Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.18.3 vsync and actvid Input Signals

Figure 12-123 shows vsync from Group 1 and actvid from Group 2 being used. Set USE_ACTVID_HSYNC_N='1' and DISCRETE_BASIC_MODE='1'. Again, since VIP is not aware of when VBI data starts and ends, all lines, including both vertical ancillary and active video, will appear in the memory. Lines starting after an inactive to active transition on vsync will delineate a start of frame. Lines are denoted by an inactive to active transition of actvid. Pixel will be saved only when actvid is active.

Note that vsync must transition active between the actvid delineated data. vsync cannot transition active when actvid is active.

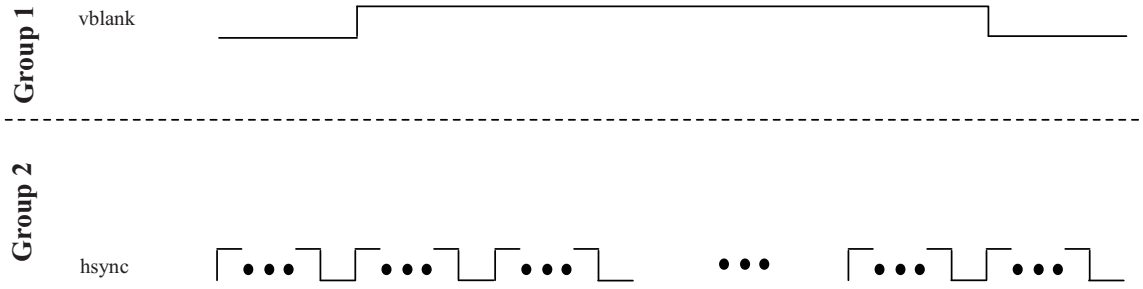
**Figure 12-123. vsync and actvid Input Signals**




#### 12.2.8.2.18.4 vblank and hsync Input Signals

Figure 12-124 shows vblank from Group 1 and hsync from Group 2 being used. In this scenario, set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='0'. Again, since VIP is aware of the start and end of vertical blanking interval start and end, vertical ancillary and active video data will appear in different memory buffers. Lines starting from inactive to active transition on vblank will delineate a start of the frame. When vblank is active, all data elements on pixel clock's active edge will appear in ancillary data buffer and when vblank is inactive, all data elements on pixel clock's active edge will appear in video data buffer.

**Figure 12-124. vblank and hsync Input Signals**



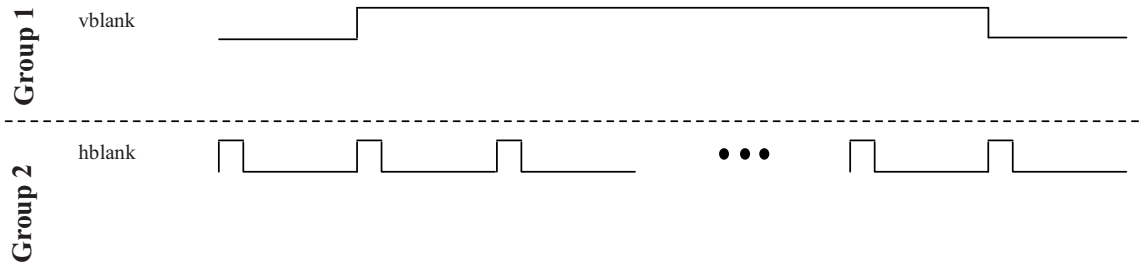


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.18.5 vblank and hblank Input Signals

Figure 12-125 shows the vblank from group1 and hblank from group2 signals being used. In this case, set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='0'. Again, since VIP is aware of the start and end of vertical blanking interval start and end, vertical ancillary and active video data will appear in different memory buffers. Lines starting from inactive to active transition on vblank will delineate a start of the frame. When vblank is active, all data elements on pixel clock's active edge will appear in ancillary data buffer and when vblank is inactive, all data elements on pixel clock's active edge will appear in video data buffer.

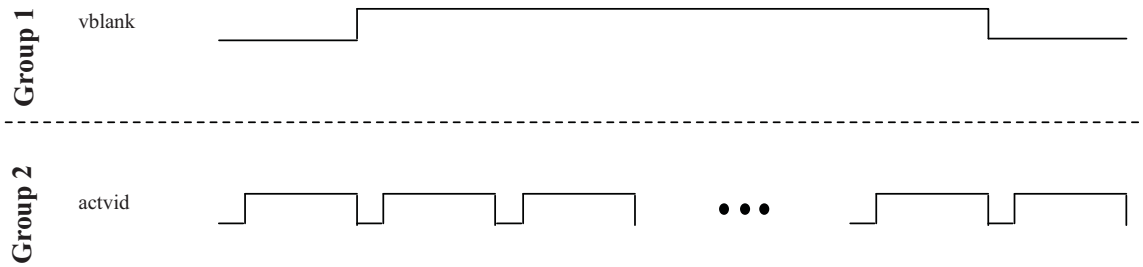
**Figure 12-125. vblank and hblank Input Signals**




#### 12.2.8.2.18.6 vblank and actvid Input Signals

Figure 12-126 shows vblank from Group 1 and actvid from Group 2 being used. Set USE_ACTVID_HSYNC_N='1' and DISCRETE_BASIC_MODE='0'. Again, since VIP is aware of the start and end of vertical blanking interval start and end, vertical ancillary and active video data will appear in different memory buffers. Lines starting from inactive to active transition on vblank will delineate a start of the frame. When vblank and actvid are active, all data elements on pixel clock's active edge will appear in ancillary data buffer and when vblank is inactive and active is active, all data elements on pixel clock's active edge will appear in video data buffer.

**Figure 12-126. vblank and actvid Input Signals**



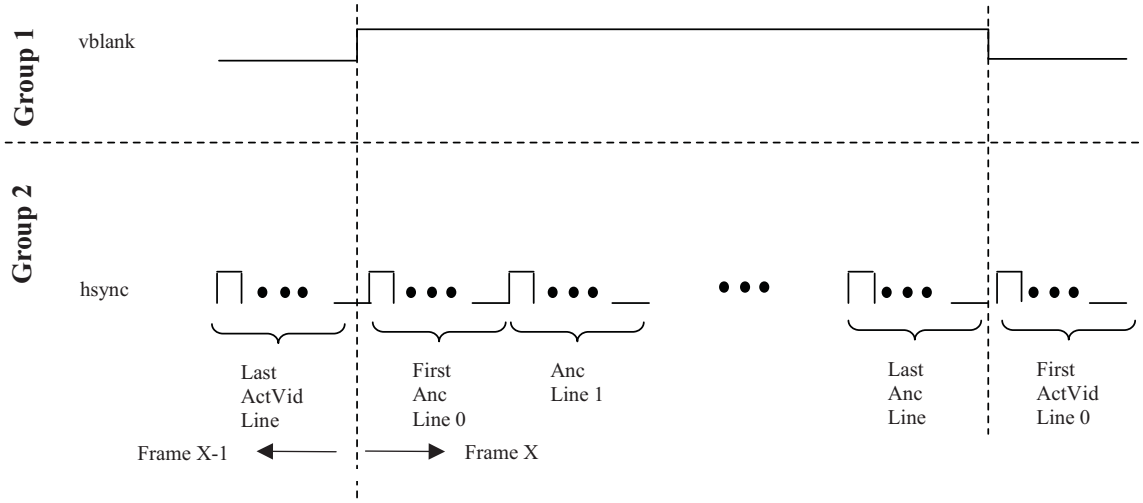


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.18.7 Line and Pixel Capture Examples

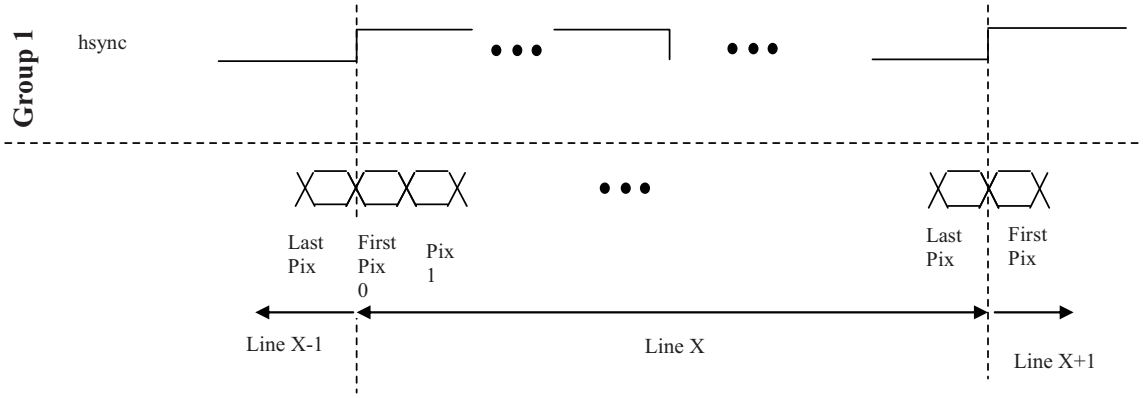
When DISCRETE_BASIC_MODE = '0', vblank is used. All the lines where the start of line is under an active vblank are sent to the Ancillary Data buffer. All the lines where the start of line is not under an active vblank are sent to the Active Video framebuffer. This situation is shown in Figure 12-127.

**Figure 12-127. Line and Pixel Capture, vblank and hsync**




The start of line is the pixel represented by the inactive to active transition on Hsync when USE_ACTVID_HSYNC_N = '1'. Figure 12-128 illustrates the delineation of a line when using USE_ACTVID_HSYNC_N = '1.'

**Figure 12-128. Line and Pixel Capture, hsync**

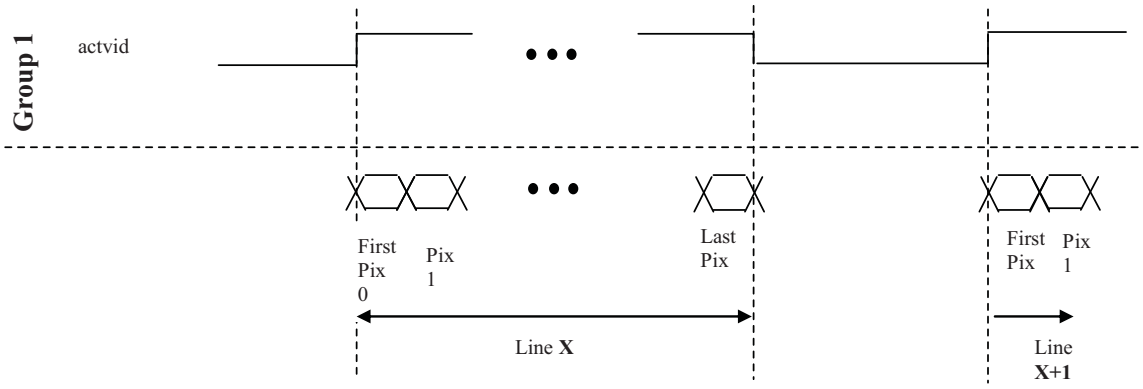




Copyright © 2011–2016, Texas Instruments Incorporated

The start of line is the pixel represented by the inactive to active transition on actvid when USE_ACTVID_HSYNC_N = '0.' Note that actvid stays active for the entire duration of active video portion of the line. This scenario is shown in Figure 12-129.

**Figure 12-129. Line and Pixel Capture, actvid**




In 8-bit mode, note that the 4:2:2 YUV input color component order is Cb, Y followed by Cr and Y. For 16 and 24-bit input modes, all the components are sent in the same cycle.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.8.2.19 VIP Overflow Detection and Recovery

It is possible that an overflow can occur in the VIP_PARSER. Overflow detection is determined by reading the VIP_PARSER_fiq_status register and checking for bits 8, 7, 5, 4, 3 and 2. If video is being captured, and any of these bits are set, it indicates that not all of the incoming video data was sent to DDR. VIP Overflow can be caused by one of the following:

1. External pixel clock is faster than processing clock
2. DDR bandwidth is temporarily over-consumed
3. VIP scaler is being used inline with external video input, and is upscaling.
   - VIP scaler in this use case can only be used for downscaling
4. VIP scaler is being used inline with external video input, but has not been configured with scaler coefficients
   - VIP scaler will not accept video input if it is not first configured with scaler coefficients. This will cause overflow
5. VIP scaler is being used inline, but has not been enabled
6. External cables are connected or disconnected while the system is running, resulting in corrupted video streams going into the VIP
7. Bad external video cable, which causes corrupted video streams going into the VIP

Items 6 and 7 are typically seen as noise events, where it is likely that multiple horizontal syncs per line and/or multiple vertical syncs per frame will be observed. These result in high peak throughput requirements, leading to DDR bandwidth being temporarily over-consumed, and thus VIP overflow.

The high level recovery method for VIP Overflow on port a is outlined below. Port b is similar :

1. Set yuv_srcnum_stop_immediately = 0xFFFF_FFFF
2. Set anc_srcnum_stop_immediately = 0xFFFF_FFFF
3. Set ENABLE = 0
4. Set CLR_ASYNC_FIFO_RD and CLR_ASYNC_FIFO_WR to 1
5. Set SW_RESET to 1
6. Reset Other VIP modules
   - Write 1 to either bit 16 or bit 17 (depending on VIP0 or VIP1) of the clkc_rst register
7. Abort VPDMA channels
   - Write to list attribute to stop list 0
   - Write to list address register location of abort list
   - Write to list attribute register list 0 and size of abort list
8. Set SW_RESET to 0
9. Un-reset Other VIP modules
   - Write 0 to either bit 16 or bit 17 (depending on VIP0 or VIP1) of the clkc_rst register
10. (Delay)
11. SC coeff downloaded (if VIP_SCALER is being used)
12. (Delay)
13. Set yuv_srcnum_stop_immediately = 0x0000_0000
14. Set anc_srcnum_stop_immediately = 0x0000_0000
15. Set ENABLE = 1
16. Set CLR_ASYNC_FIFO_RD and CLR_ASYNC_FIFO_WR to 0


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.9 De-Interlacer (DEI) Module

The de-interlacer (DEI) module is used to generate progressive data output from interlaced input format (also known as, deinterlacing).

#### 12.2.9.1 Features

- Motion-adaptive deinterlacing
  - Supports 4-field Motion detection
  - Motion sensitivity adaptive to the frequency of luma texture
- Edge-Directed Interpolation (EDI)
  - Edge detection using luma pixels in a 2x7 window
  - Half-pixel resolution edge vectors
  - Edge-directed luma and chroma interpolation
  - Soft-switch between edge directed interpolation and vertical interpolation depending on the confidence factor
- Film Mode Detection (FMD)
  - 3-2 pull down detection
  - 2-2 pull down detection
  - Hysteresis controls how fast FMD can enter/exit film mode (software function)
- Bad Edit Detection (BED)
- The module can pass the inputs data directly to the outputs in a bypass configuration. No internal processing is performed.

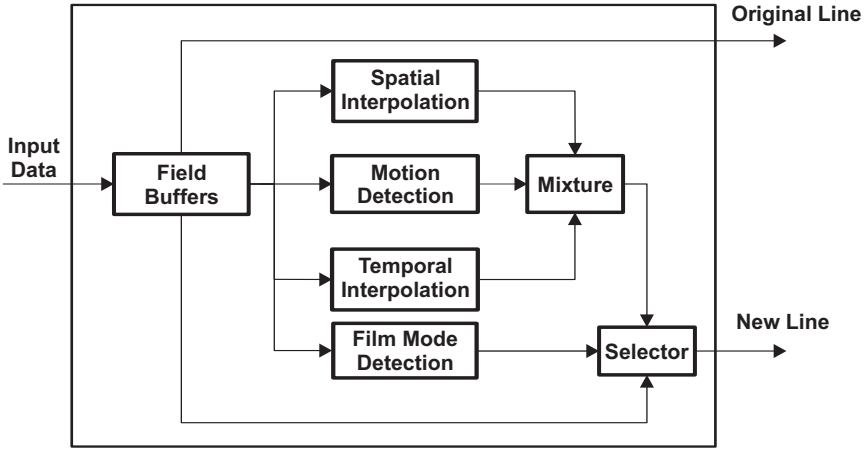
#### 12.2.9.2 Functional Description

The following figure illustrates the block-diagram of motion-adaptive Deinterlacer. The general concept behind motion adaptive deinterlacing is that spatial filtering works very well for images with motion, while temporal filtering works very well for static images. So, the intuitive way is to combine them together. Motion detection is used to switch or fade between the use of spatial deinterlacing and temporal deinterlacing, as shown in the following formula.

$$\hat{y}(j, i, n) = \alpha y_{spat}(j, i, n) + (1 - \alpha) y_{temp}(j, i, n)$$

where $y_{spat}(j, i, n)$ is the spatial interpolation output, $y_{temp}(j, i, n)$ is temporal interpolation output, $\alpha$ is the motion detection output ranging from 0 to 1, $\hat{y}(j, i, n)$ is the final output from deinterlacer, and j, i, n are the vertical, horizontal, and temporal indexes, respectively. From the previous formula, the final output is controlled by the motion detector output, $\alpha$. The higher the motion, the higher value of $\alpha$, and the output favors spatial interpolation. If the motion is absent or very low, the temporal interpolation has higher weight.

**Figure 12-130. Block Diagram of Motion-Adaptive Deinterlacer**





Copyright © 2011–2016, Texas Instruments Incorporated

Temporal interpolation can be disabled by programmable control registers. In that case we have:

$\hat{y}(j, i, n) = y_{spat}(j, i, n)$

Chroma interpolation is handled in the same manner as luma with regards to the above equations. There is a separate control for disabling temporal interpolation for chroma, such that luma can perform the full mixture, and chroma can only be spatial. If luma temporal interpolation is disabled, chroma temporal interpolation is also disabled.

The Figure 12-130 provides a simple description of how a motion-adaptive deinterlacer operates. The actual interpolation used is edge directed interpolation. Edge directed interpolation is only performed spatially prior to the output mixing.

#### 12.2.9.2.1 Modes of Operation

The DEI can be operated in bypass mode and deinterlacer mode.

##### 12.2.9.2.1.1 Bypass Mode

In the bypass mode, input luma and chroma are buffered and sent to the stage after DEI without processing.

To set DEI in the bypass mode, the bypass bit must be set accordingly in the configuration register.

##### 12.2.9.2.1.2 Film Mode Detection (FMD)

Film mode detection is performed by both hardware and software. In every field, the hardware accumulates difference between two neighboring frames (two top fields or two bottom fields), difference between two adjacent fields, and combing artifact. At the end of DEI data processing, an interrupt will be sent to the processor. The processor needs to read three status registers from DEI, calculate whether the film mode has been entered or existed, and update film mode lock and jam direction (fmd_cfg0) MMR of DEI before the next field starts. Since it is a time sensitive processing, the FMD interrupt has to be a high priority interrupt to the processor.

Film mode detection can be performed on a full picture size or a smaller window specified by a few MMR registers. The interrupt is always asserted at the end of the entire DEI data processing and is not affected by the window size and location.

Implementing film mode detection algorithm in software allows more flexibility at the cost of timing critical interface between DEI and the processor. For example, customers may find later that supporting 3-2-2-3 pull down is needed. This feature can be easily added by upgrading only the FMD software.

FMD feature is supported by the DEI module. More details about FMD mode, they are discussed in the DEI_H section.

See the appendix for configuration details on FMD.

##### 12.2.9.2.1.3 Deinterlacer Mode

In the deinterlacer mode, DEI takes in alternative field YUV data and sends out a sequential frame YUV data by interpolation techniques mentioned below.

DEI supports four interpolation modes when converting interlaced pictures to progressive pictures.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.9.2.2 Modes of Interpolation

##### 12.2.9.2.2.1 Interpolation Mode 0

In mode 0, the interpolated field is created by simple line averaging from the original YUV data. That is, the interpolated line is created by averaging its top and bottom line. Setting of MDT mode has no effect on output pictures.

##### 12.2.9.2.2.2 Interpolation Mode 1

In mode 1, the interpolated field is created by averaging pixels from fields before and after the current field. In other words, if the current field is a top field, the interpolated bottom field picture is created by averaging pixels from bottom field pictures before and after the current field. MDT setting has no effect on the result.

##### 12.2.9.2.2.3 Interpolation Mode 2 (EDI)

Mode 2 is an edge assisted interlace mode with edges detected from the luma information of a 2 × 7 (H × W) frame window. It detects seven possible edges between 45 and 135 degrees. Edges with slopes greater than 135 degrees are treated as 135 degree lines, and edges with slopes less than 45 degrees are treated as 45 degree lines.

Luma for missing lines are interpolated using original luma along the detected edge.

Motion value (MV) from the MDT module is used to select coefficients from a look up table on how 2D interpolation from the current field and 3D interpolation from two fields adjacent to the current fields are blended. The way to perform blending for chroma is slightly different than for luma, because the MV is based on luma and extra care needs to be taken when blending is applied to chroma.

At this mode, edge-directed interpolation is applied to luma only. Vertical interpolation is performed for chroma data.

##### 12.2.9.2.2.4 Interpolation Mode 3 (EDI)

The edge detection method used in this mode is similar to interpolation mode 2. The only difference is that the edge-directed interpolation is performed on both Luma and Chroma. Chroma is interpolated similarly according to the edge vectors obtained based on luma information. The reason chroma is interpolated slightly different is because of the down-sampled chroma data.

#### 12.2.9.2.3 Motion Detection (MDT)

Motion values from the MDT module are calculated only based on luma information. The motion value is used to determine how 2D and 3D interpolated results should be blended. The motion value is also used in the Film Mode Detection (FMD) module to determine field and frame differences between frames, which are in turn used in film mode detection.

The motion detection module in DEI supports 4-field mode.

Motion value (MV) is first estimated from y_f0 and y_f2 (current and two field delayed Y). The max value of MV and one field delayed data of MV, MV_f1, is used to create the overall MV per pixel. Since this motion detection method utilizes three fields of luma data and one field delayed MV, it is called 4-field mode as MVs come from four fields of luma data.

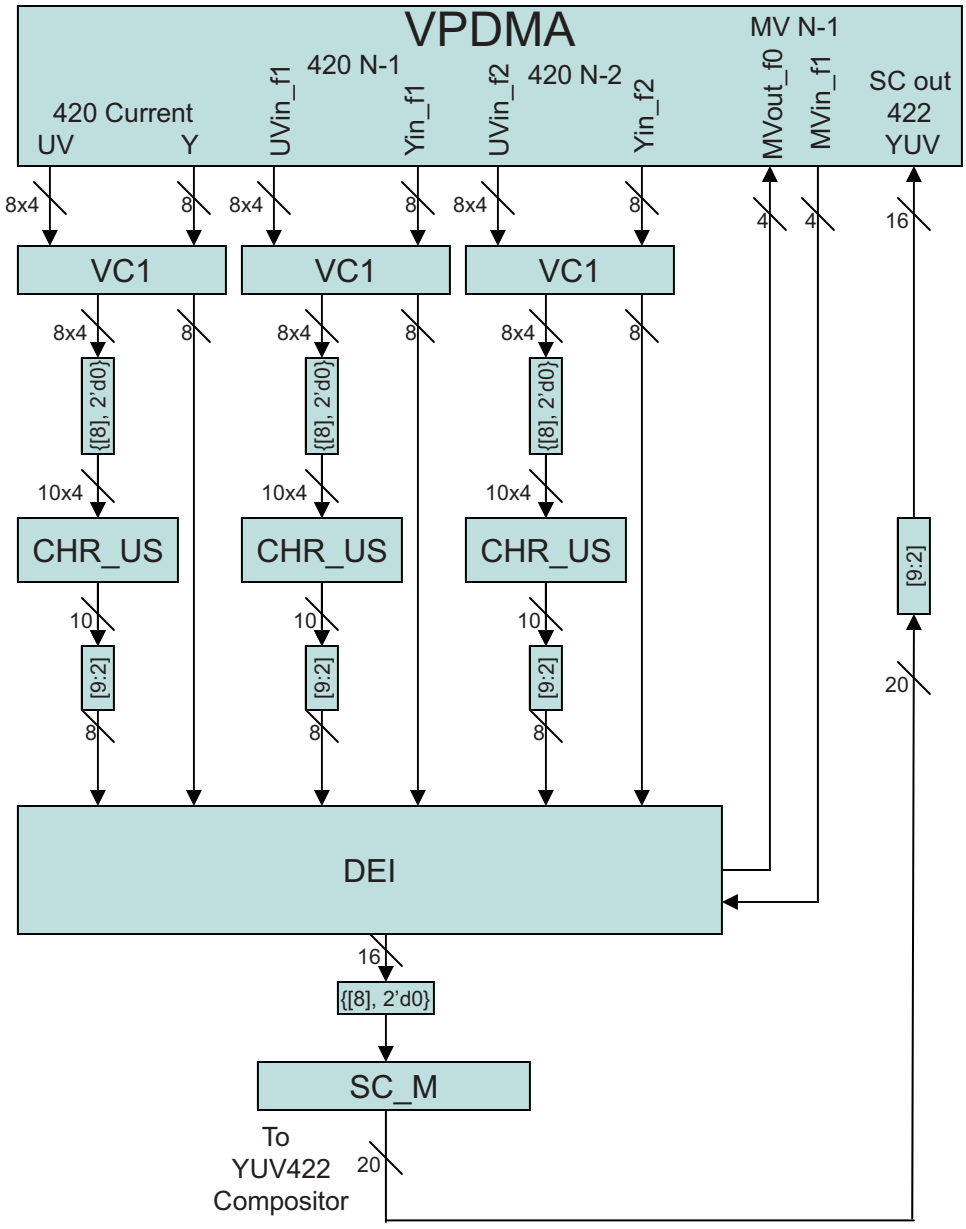
There is one field delay between the YUV inputs and the YUV outputs. The “current_field” that the interpolator works on is the YUV_f1 data.

#### 12.2.9.2.4 Data Buffer Management

VPDMA is used to transfer the data in and out of DEI as shown in the following figure.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-131. VPDMA Transfer Ports**




The following are the VPDMA ports (from DEI to memory) available:

- Inputs
  - Current (N)
    - Y (current Luma)
    - UV (current Chroma
  - One field delay (N-1)
    - Yin_f1 (1-field delayed Luma)
    - UVin_f1 (1-field delayed Chroma)
    - MVin_f1 (1-field delayed Motion Vectors)
  - Two field delay (N-2)
    - Yin_f2 (2-field delayed Luma)
    - UVin_f2 (2-field delayed Chroma)


Copyright © 2011–2016, Texas Instruments Incorporated

- Outputs
  - Current
    - Scalar output (422 interleaved output from scalar)
    - MVout_f0 (Motion vectors)

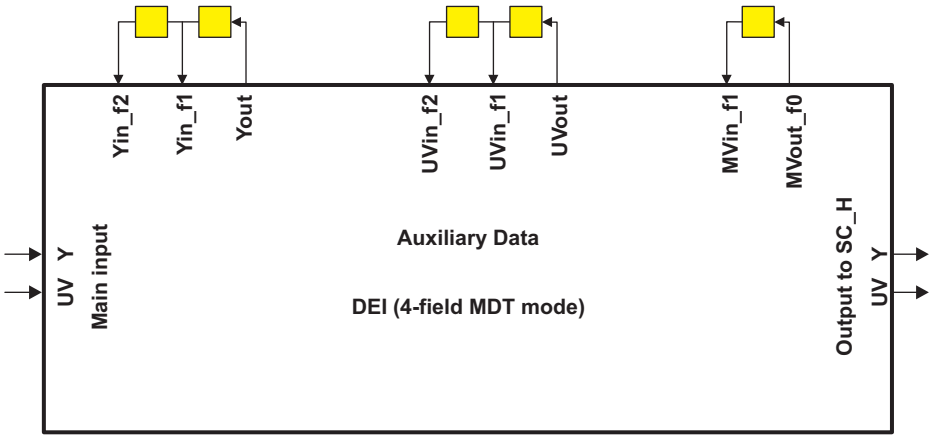
In the previously listed ports, except Y and UV (current) and Scalar output (current) (referred to as main data), all the other ports directly interact with DEI through VPDMA and are referred to as auxiliary data for DEI module. All the Y and UV ports in auxiliary data require 8 bits/pixel while MV port requires 4 bits/pixel.

#### 12.2.9.2.4.1 4-field MDT-based Interpolation Mode

In this mode, the required external memories are three fields of Luma input, three fields of chroma input, and two fields of 4-bit MV.

- Inputs
  - Y
  - UV
  - Yin_f1
  - Yin_f2
  - UVin_f1
  - UVin_f2
  - MVin_f1
- Outputs
  - MVout_f0
  - Output Scalar

**Figure 12-132. Auxiliary Data**





Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.9.3 Interrupts

The following two interrupts are generated by the module.

**Table 12-65. Module Interrupts**

| Interrupt | Description |
|---|---|
| dei_fm_int | Interrupt indicating film mode is enabled and software needs to read Film Mode statistics registers and update the Film Mode lock and jam direction configuration registers. This is required to be a high priority interrupt. |
| dei_error_int | Interrupt indicating an error condition has occurred |

#### *12.2.9.3.1 Interrupt Description*

Two interrupts from DEI module are mapped to HD-VPSS level interrupt handler. Those are discussed below.

##### *12.2.9.3.1.1 dei_fm_int*

This interrupt is controlled by enabling film mode detection. Once enabled, this interrupt will occur on every input field and tells software that it needs to read the fmd_reset, fmd_frame_diff, fmd_field_diff and fmd_caf registers and execute the Film Mode software program to determine whether to lock film mode and the jamming direction.

This interrupt needs to be serviced as a high priority interrupt. The position the interrupt is generated within a field can be controlled through configuration registers (fmd_window_enable, fmd_window_minx, fmd_window_miny, fmd_window_maxx and fmd_window_maxy). If the window is enabled, the statitics will be gathered over the window bounded by the other register settings, and the interrupt will be generated at the bottom of the window, meaning, the window can be set at the top of the field to generate the interrupt as soon as possible. This interrupt needs to be serviced before the next field is input to the deinterlacer.

The interrupt is fired at the end of the defined film mode window. If the FMD window is not enabled, the entire frame size is used, and the interrupt will occur at the end of the field. Software services the interrupt and then applies “lock and jam” MMR controls to put the design in film mode. This must occur before the next field input. If the entire window is used, there is very little time to perform this servicing and register setting when the design is used in a memory to memory mode. When in a memory to memory mode, and particularly if the “field_flush” control bit is low, the film mode window must be used, and set to provide as much time as possible between the end of the film window and the end of the actual input field. Generally speaking, the film window should end 1 or 2 lines from the bottom of the real field to provide 1 or 2 line times to service the interrupt and apply the lock and jam registers for the next input field. While this means that the entire window is not used for calculating film mode statistics, excluding the last 1 or 2 lines should have negligible impact.

##### *12.2.9.3.1.2 dei_error_int*

This interrupt will occur under the following conditions:

(a) The field ID of the current input frame is the same as the field ID of the last frame. The field ID corresponds to whether the input field is the “top” (all even lines) or “bottom” (all odd lines) field. Two subsequent input frames must have opposite field IDs.

(b) An end of frame input has occurred on the input field data (VPI end of frame for current field), but the internal counters have not counted to either the programmed width or height. This indicates that the input data to the Deinterlacer does not match the programmed width and height the Deinterlacer is expecting.

(c) A start of frame input has occurred on the input field data (VPI start of frame for current field), but the internal counters are not at 0. This indicates that a new video field was started before the last video field was completed.

In any of these cases, the input video data is not matching what the deinterlacer programming is expecting. Software should reconfigure the deinterlacer to match the proper input video timing.


Copyright © 2011–2016, Texas Instruments Incorporated

## 12.2.10 Noise Filter (NF)

### 12.2.10.1 Overview

Noise is one of the most important factors affecting video quality. Random noise can be sensor noise, A/D conversion noise, analog transmission noise, etc. Since human eyes are very sensitive to oscillating signals, the visual quality degenerates significantly even when the noise level is small. Video coding efficiency also suffers from noise because the video encoder has trouble in matching noisy blocks and has to spend extra bits to encode useless noise. Noise reduction plays a fundamental role in improving visual quality and coding efficiency.

There are normally two types of noise reduction techniques – spatial filter and temporal filter. Spatial filter processes each frame individually. Even though each frame becomes clean, the video may still look noisy due to frame-to-frame intensity changing. Temporal filter can remove the frame-to-frame intensity changing. But for moving areas, ghosting artifacts may occur because the pixel-to-pixel correspondence between two consecutive frames is impacted due to the processing. In the noise reduction algorithm of the NF module, a combination of spatial and temporal filter is applied to overcome the shortcomings of both techniques.

It is desirable to have the spatial and temporal filters adaptive to the noise level of the video. If the filters are too weak, the noise may be still obvious. If the filters are too strong, edges and details may be blurred and ghosting artifacts may appear. Noise levels may vary a lot within a video sequence. Automated noise estimation is necessary to measure the noise level of each video clip and adjust the parameters of the filters properly.

This document describes the Video Noise Filter (NF) which is used to reduce noise in a video sequence in order to improve video image quality and compression efficiency.

### 12.2.10.2 Features

The NF module has the following features:

- Spatial and temporal video noise filtering
- Built-in noise measurement
- Adaptive to noise and motion
- YCbCr 4:2:0 Video format support

### 12.2.10.3 Functional Description

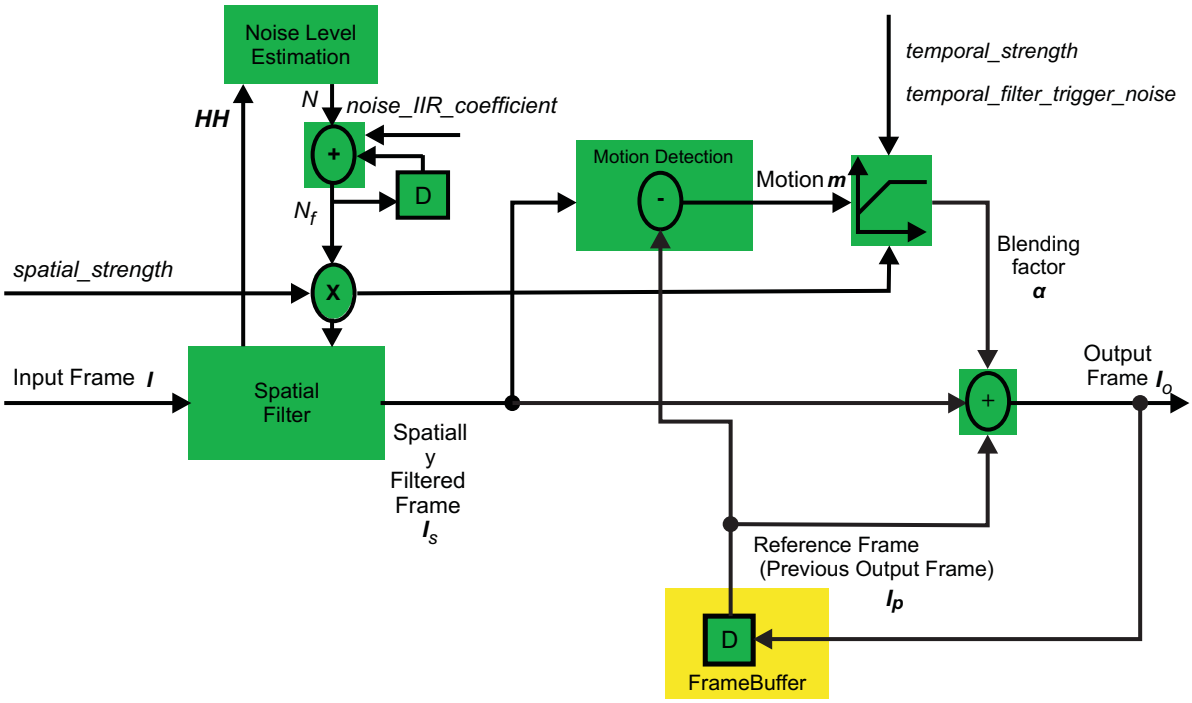
The Video Noise Filter implemented in this device is a combination of spatial and temporal IIR filters. The spatial filter is performed with neighboring pixels in the current frame while the temporal filter is performed with a past filtered frame to suppress both spatial and temporal noises.

The spatial and temporal filters are adaptive to the noise level of the video to deal with varying noise levels in different video contents. Noise estimation is performed by the hardware to measure the noise level of each video clip and to automatically adjust the parameters of the filters properly. Figure 12-133 shows the algorithmic block diagram of the noise filter.

The implementation of the video noise filter uses tile-based processing. The algorithm divides a frame into 32x32 tiles and process each of them independently. If the width or height of the frame is not a multiple of 32, 0x0s (for Luma) and 0x80s (for Chroma) will be padded into rightmost and/or bottommost tiles.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-133. Noise Filter Architecture Block Diagram**




### 12.2.10.3.1 Spatial Filter

The algorithm kernel size of the spatial filter is 3 (row) x 7 (column). With edge pixels replicated as shown in Figure 12-134, a 3x7 sliding window is used to process the entire 32x32 tile.

**Figure 12-134. Top-left Corner of a Tile After the Boundary Replication**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 00 | 00 | 00 | 00 | 01 | 02 | 03 | 04 | 05 | 06 |
| 00 | 00 | 00 | 00 | 01 | 02 | 03 | 04 | 05 | 06 |
| 10 | 10 | 10 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 20 | 20 | 20 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 30 | 30 | 30 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| 40 | 40 | 40 | 40 | 41 | 42 | 43 | 44 | 45 | 46 |

For each 3x7 input, one pixel comes out as output as shown in 3x7 Pixel Output.

**3x7 Pixel Output**

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | X |
| | | | O | | | X |
| | | | | | | X |

For U and V channels, this filtering needs to be performed only at even rows and even columns due to the 4:2:0 format.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.10.3.2 Temporal Filter

The temporal filter is motion- and noise- adaptive. The motion at each pixel is measured as the sum of the absolute differences between the spatially filtered pixel (Ys, Us, Vs) of the current frame and the output pixel (Yp, Up, Vp) of the previous frame.

**m = |Ys – Yp|+|Us – Up|+|Vs – Vp|**

**Figure 12-135. Motion versus Blending Factor Function**

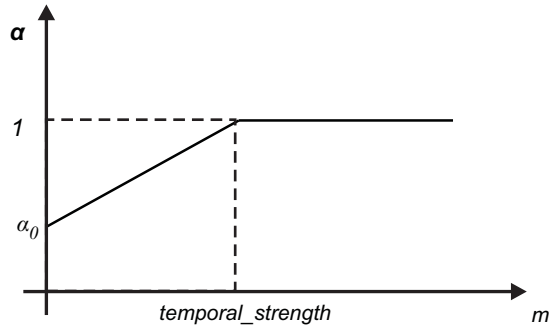



Figure 12-135 shows how the motion value, **m**, translates into the blending factor α. It can be written as:

$\alpha = \alpha_0 + (1 - \alpha_0) * m /$ temporal_strength if m < temporal strength

=1 if m >= temporal strength

where temporal_strength is a register to control the strength of the temporal filter. Setting it to 0 means the temporal filter is essentially bypassed. The discussions on the parameter $\alpha_0$ will be detailed in Section 12.2.10.3.4.

The output pixel value of the temporal filter is:

$\mathbf{I_o = \alpha I_s + (1 - \alpha) I_p}$

where $\mathbf{I_s}$ is the spatially filtered pixel value at the current frame and $\mathbf{I_p}$ is the pixel value of previous output frame.

#### 12.2.10.3.3 Noise Estimation

For each tile, the noise level *tileNoise* is the average of absolute pixel differences between the input frame **I** and the previous output frame $\mathbf{I_p}$. In the averaging, the absolute pixel differences which are smaller than *max_noise* are only taken to avoid strong moving edges. The noise level of a frame *Frame_noise* is the average noise level of all *tileNoises* in a frame. Some tiles that contain inaccurate noise level need to be excluded from the averaging calculation. They are:

(a) The boarder tiles on the four edges of a frame. They may contain zeroes that are padded to make the frame size a multiple of 32.

(b) Pure dark or pure white tiles. The max and min Y values of a tile are used to determine if a tile is pure black or pure while one. If maxY<pure_black_threshold, it is a pure black tile. If minY>255 - pure_white_threshold, it is a pure white tile.

Y, U, V channels have their own noise level respectively, denoted as *Frame_noise_Y*, *Frame_noise_U*, *Frame_noise_V*.

#### 12.2.10.3.4 Threshold and $\alpha_0$

At the end of each frame, the accumulated noise level *Frame_noise_Y*, *Frame_noise_U*, *Frame_noise_V* are processed to generate the thresholds for the spatial filter and the $\alpha_0$ of the temporal filter. The noise level will first pass an IIR low pass filter as shown in the following example.

```
Frame_noise_filtered=Frame_noise_previous* noise_IIR_coefficient +Frame_noise *(1-
 noise_IIR_coefficient )
```


Copyright © 2011–2016, Texas Instruments Incorporated

This IIR low pass filter will make the measured noise level change slowly across frames. The register *noise_IIR_coefficient* controls how fast it changes.

The thresholds of the spatial filter will be the filtered frame noise scaled by *spatial_strength*.

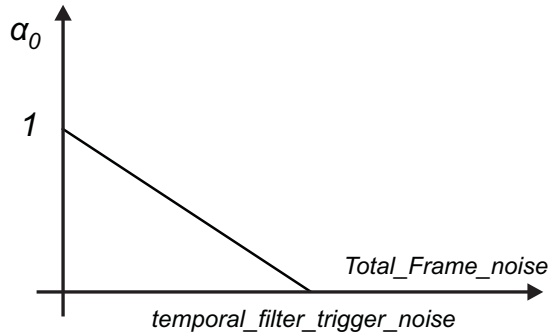
```
Threshold = spatial_strength * Frame_noise_filtered
```

$\alpha_0$ is the offset of the motion vs blending factor function. It is controlled by the total noise level *Total_Frame_noise* as shown in Figure 12-136. The total noise of all three channels is

```
Total_Frame_noise = Frame_noise_Y + Frame_noise_U + Frame_noise_V
```

where *temporal_filter_trigger_noise* is a register to control the slope of the function in Figure 12-136. This function will make $\alpha_0$ to be close to 1 if the measured noise level is low, and therefore the temporal filter to be very weak to avoid ghosting artifacts.

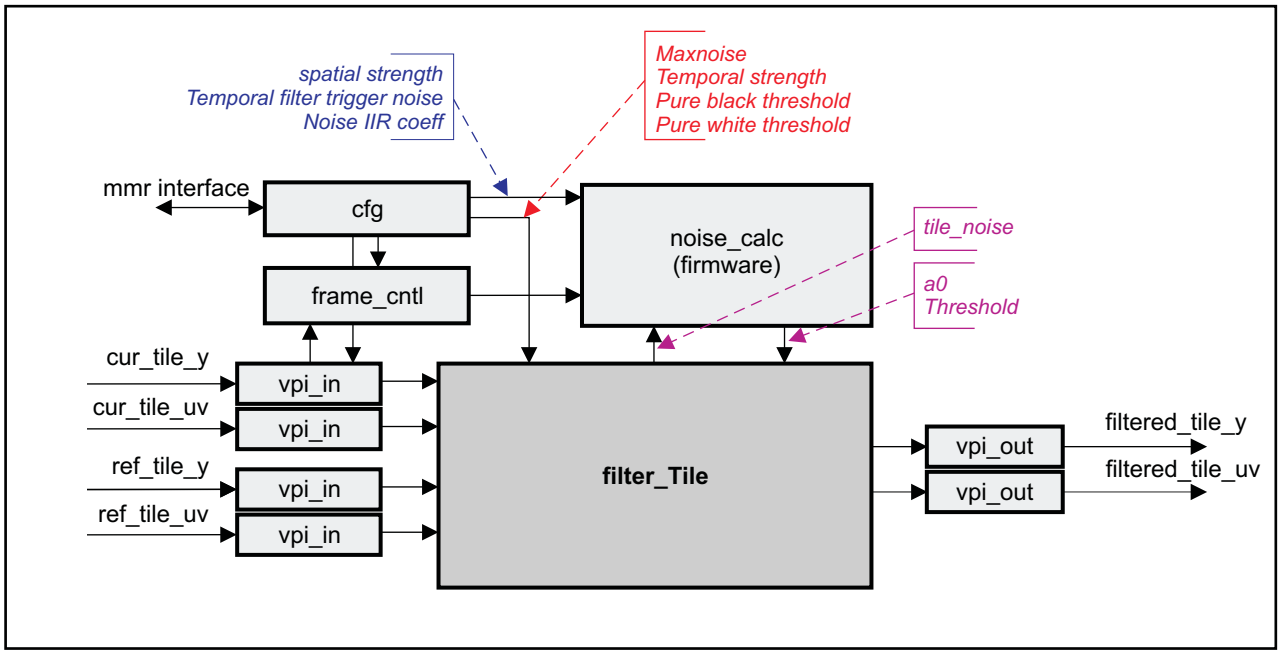
**Figure 12-136. $\alpha_0$ versus *totalFrame_noise* function**




### 12.2.10.3.5 Hardware Implementation

The NF hardware consists of the main filter_Tile processing unit which contains the main spatial and temporal filter logics and a number of support logics which include mainly MMR configuration module, VPI data in and out interface modules, frame noise calculation module (noise_calc) as shown in Figure 12-137.

**Figure 12-137. Hardware Implementation Block Diagram**

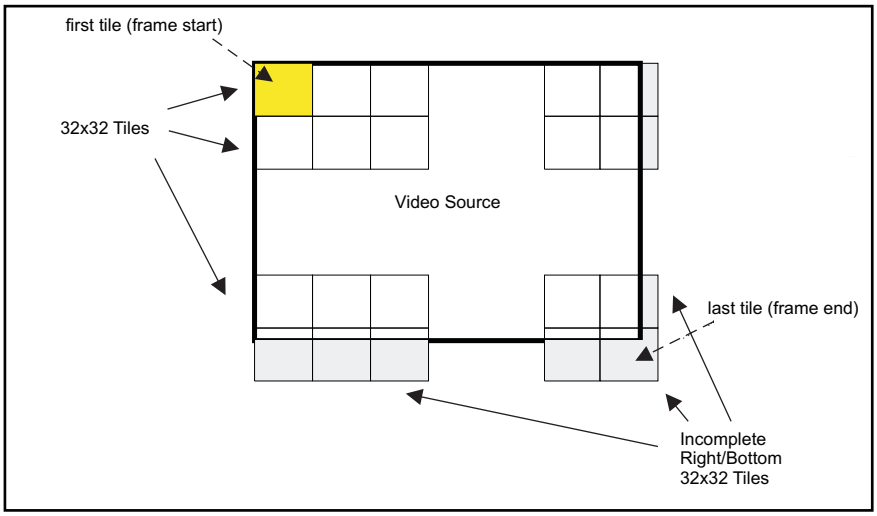




Copyright © 2011–2016, Texas Instruments Incorporated

The NF always works on blocks of 32x32 Luma and 32x16 Chroma tiles (U and V channels together). The video source size (width/height) configured in NF_REG1 can be non-multiples of 32. But, the VPDMA/CHR_DS still must send full size tiles to the NF module. Any pixels outside the valid video size (see Figure 12-138) are replaced internally by the vpi_in sub-block with 0x0 for luma and 0x80 for chroma data.

**Figure 12-138. Incomplete Boundary Tile Data Masking**




Note that ref_tile in the above diagram refers to the (N-1)th output field/frame (previous filtered_tile) while cur_tile refers to the Nth input field/frame.

The arrival of the very first tile of each video frame is used to trigger the frame processing set up and the timing of the last tile sent out is used to trigger the last frame noise calculation and saving.

The noise_calc module (which implements the task of “firmware” described in Section 12.2.10.3.4) keeps track of each tile’s noise output and generates a frame noise at the end of last tile processing.

#### *12.2.10.3.6 Video Source Multiplexing*

The hardware has 32 sets of frame noise registers to allow video source switching for up to 32 sources without the software having to read and re-initialize the frame noise values every time the video source is switched. The software simply sets the “nf_video_index” parameter of NF_REG0 register to indicate which stream the NF module is currently processing. The NF then uses the selected frame noise data to generate the threshold for the spatial filter and the $\alpha_0$ of the temporal filter for next frame processing and updates the frame noise data at the end of the frame.

If more than 32 sets of video sources are to be processed through the NF, then the software needs to manually read the saved frame noise (indexed by “frame_noise_read_index” of NF_REG7) from NF_REG8 and NF_REG9. The read noise values can then be reloaded next time the same video source is to be processed by writing to data to the NF_REG8 and NF_REG9 and also by setting the “nf_frame_noise_init_en” parameter of NF_REG0.

NF_REG0 (nf_frame_noise_update_en) bit can be used to disable the saving of collected Frame noise information in cases where sliced based filtering is used and the same filter setup is desired for all slices. In this case, the update should only be done at the beginning or end of the frame – once per frame.


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.10.4 Configuration Examples

#### *12.2.10.4.1 Example of 16-Channel Use Case*

For example, when NF is used to process 16 channels, the steps below show how we should configure for channel 12.

Frame 0, Initialization:

1. nf_video_index = 12
2. nf_frame_noise_init_en=1
3. Initialize NF_REG8 and NF_REG9 (Frame_noise_y, Frame_noise_u, Frame_noise_v) with the desired initialization values.
4. nf_frame_noise_update_en =1

Complete frame processing (the steps 1, 2, and 3 above are used to set the initialization values for noise measurement, and step 4 means the measured noise value per field/frame will be updated through the IIR filter.)

Clear the initialization flags at Frame 1:

nf_frame_noise_init_en=0

At this point, to read the measured noise levels:

Set frame_noise_read_index = 12

Read stored Frame_noise_y/u/v through NF_Reg8 and Reg9

#### *12.2.10.4.2 Basic Configurations*

NF can be easily configured to perform spatial-temporal filtering, spatial-only filtering, temporal-only filtering, or at by-pass mode. Table 12-66 shows how NF should be configured for each basic mode. Note that the temporal bypass mode here is mainly used for debug purpose, and it is not a true bypass mode.

**Table 12-66. NF Configuration for Each Basic Mode**

| Register Name | Spatial-Temporal | Spatial Only | Temporal Only | Spatial Bypass | Temporal Bypass |
|---|---|---|---|---|---|
| nf_en | 1 | 1 | 1 | 1 | 1 |
| nf_ref_cfg | b'00 | Any value, but preferred to set as b'10 or b'11 to disable DMA for reference frame | b'00 | Any value, but preferred to set as b'10 or b'11 to disable DMA for reference frame | b'00 |
| nf_bypass_cfg | b'00 | b'00 | b'00 | b'01 (or b'11) | b'10 |
| spatial_strength X 6 | | | 0 | | |
| temporal_strength | | 0 | | | |


Copyright © 2011–2016, Texas Instruments Incorporated

## 12.2.11 Scaler (SC)

The scaler module is used to resize YCbCr422 video images.

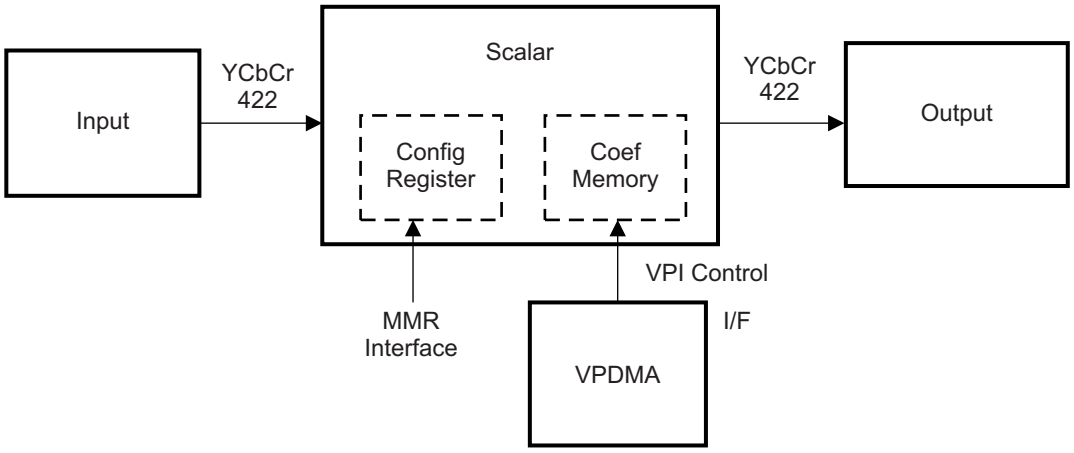
### 12.2.11.1 Features

- Independent vertical and horizontal up/down scaling
- Polyphase filter vertical up/down scaling
- Running average vertical down scaling for memory optimization
- Decimation and polyphase filtering for horizontal scaling
- Non-linear scaling for stretched/compressed left and right sides
- Input image trimmer for pan/scan support
- Pre-scaling peaking filter for enhanced sharpness
- Scale field as frame
- Interlacing of scaled output
- Full 1080p input and output support
- YCbCr422 input and output
- Maximum horizontal scaling ratio only limited by output line buffer (2047 pixels)
- Scaling filter Coefficient memory download via VPI (Video Port Interface) Control interface

### 12.2.11.2 Functional Description

Scaler takes in a 10-bit YCbCr 422 video frame from an upstream module, performs vertical/horizontal scaling, and outputs a YCbCr422 scaled image to a next downstream module. All configurations are done via the MMR interface except for the scaler coefficient memory configuration that is done via the common VPI control interface bus by VPDMA. Figure 12-139 shows the high level block diagram of the scaler module.

**Figure 12-139. High Level Block Diagram**





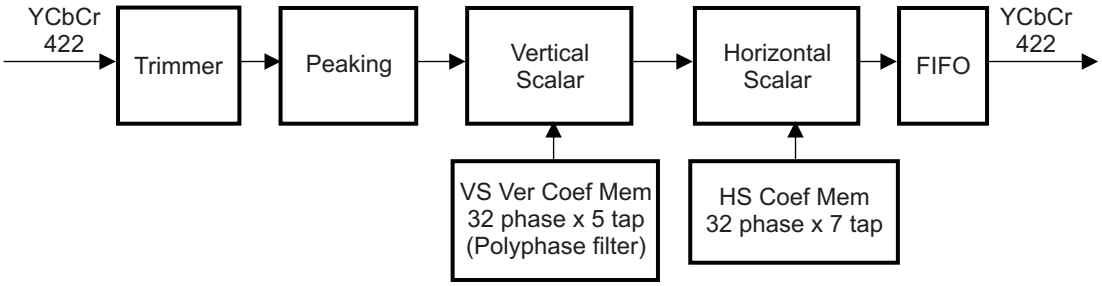
Copyright © 2011–2016, Texas Instruments Incorporated

The SC (Figure 12-140) is used in the video path and in all other video write-back data paths in the HDVPSS module.

Scaling is performed in following three steps:

1. Trimming and Pre-peaking filtering
2. Vertical Scaling (Polyphase/Running Average Filter)
3. Horizontal polyphase scaling

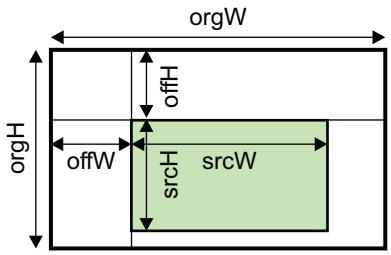
**Figure 12-140. SC Block Diagram**




#### *12.2.11.2.1 Trimmer*

The trimmer can be programmed to re-define a new source image within the input video frame sent from an upstream module before it is sent to the scaling sub-modules. This feature enables small area zoom out, source pan/scan, or removal of unwanted area in the video (such as black box/curtains/noisy line-21 video) without modifying the VPDMA parameters.

**Figure 12-141. Input Image Trimming**



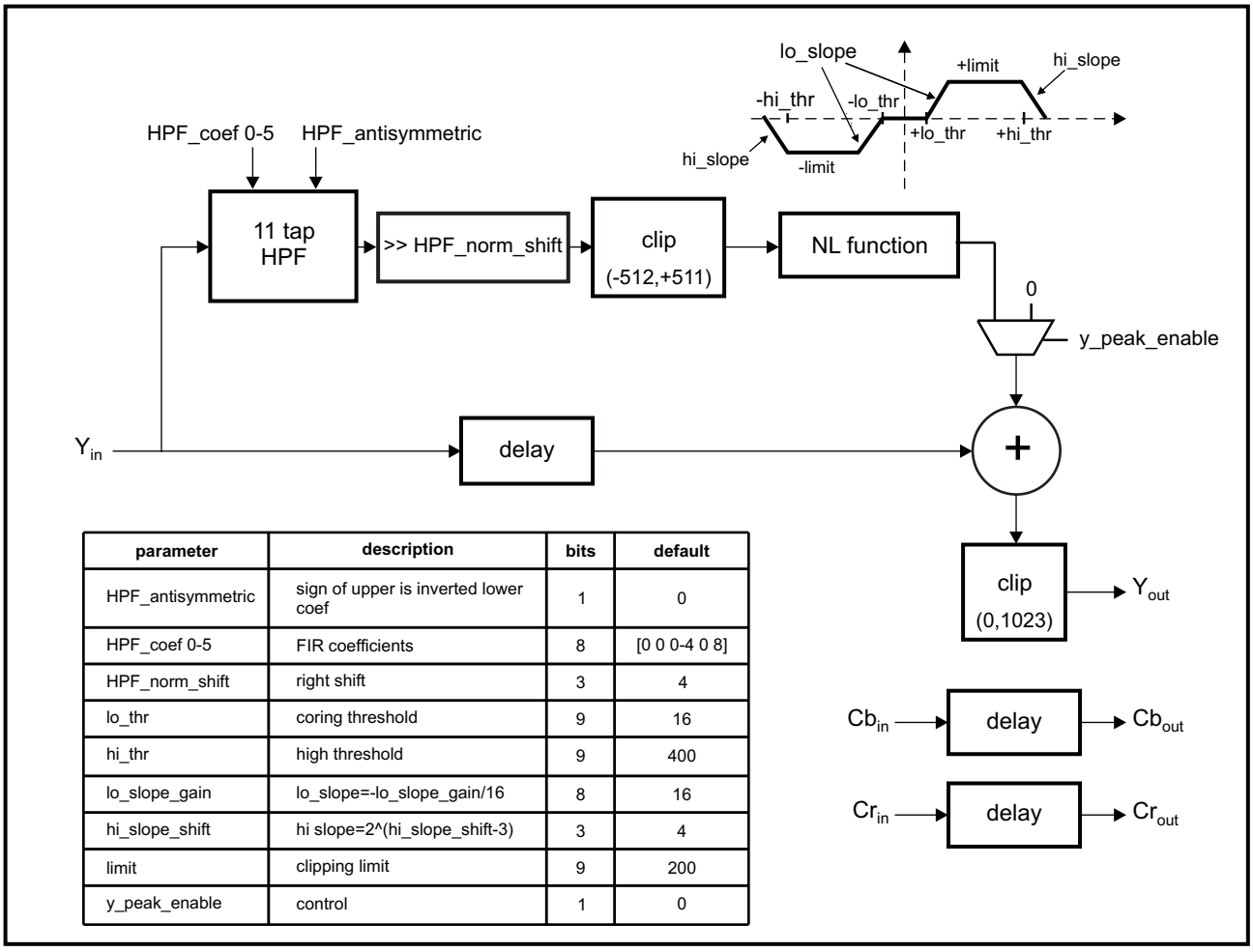


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.2.2 Peaking

The peaking block increases the amplitude of high frequency luminance information in horizontal direction to increase the sharpness of a video image before it is scaled. As shown in Figure 12-142, the high-frequency luminance is increased using an 11-tap High-Pass filter with adjustable gain. The non-linear coring function removes low-level noise and the modified luminance is then added to the original luminance signal. The implementation details are shown in Figure 12-142.

**Figure 12-142. Filter Implementation and Parameter Description**

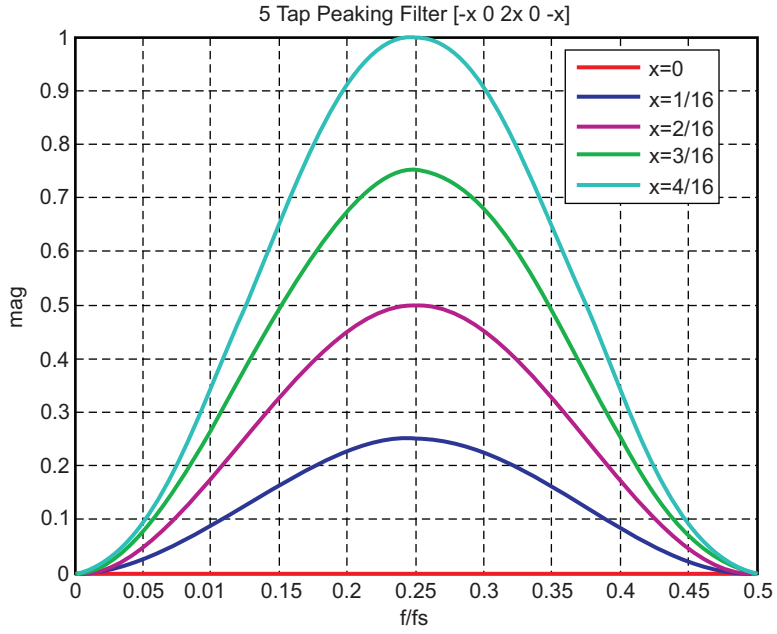



| parameter | description | bits | default |
|---|---|---|---|
| HPF_antisymmetric | sign of upper is inverted lower coef | 1 | 0 |
| HPF_coef 0-5 | FIR coefficients | 8 | [0 0 0-4 0 8] |
| HPF_norm_shift | right shift | 3 | 4 |
| lo_thr | coring threshold | 9 | 16 |
| hi_thr | high threshold | 9 | 400 |
| lo_slope_gain | lo_slope=-lo_slope_gain/16 | 8 | 16 |
| hi_slope_shift | hi slope=2^(hi_slope_shift-3) | 3 | 4 |
| limit | clipping limit | 9 | 200 |
| y_peak_enable | control | 1 | 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

Parameters for the Peaking filters are defined in CFG_SC19-22. The frequency responses of the peaking-filter with different sets of coefficients are shown in Figure 12-143. If the source of the input video is NTSC or PAL format, the peaking filter can be configured to reject the color subcarrier frequency.

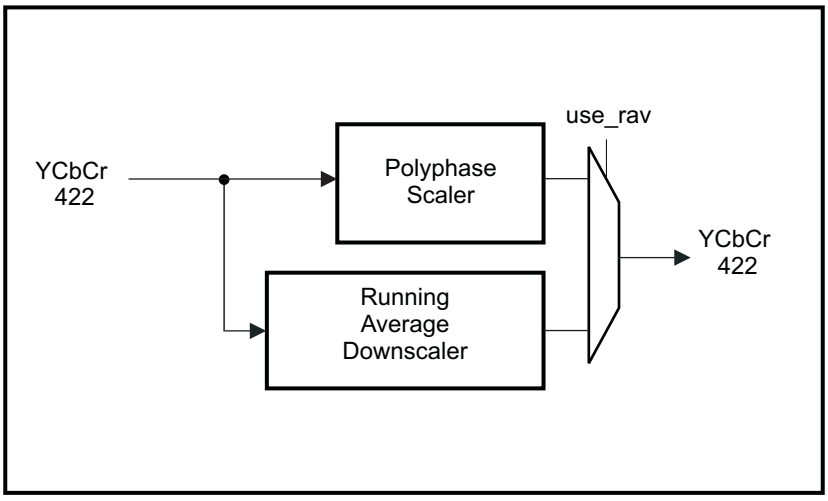
**Figure 12-143. Peaking Filter at fs/4**




### 12.2.11.2.3 Vertical Scaler

The vertical scaler has a poly-phase and a running average filter as shown in Figure 12-144. While the poly-phase filter can be used for any up-scaling and preferably downscaling to 3/16 scale factor, the running average filter is used only for downscaling to a ½ or less size. Selection between these two scalers is based on the user setting of “use_rav” parameter, according to the user preference of the tradeoff between sharpness preserving and introduced artifacts.

**Figure 12-144. Vertical Scaler Block Diagram**



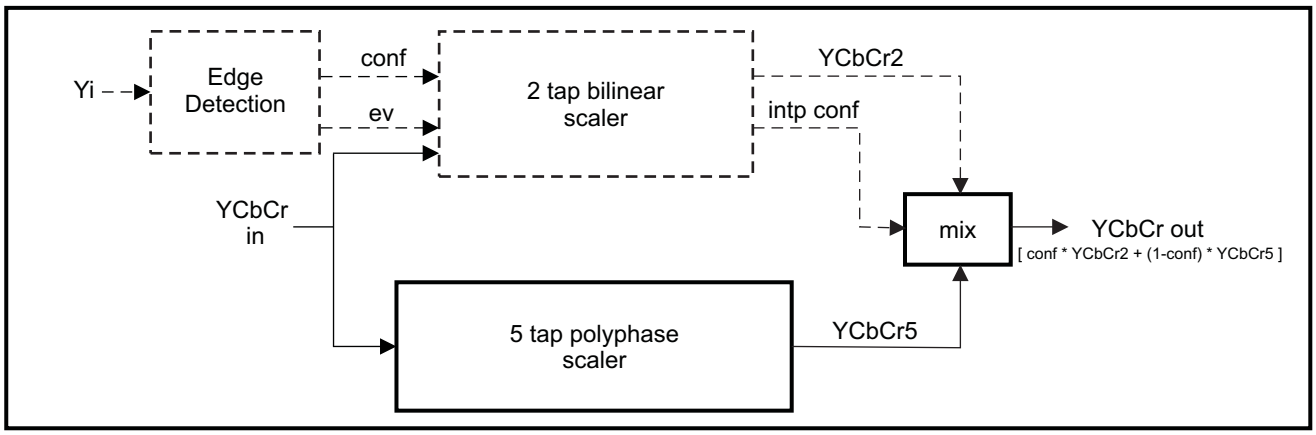


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.2.3.1 Polyphase Scaler (in Vertical Scaler)

The vertical polyphase scaler is mainly implemented using a 32-phase 5-tap polyphase filter.

**Figure 12-145. Mixed 2-tap and 5-tap vertical interpolation**

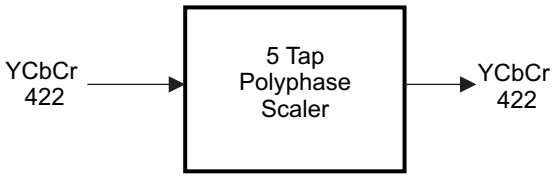



**NOTE:** In SC_M modules present in this device, edge detection and 2-tap bilinear filter modules are not available.

##### 12.2.11.2.3.1.1 Polyphase Filter Selection

The scaler (SC) uses 5-tap polyphase filtering for vertical up-scaling.

**Figure 12-146. Vertical Polyphase Scaler Block Diagram (SC)**




#### 12.2.11.2.3.2 Running Average Filter

When a poly-phase filter is used, usually it has to have many taps in order to achieve acceptable quality for very small downscaling ratio, which requires the use of many line buffers. In HDVPSS, we have a weighted Running Average filter for downscaling when the scaling factor is small (for example, when the scaling ratio is less than 0.5). This highly optimized design requires only one line buffer for luma and one for chroma, which still achieving acceptable quality. The output of the running average filter is based on weighted average of pixels in the current and previous rows in vertical direction. Initializations of accumulators affect the weights.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.2.3.3 Vertical Scaler Configuration Parameters

**Table 12-67. Vertical Scaler Configuration Parameters**

| Parameter | Bits | Typical | Controls | Description |
|---|---|---|---|---|
| interlace_in | 1 | | Frame or Field | 0 = progressive, 1 = interlace |
| interlace_out | 1 | | | 0 = progressive 1 = interlace |
| invt_fid | 1 | | | Invert field ID input |
| use_rav | 1 | | Scaler Mode | 0 = use polyphase scaler<br>1 = use running average scaler |
| row_acc_inc | 11.16 | | Bilinear & Polyphase Scalers | round(2^16*(srcH)/(tarH)) |
| row_acc_offset | 11.16 | 0 | | Initial row accumulator value for progressive frame and top field |
| row_acc_offset_b | 11.16 | 0 | | Initial row accumulator value for bottom field |
| ev_thr | 3.2 | 0xC | Bilinear Scaler | Edge vector threshold used in luma soft switch |
| delta_ev_thr | 4 | 1 | | Range of luma soft switch based on edge vector |
| delta_luma_thr | 4 | 4 | | Range for luma soft switch based on pixel differences (max limit = 8) |
| delta_chroma_thr | 4 | 4 | | Range for chroma soft switch based on pixel differences (max limit = 8) |
| chroma_intp_thr | 10 | 64 | | Threshold used in chroma soft switch based on pixel differences |
| luma polyphase filter coefficients | s1.11 | | | 32 phases × 7 taps × 13 bits array (interpolation only: scale factor = 1) |
| chroma polyphase filter coefficients | s1.11 | | | 32 phases × 7 taps × 13 bits array (interpolation only: scale factor = 1) |
| sc_factor_rav | 10 | | Running Average Scaler | Scale factor = round(1024 × tarH/srcH) |
| row_acc_init_rav | 10 | | | Initial row accumulator value for progressive frame and top field |
| row_acc_init_rav_b | 10 | | | Initial row accumulator value for bottom field |
| luma polyphase filter coefficients | s1.11 | | Polyphase Scaler | 32 phases × 5 taps × 13 bits array (dependent on scale factor) |
| chroma polyphase filter coefficients | s1.11 | | | 32 phases × 5 taps × 13 bits array (dependent on scale factor) |

**Note:** Bi-linear scaler is not present in this device.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.2.4 Horizontal Scaler

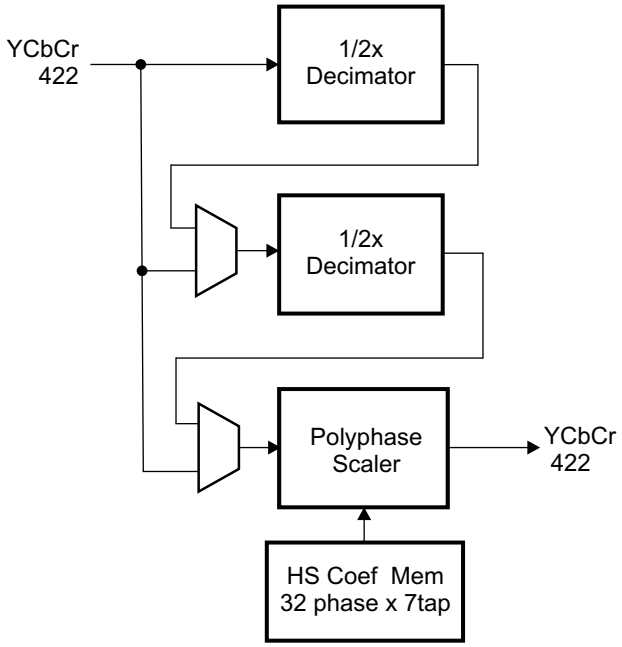
The horizontal scaler is implemented using a 32-phase × 7-tap polyphase filter preceded by two sets of x1/2 decimators. The general configuration of the Horizontal scaler is performed as follows:

- For up scaling, the input video is interpolated using the polyphase scaler.
- For downscaling, it is recommended that input video is decimated by 2 until the modified scale factor falls between 1/2 and 1. Then, a polyphase filter is configured with coefficients selected based on the mod_scale_factor calculated as shown below from one of nine different sets of coefficients: (8,9,10,11,12,13,14,15,16)/16.

```
if (scale_factor>=1/2) {
    dcm_2x=0;
    dcm_4x=0;
    mod_scale_factor=scale factor;
} else if (scale_factor>=1/4) {
    dcm_2x=1;
    dcm_4x=0;
    mod_scale_factor=2*scale_factor;
} else {
    dcm_2x=0;
    dcm_4x=1;
    mod_scale_factor=4*scale_factor;
}
```

In auto mode (auto_hs == 1), scaler will operate as per above recommendation. In addition to this, for (auto_hs==1), polyphase filtering will be bypassed when (scale_factor == 1) or (scale_factor == ½) or (scale_factor == ¼). If (auto_hs==0) is used, user must provide proper values for dcm_2x, dcm_4x, proper values for all inputs to polyphase filter in the registers as well as appropriate coefficient sets. There is no constraint on polyphase filtering in terms of scaling ratio. However, there are constraints on input and output image width for scaler. Frame dimensions are limited from 64x64 to 2047x2047.

**Figure 12-147. Horizontal Scaler Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.2.4.1 Half Decimation Filter

The half-decimation filter is an 11-tap filter with following coefficients – (14, 0, -29, 0, 79, 128, 79, 0, -29, 0, 14). For processing left and right edge pixels, the first and last data are repeated to pre-fill and extend the filter data pipeline respectively.

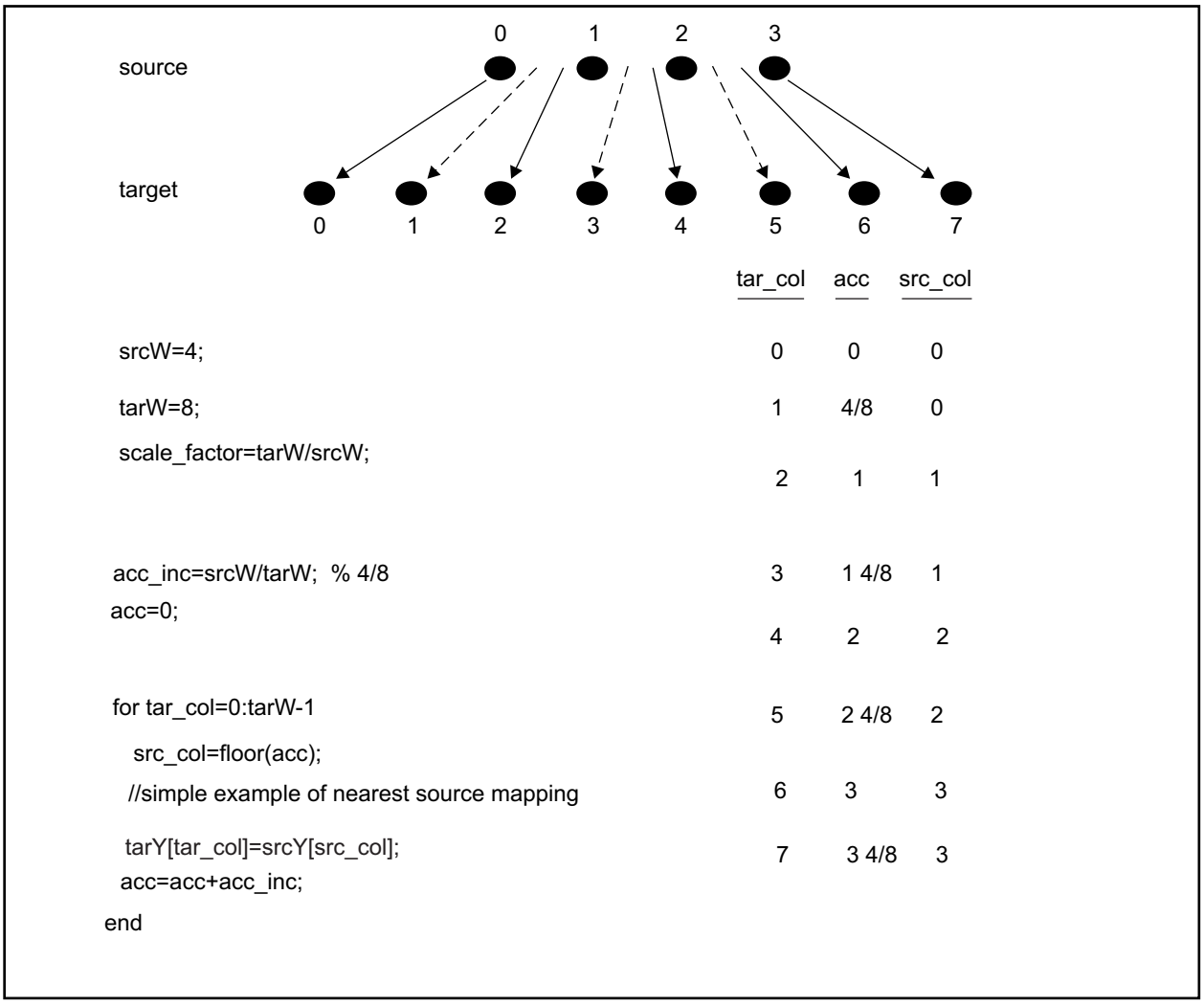
These coefficients are hard-coded into scaler design and user cannot modify these.

#### 12.2.11.2.4.2 Polyphase Filter

The horizontal scaling is done by stepping across the target row and interpolating target pixels based on source pixels. Accumulator points to the source pixel that corresponds to the target pixel.

Figure 12-148 shows an up-scaling example.

**Figure 12-148. Polyphase Filtering Example**



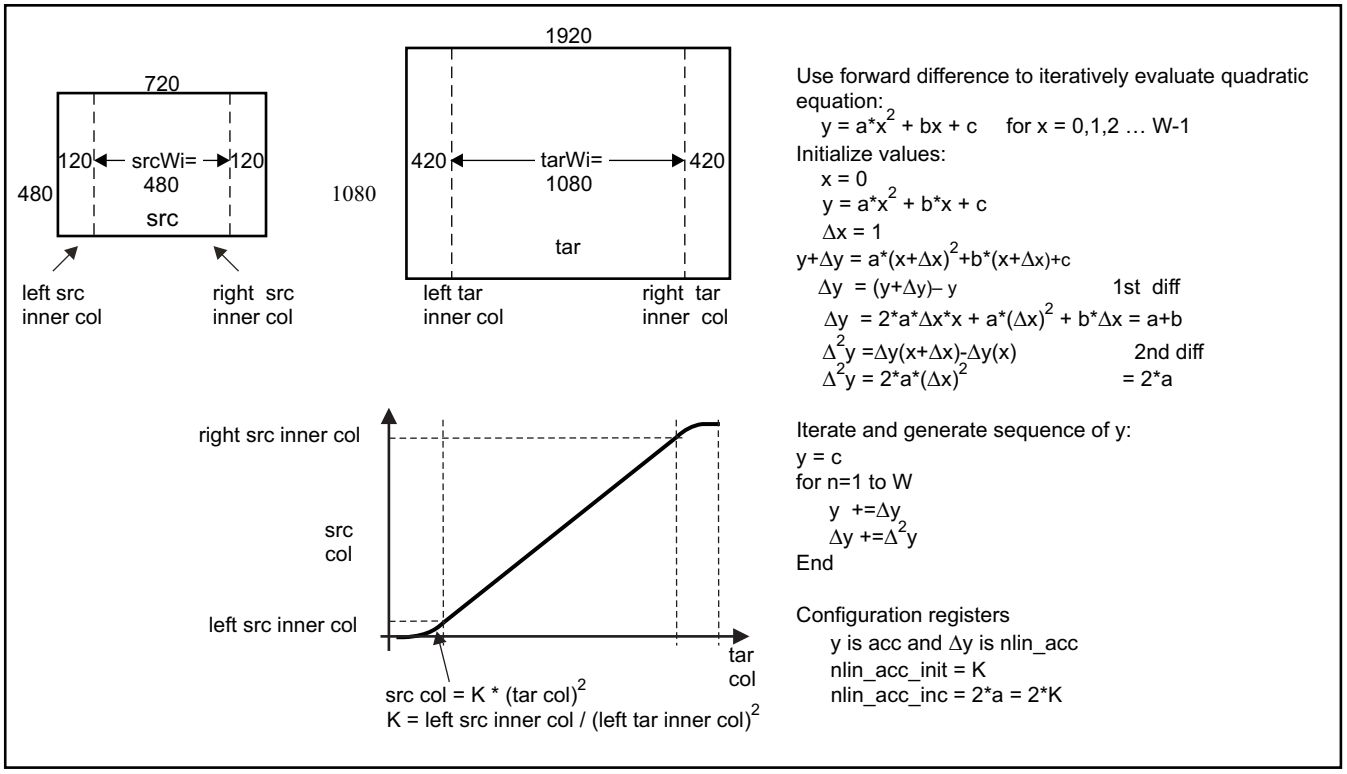


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.11.2.4.3 Nonlinear Horizontal Scaling

The horizontal scaler supports non-linear scaling to maintain the same aspect ratio for inner picture while stretching picture edges to fill the screen when displaying a 4x3 picture onto a 16x9 display. When setting up a non-linear scaling application, the inner picture is defined as the center square portion of the image with width and height equal to the height of the source image. Remaining side regions are then scaled non-linearly. Figure 12-149 shows a non-linear scaling case:

**Figure 12-149. Non-linear Scaling Example**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.2.4.4 Horizontal Scaler Configuration Registers

**Table 12-68. Horizontal Scaler Related Parameter Configuration**

| Parameter | Bits | Controls | Description | | | |
|---|---|---|---|---|---|---|
| srcW | 11 | Image Dimension | Source Width | | | |
| tarW | 11 | | Target Width | | | |
| linear | 1 | Scaler Mode | If (linear == 1) srcWi=srcW and tarWi=tarW<br>Else srcWi=srcH and tarWi=tarH | | | |
| sc_bypass | 1 | | 0 = enable scaler, 1 = bypass scaler | | | |
| auto_hs | 1 | | Auto_hs | dcm_2x | dcm_4x | Definition |
| | | | 0 | 0 | 0 | Polyphase scaling |
| dcm_2x | 1 | | 0 | 0 | 1 | Horizontal decimation by 4 and polyphase scaling |
| | | | 0 | 1 | 0 | Horizontal decimation by 4 and polyphase scaling |
| dcm_4x | | | 1 | - | - | Automatic (selection of decimation filter is automatic) |

**Table 12-69. Horizontal Decimation Selection**

| Scale Factor | Decimation |
|---|---|
| < ¼ | By 4 |
| < 1/4 | By 4 |
| == 1/4 | By 4 |
| 1/4 < and < 1/2 | By 2 |
| 1/2 | By 2 |
| 1/2 < and < 1 | Bypassed |
| 1 | Bypassed |
| > 1 | Bypassed |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-70. Horizontal Polyphase Filter Related Parameter Configuration**

| Parameter | Bits | Controls | Description |
|---|---|---|---|
| lin_acc_inc | 11.24 | Polyphase Scaler | if upscaling then lin_acc inc = round(2^24*(srcWi)/(tarWi))<br>elseif downscaling lin_acc inc = round(2^24*(srcWi/n)/(tarWi))<br>where<br>n=1 when (cfg_dcm_2x == 0 && cfg_dcm_4x == 0)<br>n=2 when (cfg_dcm_2x == 1 && cfg_dcm_4x == 0)<br>n=4 when (cfg_dcm_2x == 0 && cfg_dcm_4x == 1) |
| nlin_left (Ltar) | 11 | | if linear==1 nlin_right = tarW-1<br>else nlin_right = Ltar + tarWi – 1 |
| nlin_right (Rtar) | 11 | | if linear==1 nlin_left = 0 |
| nlin_acc_inc | 11.24 | | if tarW/srcW>=1 then<br>d = 0<br>if Ltar!=0 K = round[2^24*Lsrc/(Ltar*Ltar) ] where Lsrc= (srcW-srcWi)/2<br>else K = 0<br>else<br>d = (tarW-1)/2<br>if Ltar!=0 K = round[ 2^24 * Lsrc / (Ltar*(Ltar-2d))]<br>where Lsrc= (srcW-srcWi)/(2n) and n=1,2 or 4<br>else K = 0<br>nlin_acc_inc = 2*K (negative for downscaling) |
| nlin_acc_int | 11.24 | | nlin_acc_init = K*(1-2*d) |
| luma polyphase filter coefficients | s1.11 | | 32 phases × 7 taps × 13 bits array |
| chroma polyphase coefficients | s1.11 | | 32 phases × 7 taps × 13 bits array |

**Note:** Source width and height variables for the polyphase filter are internally set with trimmer and decimation filter adjusted values.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.2.5 Coefficient Memory

##### 12.2.11.2.5.1 Overview

The scaler requires initialization of eight coefficient SRAMS prior to processing a video frame. These coefficients are stored in the external DDR memories and downloaded by the VPDMA. The VPDMA writes the coefficient data through the VPI Control Interface bus.

The eight coefficient SRAMs are:

- HS horizontal polyphase scaler, Luma and Chroma (7tap)
- VS vertical polyphase scaler, Luma and Chroma (5tap or 3tap)
- VS vertical bilinear scaler, top line Luma and Chroma (7tap)
- VS vertical bilinear scaler, bottom line Luma and Chroma (7tap)

Only the SC_H requires the VS vertical bilinear scaler coefficients and has the memories to store them. Nevertheless, SC will accept the VPI Write command to these memories although it does not store the data.

##### 12.2.11.2.5.2 Physical Coefficient SRAM Layout

Each of the six legacy coefficient SRAMs is 32 phases × 91 bits. A single coefficient value is 13 bits. Therefore, one word of the SRAM contains 7 coefficient values as shown in Figure 12-150, and 224 coefficient values are stored in each SRAM.

**Figure 12-150. SRAM Layout for 7tap Coefficient**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Phase 0 | C6 | C5 | C4 | C3 | C2 | C1 | C0 |
| | | | | | | | |
| Phase 31 | C223 | C222 | C221 | C220 | C219 | C218 | C217 |

The two vertical polyphase SRAMs are 32 phases × 65 bits. A single coefficient is 13 bits. Therefore, one word of SRAM contains 5 coefficient values as shown in Figure 12-151.

**Figure 12-151. SRAM Layout for 5tap Coefficient**

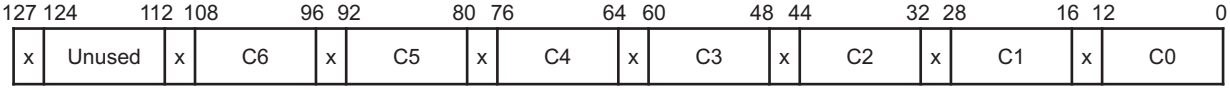
| | | | | | |
|---|---|---|---|---|---|
| Phase 0 | C4 | C3 | C2 | C1 | C0 |
| | | | | | |
| Phase 31 | C221 | C220 | C219 | C218 | C217 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.2.5.3 Scaler Coefficients Packing on 128-bit VPI Control I/F

A coefficient value is 13 bits wide and takes a single half word. Thus, one VPI Control write carries one phase's worth of coefficients. The last half-word in a quad-word is not used for 7tap coefficients.

**Figure 12-152. VPI Control I/F Coef Data Format (7tap)**




The Vertical Polyphase scaler coefficients have only five or three values per phase, so the last three (or five) entries are unused. The Vertical Polyphase coefficient packing is shown in Figure 12-153 and Figure 12-154.

**Figure 12-153. VPI Control I/F Coef Data Format (5tap)**

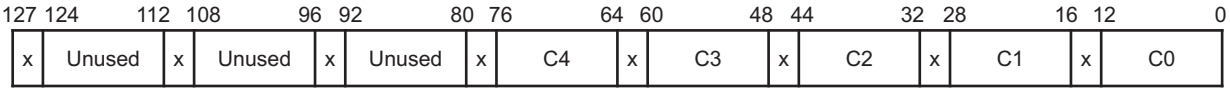
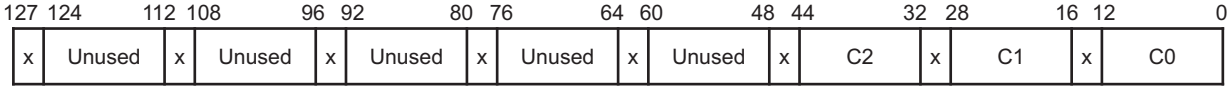


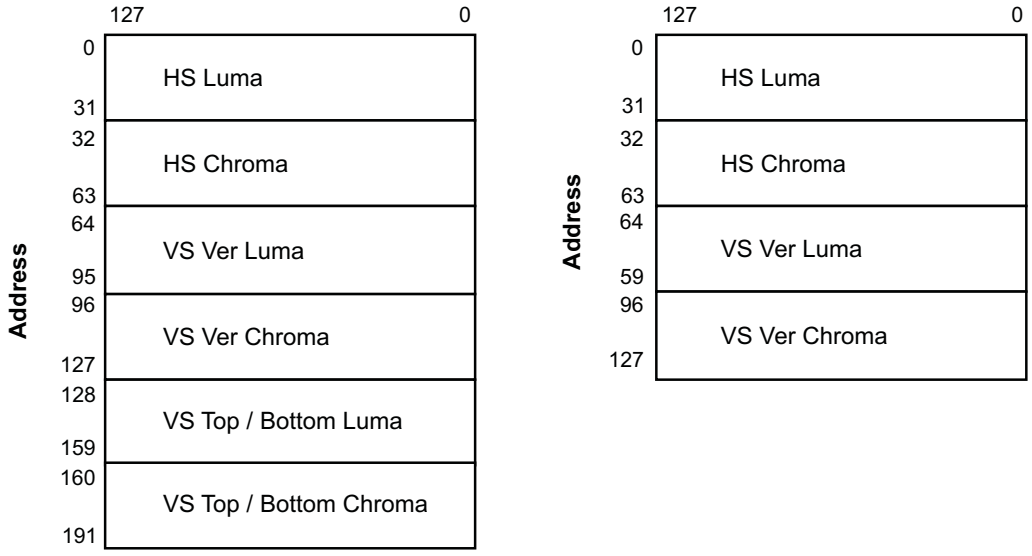

**Figure 12-154. VPI Control I/F Coef Data Format (3tap)**




#### 12.2.11.2.5.4 VPI Control I/F Memory Map for Scaler Coefficients

The memory map of the VPI Control I/F for the Scaler coefficients is shown in Figure 12-155. All coefficients can be filled up by a single VPI Control write command. The update address feature of VPI Control I/F enables individual access to a certain location of memories. The VS top and bottom coefficients of SC_H are written concurrently through VPI Control Write.

**Figure 12-155. VPI Control I/F Memory Map (Write)**

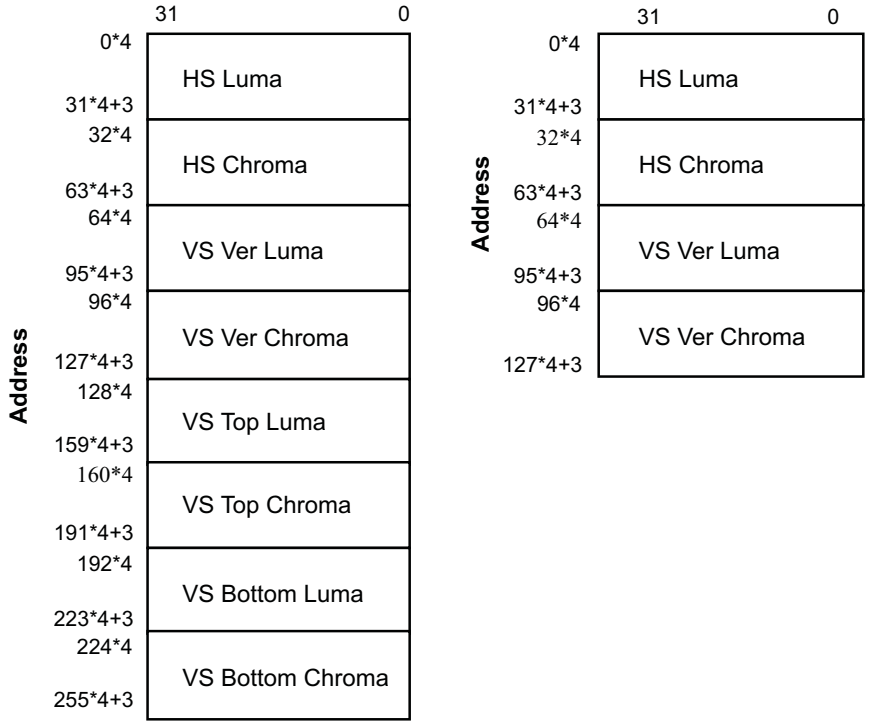



**(a) sc_h** **(b) sc_m**


Copyright © 2011–2016, Texas Instruments Incorporated

The module supports the VPI Control Read to read back the contents in the coefficient memories for debug purpose. Figure 12-156 shows the memory map of the VPI Control Read. As the VPI Control Read has only 32 bit data bus, it requires the word addressing while the write does the quad-word addressing. The VS top and bottom coefficients can be accessed separately for the read.

**Figure 12-156. VPI Control I/F Memory Map (Read)**




### 12.2.11.2.5.5 VPI Control Interface

VPDMA is used to configure the coefficient memories of scaler through VPI control interface. Since the coefficient memories are not shadowed (unlike the MMR registers), VPI control write access needs to be done only during the gap between video frame processing times. If a write request is made while the Scaler is active, the access will be held off by the hardware until the last data of the currently processed frame is sent out. Care must be given to the order of the DMA descriptors so that blocking of VPI control bus does not occur.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.2.5.6 Coefficient Table Selection Guide

The scaler filter coefficient tables are pre-generated using a Matlab program for various scaling factor ranges. Table 12-71 is a general selection guide table for coefficient data files.

**Table 12-71. Coefficient Data Files**

| Scaler | Scale Factor | Coeff table |
|---|---|---|
| HS Polyphase Filter | All upscaling | ppfcoef_scale_eq_1_32_phases_flip.dat |
| HS Polyphase Filter | All upscaling | ppfcoef_scale_eq_1_32_hases_flip.dat |
| | ½ or ¼ down scaling | ppfcoef_scale_eq_1_32_phases_flip.dat |
| | > 15/16 | ppfcoef_scale_eq_15div16_32_phases_flip.dat |
| | > 14/16 | ppfcoef_scale_eq_14div16_32_phases_flip.dat |
| | > 13/16 | ppfcoef_scale_eq_13div16_32_phases_flip.dat |
| | > 12/16 | ppfcoef_scale_eq_12div16_32_phases_flip.dat |
| | > 11/16 | ppfcoef_scale_eq_11div16_32_phases_flip.dat |
| | > 10/16 | ppfcoef_scale_eq_10div16_32_phases_flip.dat |
| | > 9/16 | ppfcoef_scale_eq_9div16_32_phases_flip.dat |
| | > 8/16 | ppfcoef_scale_eq_8div16_32_phases_flip.dat(1) |
| VS Polyphase Filter | Upscaling | ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat |
| | > 15/16 | ppfcoef_scale_eq_15div16_32_phases_ver_5tap_flip.dat |
| | > 14/16 | ppfcoef_scale_eq_14div16_32_phases_ver_5tap_flip.dat |
| | > 13/16 | ppfcoef_scale_eq_13div16_32_phases_ver_5tap_flip.dat |
| | > 12/16 | ppfcoef_scale_eq_12div16_32_phases_ver_5tap_flip.dat |
| | > 11/16 | ppfcoef_scale_eq_11div16_32_phases_ver_5tap_flip.dat |
| | > 10/16 | ppfcoef_scale_eq_10div16_32_phases_ver_5tap_flip.dat |
| | > 9/16 | ppfcoef_scale_eq_9div16_32_phases_ver_5tap_flip.dat |
| | > 8/16 | ppfcoef_scale_eq_8div16_32_phases_ver_5tap_flip.dat |
| | else | For down-scaling <8/16, RAV filter is recommended. In this case, coefficients for vertical scaling need not be loaded. |

(1) HS Scaler has two sets of ½-decimator to perform downscaling ratios below ½ and ¼.

The above mentioned .dat files are available in Section 12.2.11.4.

### 12.2.11.3 Code

#### 12.2.11.3.1 Generate Coefficient Memory Image

The following Perl script is used to generate a coefficient memory image for a given set of scaling factors.

```
#!/usr/local/bin/perl

#$dir="coef/";                    # directory which contains the coef files
#$cfg_file="sc_config1.cfg";      # configuration file name
#$spl_file="sc_config_supl.cfg";  # supplemental configuration file name $cfg_file=$ARGV[0];  #
configuration file name $spl_file=$ARGV[1];  # supplemental configuration file name
$dir=$ARGV[2];       # directory which contains the coef files

$coef_width=13;  # coef bit width
$coef_ntap=7;    # coef tap
$coef_nphase=32; # coef phase
$coef_norm=11;   # coef norm

#-----------------------------------------------------------
# read config file to get srcH/tarH/interlace_i/interlace_o
#-----------------------------------------------------------
open(INFILE, "<$cfg_file") or die "### ERROR: Cannot open $cfg_file";
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
while(<INFILE>){
  if (m/([-0-9]+) +\/\/ +srcW/) {
    $srcW = $1;
  } elsif (m/([-0-9]+) +\/\/ +srcH/) {
    $srcH = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarW/) {
    $tarW = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarH/) {
    $tarH = $1;
  } elsif (m/([-0-9]+) +\/\/ +interlace_in/) {
    $interlace_i = $1;
  } elsif (m/([-0-9]+) +\/\/ +interlace_out/) {
    $interlace_o = $1;
}
}
close(INFILE);

#-------------------------------------------------------------
# read supplemental config file to get srcWi/tarWi from
#-------------------------------------------------------------
open(INFILE,"<$spl_file") or die "### ERROR: Cannot open $spl_file";
while(<INFILE>){
  if (m/([-0-9]+) +\/\/ +srcWi/) {
    $srcWi = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarWi/) {
    $tarWi = $1;
  } elsif (m/([-0-9]+) +\/\/ +profile/) {
    $profile = $1; # 0:HIGH,1:MEDIUM,2:LOW
  }
}
close(INFILE);
#-------------------------------------------------------------
# determine coef file based on the width/height
#-------------------------------------------------------------
# VS
#$vsc_file0 = "mod_ppfcoef_scale_eq_1_32_phases_flip.dat";
$vsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip_PPF3_peak5_gain_eq_1_25.dat";
# VS VER
$mod_tarH = ($interlace_i == 0 && $interlace_o == 1) ? $tarH<<1 : $tarH; if ($profile==2) {
  # LOW profile
  if ($mod_tarH >= $srcH) {
    $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_3tap_flip.dat";
} else {
    if ($mod_tarH >=($srcH>>1)) {
      $n = int(16.0*$mod_tarH/$srcH);
      $vsc_ver_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_ver_3tap_flip.dat",$n);
    } else {
      $n = 0;
      $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_3tap_flip.dat";
    }
  }
} else {
  if ($mod_tarH >= $srcH) {
    $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat";
  } else {
      $n = int(16.0*$mod_tarH/$srcH);
      $vsc_ver_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_ver_5tap_flip.dat",$n);
  }
}

# HS
if ($tarWi >= $srcWi) {
  $hsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip.dat";
} elsif ( ($tarWi == ($srcWi>>1)) || ($tarWi == ($srcWi>>2)) ) {
  $hsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip.dat";
} else {
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
if ($tarWi > ($srcWi>>1)) {
    $n = int(16.0*$tarWi/$srcWi);
  } elsif ($tarWi > ($srcWi>>2)) {
    $n = int(16.0*$tarWi/($srcWi>>1));
  } elsif ($tarWi >=($srcWi>>3)) {
    $n = int(16.0*$tarWi/($srcWi>>2));
  } else {
    $n = 0;
  }
  $hsc_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_flip.dat",$n);
}
#-------------------------------------------------------------
# write out the coef hex file
#-------------------------------------------------------------
&write_coef($hsc_file0);
&write_coef($hsc_file0);
&write_coef($vsc_ver_file0);
&write_coef($vsc_ver_file0);
&write_coef($vsc_file0);
&write_coef($vsc_file0);
sub write_coef {

  my ($filename) = @_;

  open(INFILE, "<$dir/$filename") or die "### ERROR: Cannot open $dir/$filename";

  $line=<INFILE>;
@val=split(' ',$line);
  $ntap=$val[0];
  $nphase=$val[1];
  $norm=$val[2];
for ($p=0;$p<$nphase;$p++) {
    $line=<INFILE>;@val=split(' ',$line);
    for($i=0;$i<$ntap;$i++) {
      if ($val[$i]<0) {
        $val[$i]+=(1<<$coef_width);
      }
    }
    undef(@coef);
    unshift(@coef, sprintf("%04x",$val[0]));
    unshift(@coef, sprintf("%04x",$val[1]));
    unshift(@coef, sprintf("%04x",$val[2]));
    unshift(@coef, sprintf("%04x",$val[3]));
    unshift(@coef, sprintf("%04x",$val[4]));
    unshift(@coef, sprintf("%04x",$val[5]));
    unshift(@coef, sprintf("%04x",$val[6]));
    unshift(@coef, sprintf("%04x",0));
    $coef=join("",@coef);
    print "$coef\n";
  }

  close(INFILE);
}
```


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.3.2 Scaler Configuration Calculation

The following C-code shows how configuration parameters are calculated:

```
// ======================================
// Required Input Parameter
// ======================================
//   srcW, srcH, tarW, tarH, srcWi, tarWi
//   input/output scan modes


pixel_scale_factor=4;  // 10 bit pixel
hor_pixel_offset =0.0

  // ================================================================================================
  // Peaking Filter Configuration
  // ================================================================================================
  // ----------------------------------------------------------------
  // HPF Coef
  // ----------------------------------------------------------------
    y_peak_enable     =   0;

    peak_select=0; // 0=peak at fs/4  1=NTSC  2=PAL
    switch(peak_select) {
    case 0: {// peak at fs/4 and gain = 1
             HPF_coef0         =   0;
             HPF_coef1         =   0;
             HPF_coef2         =   0;
             HPF_coef3         =  -4;
             HPF_coef4         =   0;
             HPF_coef5         =   8; // mid tap
             HPF_norm_shift    =   4;
         break;
}
    case 1: {// NTSC: peak at 0.133*fs and gain=1
             HPF_coef0         =  -2;
             HPF_coef1         =  -8;
             HPF_coef2         =  -8;
             HPF_coef3         =  -2;
             HPF_coef4         =  12;
             HPF_coef5         =  16; // mid tap
             HPF_norm_shift    =   6;
             break;
    }
    case 2: {// PAL: peak at 0.163*fs and gain=1
             HPF_coef0         =   2;
             HPF_coef1         =  -4;
             HPF_coef2         = -11;
             HPF_coef3         =  -7;
             HPF_coef4         =   9;
             HPF_coef5         =  22; // mid tap
             HPF_norm_shift    =   6;
             break;
    }
    }

  // ----------------------------------------------------------------
  // NonLinear Coring Function typical values
  // ----------------------------------------------------------------
    NL_coring_thr     =  16;
    NL_limit          = 200;
    NL_lo_slope       =  16;
    NL_hi_thr         = 400;
    NL_hi_slope_shift =   4;
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  // ================================================================================================
  // Edge Detection Configuration
  // ================================================================================================
    // edge detection
    confidence_default = 0; // 0  =use 5 tap polyphase filter for SC or sc_h with ev_enable =0

    min_Gy_thr         =  64; // 64
    min_Gy_thr_range   =   3; // 3 power of 2
    gradient_thr       = 200; // 200
gradient_thr_range =   6; // 6 power of 2

    ev_thr = int(4.0*3.111+0.5); // edge vector soft switch threshold (3.2)


// ============================================================================
// vertical scaler configuration
// ============================================================================
//srcH/tarH is field height for interlaced data //srcH/tarH is frame height for progressive data

// ----------------------------------------------------------------
// vertical scaler typical parameters
// ----------------------------------------------------------------
invert_field_ID  = 0;   // invert field ID input
ver_pixel_offset = 0.0; // User may modify this is required
uv_intp_thr      = pixel_scale_factor*16;
delta_y_thr      = 4;   // luma soft switch range
delta_uv_thr     = 4;   // chroma soft switch range

// ----------------------------------------------------------------
// Vertical Scaler Mode Determination
// ----------------------------------------------------------------
if((interlace_in == 0) && (interlace_out == 1)) {
    if(2*tarH > (srcH/2))
        use_rav = 0;  // 1=use RAV scaler 0=use polyphase scaler
    else
        use_rav = 1;
}
else
{
    if(tarH > (srcH/2))
        use_rav = 0;  // 1=use RAV scaler 0=use polyphase scaler
    else
        use_rav = 1;
}

// ----------------------------------------------------------------
// RAV or Polyphase parameters
// ----------------------------------------------------------------
if(use_rav) // downscale only
{
    // ------------
    // --- RAV ----
    // ------------
    if(use_internal_defaults)
        enable_edge_detection = 0;

    if((interlace_in == 0) && (interlace_out == 1))
    {
        scale = double(2*tarH)/double(srcH);
    }
    else
    {
        scale = double(tarH)/double(srcH);
    }
    sc_factor_rav = int(1024.0*scale+0.5);
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
    // Peter's method
    delta = (1.0/scale-1.0)/2.0;
    int_part         = floor(delta);
    frac_part        = delta-int_part;
    row_acc_init_rav = int(1024*(scale+(1.0-scale)/2.0)+0.5); // top field

    // bottom field
    tmp1 = (1.0-2.0*frac_part);
    tmp2 = (1.0- (1.0+2.0*int_part)*scale);
    row_acc_init_b_rav = int(1024*(scale + tmp1*tmp2/2.0) + 0.5);

    row_acc_inc      = 0; // polyphasescaler
    row_acc_offset   = 0; // polyphasescaler
    row_acc_offset_b = 0; // polyphasescaler } else {
    //upscale using polyphase scaler
    // ------------
    // --- PPF ----
    // ------------
    if(use_internal_defaults)
        enable_edge_detection = 1;
    sc_factor_rav      = 0;
    delta_rav          = 0;
    row_acc_init_rav   = 0;
    row_acc_init_b_rav = 0;

    // progressive or top field
    row_acc_offset = int(65536.0*ver_pixel_offset +0.5);
    row_acc_inc    = int(65536.0*double(srcH)/double(tarH)+0.5);

    //bottom field
    if((interlace_in == 1) ampamp (interlace_out == 0))
    {
        if((-0.5+row_acc_offset) < 0.0)
            round_factor = -0.5;
        else
            round_factor = 0.5;
        row_acc_offset_b = int(65536.0*(-0.5)+round_factor)+row_acc_offset;
    }
    if((interlace_in == 0) && (interlace_out == 1))
    {
        row_acc_offset_b =
            int(65536.0*double(srcH)/double(tarH)+0.5)+row_acc_offset;
    }
    if((interlace_in == 1) ampamp (interlace_out == 1))
    {
        row_acc_offset_b =
            (int(65536.0/2.0*(double(srcH)/(double(tarH))-1.0)+0.5))+
            row_acc_offset;
    }
}


// ================================================================================================
  // Horizontal Scaler configuration
  // ================================================================================================
  // ----------------------------------------------------------------
  // horizontal scaler mode determination
  // ----------------------------------------------------------------
  auto_hs                = 1;
  dcm_2x                 = 0;
  dcm_4x                 = 0;
  hp_bypass              = 0;
  if (srcWi==srcW) linear = 1;
  else             linear = 0;

  // hor scaler parameters
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  if (tarW>srcW) {              // upscale
    mod_srcW  = srcW;
    mod_srcWi = srcWi;
} else if (tarW<=(srcW>>2)) {  // downscale by <=1/4
    mod_srcW  = srcW>>2;
    mod_srcWi = srcWi>>2;
  } else if (tarW<=(srcW>>1)) {  // downscale by <=1/2
    mod_srcW  = srcW>>1;
    mod_srcWi = srcWi>>1;
  } else {                      // downscale by <=1
    mod_srcW  = srcW;
    mod_srcWi = srcWi;
  }

  // Not used any more:
  // hs_factor       = int(16.0*double(tarWi)/double(mod_srcWi)+0.5);  // hor scale factor (6.4)

  // -----------------------------------------------------------------
  // Horizontal PolyPhase Settings  --
  // -----------------------------------------------------------------
  lin_acc_inc     = int(16777216.0*double(mod_srcWi-1)/double(tarWi-1)+0.5);
  col_acc_offset  = int(16777216.0*hor_pixel_offset +0.5);
  nlin_left       = (tarW-tarWi)>>1;
  nlin_right      = nlin_left+tarWi-1;
  if (linear) {
    nlin_acc_inc    = 0;
    nlin_acc_init   = 0;
  } else {
     // ---------------------------------
     // Non-linear scaling configuration
     // ---------------------------------
    nlin_left_src   = (mod_srcW-mod_srcWi)>>1;

       if (tarWi>=srcWi) {    // upscale
      d             =  0.0;
      round_factor  =  0.5;
       } else {                 // downscale
      d             = (double(tarW)-1.0)/2.0;
      round_factor  = -0.5;
    }

      K             = 16777216.0*double(nlin_left_src)/(double(nlin_left)*double(nlin_left-
2.0*d));
    nlin_acc_inc  = int(2.0*K+round_factor);
    nlin_acc_init = int(K*(1.0-2.0*d)+0.5);
  }
nlin_left_tar  = nlin_left;
  nlin_right_tar = nlin_right;

  // ================================================================================================
  // Bypass Determination
  // ================================================================================================
  // bypass
  if ((srcW==tarW)&&(srcWi==tarWi)&&(mod_srcH==mod_tarH)) sc_bypass = 1;
  else                                                   sc_bypass = 0;
  //

}
```


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.3.3 Typical Configuration Values

The following is the list of all scaler register fields that are set to constant values by the software – representing typical settings:

CFG_SC0 : InvertFieldId = 0
CFG_SC0 : EnableEdgeDetection = 1
CFG_SC0 : AutoHorizontalScaling = 1
CFG_SC0 : Decimation2x = 0
CFG_SC0 : Decimation4x = 0
CFG_SC0 : EnableSin2VerticalInterpolation = 1
CFG_SC0 : EnableLumaPeaking = 0
CFG_SC0 : EnableTrimming= 1
CFG_SC12 : ColumnAccumulatorOffset = 0
CFG_SC13 : ChromaInterpolationThreshold = 64
CFG_SC13 : DeltaChromaThreshold = 4
CFG_SC17 : EdgeVectorThreshold = 12 (3.111 in 8.2 format)
CFG_SC17 : DeltaLumaThreshold = 4
CFG_SC17 : DeltaEdgeVectorThreshold = 1
CFG_SC18 : ConfidenceFactorDefault = 0x100
CFG_SC19 = 0xFC000000
CFG_SC20 = 0x0C840800
CFG_SC21 = 0x00100010
CFG_SC22 = 0x00040190

### 12.2.11.4 Coefficient Data Files

#### 12.2.11.4.1 HS Polyphase Filter Coefficients

##### 12.2.11.4.1.1 ppfcoef_scale_eq_1_32_phases_flip.dat

```
 7  32 11
31 -112  210 1790  210 -112   31
28  -98  159 1787  264 -126   34
25  -84  111 1779  320 -140   37
22  -71   65 1767  379 -154   40
19  -58   23 1750  439 -168   43
16  -45  -17 1728  502 -181   45
14  -33  -53 1701  565 -193   47
11  -22  -86 1670  631 -205   49
 9  -11 -116 1635  696 -216   51
 7   -1 -142 1594  763 -225   52
 5    8 -166 1551  830 -233   53
 3   16 -186 1504  898 -240   53
 2   23 -204 1455  965 -245   52
 1   30 -218 1401 1031 -248   51
 0   35 -230 1345 1097 -249   50
-1   40 -238 1286 1162 -248   47
44 -244 1224 1224 -244   44    0
47 -248 1162 1286 -238   40   -1
50 -249 1097 1345 -230   35    0
51 -248 1031 1401 -218   30    1
52 -245  965 1455 -204   23    2
53 -240  898 1504 -186   16    3
53 -233  830 1551 -166    8    5
52 -225  763 1594 -142   -1    7
51 -216  696 1635 -116  -11    9
49 -205  631 1670  -86  -22   11
47 -193  565 1701  -53  -33   14
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  45 -181  502 1728  -17  -45   16
  43 -168  439 1750   23  -58   19
  40 -154  379 1767   65  -71   22
  37 -140  320 1779  111  -84   25
  34 -126  264 1787  159  -98   28
```

#### 12.2.11.4.1.2 ppfcoef_scale_eq_8div16_32_phases_flip.dat

```
7  32 11
  -28   61  542  898  542   61  -28
  -27   52  523  899  560   70  -29
  -26   44  505  898  578   79  -30
  -25   37  487  895  595   89  -30
  -24   30  468  892  613  100  -31
  -22   23  450  887  630  111  -31
  -21   17  432  883  647  122  -32
  -20   11  414  877  664  134  -32
  -19    6  396  871  680  146  -32
  -18    1  378  864  695  159  -31
  -16   -4  360  856  711  172  -31
  -15   -8  343  847  726  185  -30
  -14  -12  325  838  740  200  -29
  -13  -15  308  828  754  214  -28
  -12  -18  292  816  768  229  -27
  -10  -21  275  805  780  244  -25
  -23  258  789  789  258  -23    0
  -25  244  780  805  275  -21  -10
  -27  229  768  816  292  -18  -12
  -28  214  754  828  308  -15  -13
  -29  200  740  838  325  -12  -14
  -30  185  726  847  343   -8  -15
  -31  172  711  856  360   -4  -16
  -31  159  695  864  378    1  -18
  -32  146  680  871  396    6  -19
  -32  134  664  877  414   11  -20
  -32  122  647  883  432   17  -21
  -31  111  630  887  450   23  -22
  -31  100  613  892  468   30  -24
  -30   89  595  895  487   37  -25
  -30   79  578  898  505   44  -26
  -29   70  560  899  523   52  -27
```

#### 12.2.11.4.1.3 ppfcoef_scale_eq_9div16_32_phases_flip.dat

```
7  32 11
  -33    8  547 1004  547    8  -33
  -31    0  525 1003  570   16  -35
  -29   -7  503 1001  592   25  -37
  -27  -13  481  998  614   34  -39
  -26  -19  459  995  636   44  -41
  -24  -25  437  990  658   55  -43
  -22  -29  414  983  679   67  -44
  -20  -34  393  976  700   79  -46
  -18  -38  371  968  721   91  -47
  -17  -41  350  959  742  104  -49
  -15  -44  330  948  761  118  -50
  -13  -46  309  936  780  133  -51
  -12  -48  289  924  799  148  -52
  -11  -50  270  911  817  163  -52
   -9  -51  250  897  833  180  -52
   -8  -52  232  882  850  196  -52
  -52  213  863  863  213  -52    0
  -52  196  850  882  232  -52   -8
  -52  180  833  897  250  -51   -9
  -52  163  817  911  270  -50  -11
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  -52  148  799  924  289  -48  -12
  -51  133  780  936  309  -46  -13
  -50  118  761  948  330  -44  -15
  -49  104  742  959  350  -41  -17
  -47   91  721  968  371  -38  -18
  -46   79  700  976  393  -34  -20
  -44   67  679  983  414  -29  -22
  -43   55  658  990  437  -25  -24
  -41   44  636  995  459  -19  -26
  -39   34  614  998  481  -13  -27
  -37   25  592 1001  503   -7  -29
  -35   16  570 1003  525    0  -31
```

#### 12.2.11.4.1.4 ppfcoef_scale_eq_10div16_32_phases_flip.dat

```
7  32 11
  -30  -46  542 1116  542  -46  -30
  -28  -52  515 1115  570  -39  -33
  -25  -57  488 1113  597  -32  -36
  -23  -62  462 1109  624  -24  -38
  -20  -65  435 1104  650  -15  -41
  -18  -69  409 1097  678   -5  -44
  -16  -71  383 1089  704    6  -47
  -14  -74  358 1081  730   17  -50
  -12  -75  333 1070  756   29  -53
  -11  -76  309 1058  782   42  -56
   -9  -77  285 1045  806   56  -58
   -8  -77  262 1030  831   71  -61
   -6  -77  239 1015  855   86  -64
   -5  -76  218  997  877  103  -66
   -4  -75  196  980  899  120  -68
   -3  -74  176  961  920  138  -70
  -72  156  940  940  156  -72    0
  -70  138  920  961  176  -74   -3
  -68  120  899  980  196  -75   -4
  -66  103  877  997  218  -76   -5
  -64   86  855 1015  239  -77   -6
  -61   71  831 1030  262  -77   -8
  -58   56  806 1045  285  -77   -9
  -56   42  782 1058  309  -76  -11
  -53   29  756 1070  333  -75  -12
  -50   17  730 1081  358  -74  -14
  -47    6  704 1089  383  -71  -16
  -44   -5  678 1097  409  -69  -18
  -41  -15  650 1104  435  -65  -20
  -38  -24  624 1109  462  -62  -23
  -36  -32  597 1113  488  -57  -25
  -33  -39  570 1115  515  -52  -28
```

#### 12.2.11.4.1.5 ppfcoef_scale_eq_11div16_32_phases_flip.dat

```
7  32 11
  -19  -94  522 1230  522  -94  -19
  -17  -98  490 1230  555  -90  -22
  -14 -100  458 1227  587  -85  -25
  -12 -102  427 1223  620  -79  -29
  -10 -103  397 1217  652  -73  -32
   -8 -104  367 1209  685  -65  -36
   -6 -104  337 1199  717  -56  -39
   -4 -103  309 1187  749  -47  -43
   -3 -102  281 1174  781  -36  -47
   -1 -100  253 1159  812  -24  -51
    0  -98  227 1142  843  -11  -55
    1  -96  201 1124  874    3  -59
    1  -93  177 1105  903   18  -63
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  2  -90  153 1084  932   34  -67
  2  -87  131 1062  961   51  -72
  3  -83  109 1038  987   69  -75
-79   89 1014 1014   89  -79    0
-75   69  987 1038  109  -83    3
-72   51  961 1062  131  -87    2
-67   34  932 1084  153  -90    2
-63   18  903 1105  177  -93    1
-59    3  874 1124  201  -96    1
-55  -11  843 1142  227  -98    0
-51  -24  812 1159  253 -100   -1
-47  -36  781 1174  281 -102   -3
-43  -47  749 1187  309 -103   -4
-39  -56  717 1199  337 -104   -6
-36  -65  685 1209  367 -104   -8
-32  -73  652 1217  397 -103  -10
-29  -79  620 1223  427 -102  -12
-25  -85  587 1227  458 -100  -14
-22  -90  555 1230  490  -98  -17
```

#### 12.2.11.4.1.6 ppfcoef_scale_eq_12div16_32_phases_flip.dat

```
7  32 11
   -3 -132  486 1346  486 -132   -3
   -1 -132  449 1345  524 -131   -6
    1 -131  413 1342  562 -130   -9
    3 -130  378 1336  600 -127  -12
    4 -128  343 1328  639 -123  -15
    5 -125  309 1319  677 -119  -18
    6 -122  277 1306  716 -113  -22
    7 -118  245 1292  754 -106  -26
    8 -114  214 1276  793  -98  -31
    8 -109  185 1257  831  -89  -35
    9 -105  156 1237  869  -78  -40
    9 -100  130 1214  906  -66  -45
    9  -94  104 1190  942  -53  -50
    9  -89   79 1165  978  -38  -56
    8  -83   56 1138 1012  -22  -61
    8  -78   35 1108 1046   -4  -67
  -72   15 1081 1081   15  -72    0
  -67   -4 1046 1108   35  -78    8
  -61  -22 1012 1138   56  -83    8
  -56  -38  978 1165   79  -89    9
  -50  -53  942 1190  104  -94    9
  -45  -66  906 1214  130 -100    9
  -40  -78  869 1237  156 -105    9
  -35  -89  831 1257  185 -109    8
  -31  -98  793 1276  214 -114    8
  -26 -106  754 1292  245 -118    7
  -22 -113  716 1306  277 -122    6
  -18 -119  677 1319  309 -125    5
  -15 -123  639 1328  343 -128    4
  -12 -127  600 1336  378 -130    3
   -9 -130  562 1342  413 -131    1
   -6 -131  524 1345  449 -132   -1
```

#### 12.2.11.4.1.7 ppfcoef_scale_eq_13div16_32_phases_flip.dat

```
7  32 11
   14 -154  435 1458  435 -154   14
   15 -150  393 1458  477 -157   12
   16 -146  353 1454  521 -160   10
   16 -141  314 1447  565 -161    8
   17 -135  276 1436  609 -161    6
   17 -129  239 1425  654 -161    3
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  17 -123  204 1410  699 -159    0
  16 -116  170 1393  745 -156   -4
  16 -109  137 1373  790 -151   -8
  16 -102  107 1350  835 -146  -12
  15  -94   77 1325  879 -138  -16
  14  -87   50 1298  924 -130  -21
  13  -80   24 1269  968 -119  -27
  12  -72    0 1238 1010 -107  -33
  11  -65  -22 1204 1053  -94  -39
  10  -58  -43 1169 1093  -78  -45
 -52  -62 1138 1138  -62  -52    0
 -45  -78 1093 1169  -43  -58   10
 -39  -94 1053 1204  -22  -65   11
 -33 -107 1010 1238    0  -72   12
 -27 -119  968 1269   24  -80   13
 -21 -130  924 1298   50  -87   14
 -16 -138  879 1325   77  -94   15
 -12 -146  835 1350  107 -102   16
  -8 -151  790 1373  137 -109   16
  -4 -156  745 1393  170 -116   16
   0 -159  699 1410  204 -123   17
   3 -161  654 1425  239 -129   17
   6 -161  609 1436  276 -135   17
   8 -161  565 1447  314 -141   16
  10 -160  521 1454  353 -146   16
  12 -157  477 1458  393 -150   15
```

#### 12.2.11.4.1.8 ppfcoef_scale_eq_14div16_32_phases_flip.dat

```
7  32 11
  27 -158  370 1570  370 -158   27
  27 -150  324 1568  417 -165   27
  26 -142  281 1563  465 -172   27
  25 -133  238 1555  515 -178   26
  24 -124  198 1543  565 -183   25
  23 -115  159 1527  616 -186   24
  22 -106  122 1510  667 -189   22
  21  -97   87 1489  719 -191   20
  19  -87   54 1464  772 -191   17
  18  -78   23 1437  824 -190   14
  16  -69   -6 1407  876 -187   11
  15  -60  -32 1373  927 -182    7
  13  -52  -57 1339  979 -176    2
  12  -44  -79 1300 1030 -168   -3
  11  -36  -99 1261 1079 -159   -9
   9  -28 -117 1218 1128 -147  -15
 -21 -134 1179 1179 -134  -21    0
 -15 -147 1128 1218 -117  -28    9
  -9 -159 1079 1261  -99  -36   11
  -3 -168 1030 1300  -79  -44   12
   2 -176  979 1339  -57  -52   13
   7 -182  927 1373  -32  -60   15
  11 -187  876 1407   -6  -69   16
  14 -190  824 1437   23  -78   18
  17 -191  772 1464   54  -87   19
  20 -191  719 1489   87  -97   21
  22 -189  667 1510  122 -106   22
  24 -186  616 1527  159 -115   23
  25 -183  565 1543  198 -124   24
  26 -178  515 1555  238 -133   25
  27 -172  465 1563  281 -142   26
  27 -165  417 1568  324 -150   27
```


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.4.1.9 ppfcoef_scale_eq_15div16_32_phases_flip.dat

```
7  32 11
   33 -143  294 1680  294 -143   33
   31 -132  246 1678  345 -155   35
   30 -121  199 1671  398 -165   36
   27 -109  154 1661  452 -175   38
   25  -97  112 1647  508 -185   38
   23  -86   72 1629  564 -193   39
   21  -75   35 1607  622 -201   39
   19  -64    0 1580  681 -207   39
   17  -53  -32 1551  740 -213   38
   15  -43  -61 1518  799 -217   37
   13  -33  -88 1481  859 -219   35
   11  -24 -113 1442  919 -220   33
    9  -15 -134 1399  978 -219   30
    8   -7 -153 1354 1036 -217   27
    6    0 -170 1307 1094 -212   23
    5    7 -184 1257 1150 -205   18
   13 -196 1207 1207 -196   13    0
   18 -205 1150 1257 -184    7    5
   23 -212 1094 1307 -170    0    6
   27 -217 1036 1354 -153   -7    8
   30 -219  978 1399 -134  -15    9
   33 -220  919 1442 -113  -24   11
   35 -219  859 1481  -88  -33   13
   37 -217  799 1518  -61  -43   15
   38 -213  740 1551  -32  -53   17
   39 -207  681 1580    0  -64   19
   39 -201  622 1607   35  -75   21
   39 -193  564 1629   72  -86   23
   38 -185  508 1647  112  -97   25
   38 -175  452 1661  154 -109   27
   36 -165  398 1671  199 -121   30
   35 -155  345 1678  246 -132   31
```

### 12.2.11.4.2 VS Polyphase Filter Coefficients

#### 12.2.11.4.2.1 ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat

```
5  32 11
  -47  177 1788  177  -47
  -40  133 1785  225  -55
  -33   91 1778  276  -64
  -27   53 1765  330  -73
  -21   18 1747  386  -82
  -15  -13 1722  445  -91
  -11  -41 1693  507 -100
   -7  -66 1660  570 -109
   -3  -88 1622  635 -118
    0 -107 1579  703 -127
    2 -122 1532  771 -135
    4 -135 1482  839 -142
    5 -145 1428  909 -149
    6 -153 1371  978 -154
    7 -158 1310 1047 -158
    7 -161 1247 1116 -161
 -162 1186 1186 -162    0
 -161 1116 1247 -161    7
 -158 1047 1310 -158    7
 -154  978 1371 -153    6
 -149  909 1428 -145    5
 -142  839 1482 -135    4
 -135  771 1532 -122    2
 -127  703 1579 -107    0
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
 -118  635 1622  -88   -3
 -109  570 1660  -66   -7
 -100  507 1693  -41  -11
  -91  445 1722  -13  -15
  -82  386 1747   18  -21
  -73  330 1765   53  -27
  -64  276 1778   91  -33
  -55  225 1785  133  -40
```

#### 12.2.11.4.2.2 ppfcoef_scale_eq_3_32_phases_flip.dat

```
5,   32,   11,
  130,  515,  758,  515,  130,
  121,  503,  757,  528,  139,
  113,  490,  756,  541,  148,
  105,  477,  755,  553,  158,
   97,  464,  753,  566,  168,
   90,  451,  751,  578,  178,
   83,  437,  749,  590,  189,
   76,  424,  746,  602,  200,
   69,  411,  743,  614,  211,
   63,  398,  739,  626,  222,
   57,  386,  734,  637,  234,
   52,  373,  729,  648,  246,
   46,  360,  725,  659,  258,
   41,  347,  719,  670,  271,
   37,  335,  713,  680,  283,
   32,  322,  707,  690,  297,
  314,  710,  710,  314,    0,
  297,  690,  707,  322,   32,
  283,  680,  713,  335,   37,
  271,  670,  719,  347,   41,
  258,  659,  725,  360,   46,
  246,  648,  729,  373,   52,
  234,  637,  734,  386,   57,
  222,  626,  739,  398,   63,
  211,  614,  743,  411,   69,
  200,  602,  746,  424,   76,
  189,  590,  749,  437,   83,
  178,  578,  751,  451,   90,
  168,  566,  753,  464,   97,
  158,  553,  755,  477,  105,
  148,  541,  756,  490,  113,
  139,  528,  757,  503,  121};
```

#### 12.2.11.4.2.3 ppfcoef_scale_eq_4_32_phases_flip.dat

```
5,   32,   11,
  116,  515,  786,  515,  116,
  107,  502,  785,  530,  124,
   99,  488,  784,  544,  133,
   92,  473,  783,  557,  143,
   85,  459,  781,  571,  152,
   78,  445,  778,  585,  162,
   71,  431,  775,  598,  173,
   65,  417,  772,  611,  183,
   59,  403,  767,  624,  195,
   53,  389,  763,  637,  206,
   48,  375,  758,  649,  218,
   43,  362,  752,  661,  230,
   38,  348,  747,  673,  242,
   34,  334,  740,  685,  255,
   30,  321,  733,  696,  268,
   26,  308,  726,  707,  281,
  298,  726,  726,  298,    0,
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  281,  707,  726,  308,   26,
  268,  696,  733,  321,   30,
  255,  685,  740,  334,   34,
  242,  673,  747,  348,   38,
  230,  661,  752,  362,   43,
  218,  649,  758,  375,   48,
  206,  637,  763,  389,   53,
  195,  624,  767,  403,   59,
  183,  611,  772,  417,   65,
  173,  598,  775,  431,   71,
  162,  585,  778,  445,   78,
  152,  571,  781,  459,   85,
  143,  557,  783,  473,   92,
  133,  544,  784,  488,   99,
  124,  530,  785,  502,  107};
```

#### 12.2.11.4.2.4 ppfcoef_scale_eq_5_32_phases_flip.dat

```
5,   32,   11,
   98,  515,  822,  515,   98,
   90,  500,  821,  531,  106,
   83,  484,  820,  547,  114,
   75,  469,  819,  562,  123,
   69,  453,  816,  577,  133,
   63,  438,  813,  592,  142,
   57,  422,  809,  607,  153,
   51,  407,  805,  622,  163,
   46,  391,  801,  636,  174,
   41,  376,  795,  650,  186,
   37,  361,  789,  664,  197,
   32,  347,  782,  678,  209,
   28,  332,  775,  691,  222,
   25,  317,  767,  704,  235,
   22,  303,  759,  716,  248,
   18,  289,  750,  729,  262,
  278,  746,  746,  278,    0,
  262,  729,  750,  289,   18,
  248,  716,  759,  303,   22,
  235,  704,  767,  317,   25,
  222,  691,  775,  332,   28,
  209,  678,  782,  347,   32,
  197,  664,  789,  361,   37,
  186,  650,  795,  376,   41,
  174,  636,  801,  391,   46,
  163,  622,  805,  407,   51,
  153,  607,  809,  422,   57,
  142,  592,  813,  438,   63,
  133,  577,  816,  453,   69,
  123,  562,  819,  469,   75,
  114,  547,  820,  484,   83,
  106,  531,  821,  500,   90};
```

#### 12.2.11.4.2.5 ppfcoef_scale_eq_6_32_phases_flip.dat

```
5,   32,   11,
   77,  513,  868,  513,   77,
   70,  496,  867,  531,   84,
   63,  479,  866,  548,   92,
   57,  461,  864,  566,  100,
   51,  444,  861,  583,  109,
   46,  427,  857,  600,  118,
   41,  409,  853,  617,  128,
   36,  393,  847,  633,  139,
   32,  376,  841,  650,  149,
   28,  359,  835,  666,  160,
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  24,  343,  827,  682,  172,
  21,  327,  819,  697,  184,
  18,  311,  810,  712,  197,
  15,  296,  800,  727,  210,
  13,  281,  790,  741,  223,
  11,  266,  779,  755,  237,
 253,  771,  771,  253,    0,
 237,  755,  779,  266,   11,
 223,  741,  790,  281,   13,
 210,  727,  800,  296,   15,
 197,  712,  810,  311,   18,
 184,  697,  819,  327,   21,
 172,  682,  827,  343,   24,
 160,  666,  835,  359,   28,
 149,  650,  841,  376,   32,
 139,  633,  847,  393,   36,
 128,  617,  853,  409,   41,
 118,  600,  857,  427,   46,
 109,  583,  861,  444,   51,
 100,  566,  864,  461,   57,
  92,  548,  866,  479,   63,
  84,  531,  867,  496,   70};
```

#### 12.2.11.4.2.6 ppfcoef_scale_eq_7_32_phases_flip.dat

```
5,   32,   11,
   53,  510,  922,  510,   53,
   47,  490,  922,  529,   60,
   41,  470,  921,  549,   67,
   36,  451,  918,  569,   74,
   32,  431,  915,  588,   82,
   27,  412,  910,  608,   91,
   23,  393,  905,  627,  100,
   20,  374,  898,  646,  110,
   17,  356,  890,  665,  120,
   14,  337,  882,  684,  131,
   11,  320,  873,  702,  142,
    9,  302,  863,  720,  154,
    7,  285,  852,  737,  167,
    6,  269,  840,  753,  180,
    4,  253,  827,  770,  194,
    3,  237,  815,  785,  208,
  223,  801,  801,  223,    0,
  208,  785,  815,  237,    3,
  194,  770,  827,  253,    4,
  180,  753,  840,  269,    6,
  167,  737,  852,  285,    7,
  154,  720,  863,  302,    9,
  142,  702,  873,  320,   11,
  131,  684,  882,  337,   14,
  120,  665,  890,  356,   17,
  110,  646,  898,  374,   20,
  100,  627,  905,  393,   23,
   91,  608,  910,  412,   27,
   82,  588,  915,  431,   32,
   74,  569,  918,  451,   36,
   67,  549,  921,  470,   41,
   60,  529,  922,  490,   47};
```

#### 12.2.11.4.2.6.1 ppfcoef_scale_eq_8div16_32_phases_ver_5tap_flip.dat

```
5  32 11
   28  502  988  502   28
   24  479  987  524   34
   19  457  985  547   40
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
 15  435  982  570   46
 12  413  978  592   53
  9  392  972  614   61
  6  371  965  637   69
  4  350  957  659   78
  2  330  948  680   88
  0  310  938  702   98
 -1  291  926  723  109
 -2  272  914  744  120
 -3  254  900  764  133
 -3  237  886  783  145
 -4  220  871  802  159
 -4  204  855  820  173
188  836  836  188    0
173  820  855  204   -4
159  802  871  220   -4
145  783  886  237   -3
133  764  900  254   -3
120  744  914  272   -2
109  723  926  291   -1
 98  702  938  310    0
 88  680  948  330    2
 78  659  957  350    4
 69  637  965  371    6
 61  614  972  392    9
 53  592  978  413   12
 46  570  982  435   15
 40  547  985  457   19
 34  524  987  479   24
```

#### 12.2.11.4.2.6.2 ppfcoef_scale_eq_9div16_32_phases_ver_5tap_flip.dat

```
5  32 11
    3  489 1064  489    3
    0  464 1062  515    7
   -3  439 1060  540   12
   -5  414 1056  566   17
   -7  390 1050  592   23
   -9  366 1044  618   29
  -10  343 1035  644   36
  -11  320 1025  670   44
  -12  298 1014  695   53
  -12  277 1001  720   62
  -12  256  987  745   72
  -12  236  972  769   83
  -12  217  956  792   95
  -11  199  938  815  107
  -10  181  920  837  120
  -10  165  900  859  134
  148  876  876  148    0
  134  859  900  165  -10
  120  837  920  181  -10
  107  815  938  199  -11
   95  792  956  217  -12
   83  769  972  236  -12
   72  745  987  256  -12
   62  720 1001  277  -12
   53  695 1014  298  -12
   44  670 1025  320  -11
   36  644 1035  343  -10
   29  618 1044  366   -9
   23  592 1050  390   -7
   17  566 1056  414   -5
   12  540 1060  439   -3
    7  515 1062  464    0
```


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.4.2.6.3 ppfcoef_scale_eq_10div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -20  470 1148  470  -20
  -22  442 1147  499  -18
  -23  413 1144  529  -15
  -24  386 1139  558  -11
  -24  359 1132  588   -7
  -24  333 1124  618   -3
  -24  308 1113  648    3
  -23  283 1101  678    9
  -23  260 1088  707   16
  -22  237 1072  737   24
  -21  215 1056  765   33
  -19  194 1037  793   43
  -18  174 1017  822   53
  -16  156  995  848   65
  -15  138  973  875   77
  -13  121  949  900   91
  105  919  919  105    0
   91  900  949  121  -13
   77  875  973  138  -15
   65  848  995  156  -16
   53  822 1017  174  -18
   43  793 1037  194  -19
   33  765 1056  215  -21
   24  737 1072  237  -22
   16  707 1088  260  -23
    9  678 1101  283  -23
    3  648 1113  308  -24
   -3  618 1124  333  -24
   -7  588 1132  359  -24
  -11  558 1139  386  -24
  -15  529 1144  413  -23
  -18  499 1147  442  -22
```

#### 12.2.11.4.2.6.4 ppfcoef_scale_eq_11div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -40  444 1240  444  -40
  -40  412 1240  476  -40
  -40  381 1236  510  -39
  -39  350 1231  544  -38
  -37  321 1223  577  -36
  -36  293 1212  612  -33
  -34  265 1200  646  -29
  -32  239 1185  681  -25
  -30  214 1169  715  -20
  -28  190 1150  750  -14
  -26  167 1130  783   -6
  -23  146 1107  816    2
  -21  126 1083  849   11
  -19  107 1057  882   21
  -17   90 1030  913   32
  -15   73 1002  943   45
   58  966  966   58    0
   45  943 1002   73  -15
   32  913 1030   90  -17
   21  882 1057  107  -19
   11  849 1083  126  -21
    2  816 1107  146  -23
   -6  783 1130  167  -26
  -14  750 1150  190  -28
  -20  715 1169  214  -30
  -25  681 1185  239  -32
  -29  646 1200  265  -34
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  -33  612 1212  293  -36
  -36  577 1223  321  -37
  -38  544 1231  350  -39
  -39  510 1236  381  -40
  -40  476 1240  412  -40
```

#### 12.2.11.4.2.6.5 ppfcoef_scale_eq_12div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -56  409 1342  409  -56
  -54  373 1342  445  -58
  -51  339 1337  482  -59
  -49  306 1330  521  -60
  -46  274 1321  559  -60
  -43  244 1308  598  -59
  -40  215 1293  638  -58
  -36  187 1275  678  -56
  -33  161 1255  718  -53
  -30  137 1233  757  -49
  -27  114 1208  797  -44
  -24   93 1182  836  -39
  -21   73 1152  875  -31
  -18   55 1122  912  -23
  -16   38 1090  950  -14
  -14   23 1056  986   -3
    9 1015 1015    9    0
   -3  986 1056   23  -14
  -14  950 1090   38  -16
  -23  912 1122   55  -18
  -31  875 1152   73  -21
  -39  836 1182   93  -24
  -44  797 1208  114  -27
  -49  757 1233  137  -30
  -53  718 1255  161  -33
  -56  678 1275  187  -36
  -58  638 1293  215  -40
  -59  598 1308  244  -43
  -60  559 1321  274  -46
  -60  521 1330  306  -49
  -59  482 1337  339  -51
  -58  445 1342  373  -54
```

#### 12.2.11.4.2.6.6 ppfcoef_scale_eq_13div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -65  364 1450  364  -65
  -61  326 1448  404  -69
  -57  289 1443  445  -72
  -53  253 1435  488  -75
  -48  220 1423  531  -78
  -44  188 1408  576  -80
  -40  158 1390  621  -81
  -36  130 1370  666  -82
  -32  103 1346  713  -82
  -28   79 1320  758  -81
  -24   56 1290  805  -79
  -21   36 1259  850  -76
  -18   17 1224  896  -71
  -15    0 1188  940  -65
  -12  -15 1149  984  -58
  -10  -28 1109 1027  -50
  -40 1064 1064  -40    0
  -50 1027 1109  -28  -10
  -58  984 1149  -15  -12
  -65  940 1188    0  -15
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  -71  896 1224   17  -18
  -76  850 1259   36  -21
  -79  805 1290   56  -24
  -81  758 1320   79  -28
  -82  713 1346  103  -32
  -82  666 1370  130  -36
  -81  621 1390  158  -40
  -80  576 1408  188  -44
  -78  531 1423  220  -48
  -75  488 1435  253  -53
  -72  445 1443  289  -57
  -69  404 1448  326  -61
```

#### 12.2.11.4.2.6.7 ppfcoef_scale_eq_14div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -67  310 1562  310  -67
  -61  269 1559  353  -72
  -55  230 1553  398  -78
  -50  193 1543  445  -83
  -44  158 1529  493  -88
  -39  125 1512  543  -93
  -34   94 1491  594  -97
  -30   66 1468  645 -101
  -25   41 1439  697 -104
  -22   17 1408  751 -106
  -18   -4 1373  804 -107
  -15  -23 1336  857 -107
  -12  -40 1296  910 -106
   -9  -55 1253  962 -103
   -7  -67 1208 1013  -99
   -5  -78 1161 1064  -94
  -86 1110 1110  -86    0
  -94 1064 1161  -78   -5
  -99 1013 1208  -67   -7
 -103  962 1253  -55   -9
 -106  910 1296  -40  -12
 -107  857 1336  -23  -15
 -107  804 1373   -4  -18
 -106  751 1408   17  -22
 -104  697 1439   41  -25
 -101  645 1468   66  -30
  -97  594 1491   94  -34
  -93  543 1512  125  -39
  -88  493 1529  158  -44
  -83  445 1543  193  -50
  -78  398 1553  230  -55
  -72  353 1559  269  -61
```

#### 12.2.11.4.2.6.8 ppfcoef_scale_eq_15div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -61  248 1674  248  -61
  -54  204 1673  293  -68
  -47  163 1665  342  -75
  -41  125 1654  392  -82
  -35   90 1638  445  -90
  -29   57 1618  499  -97
  -24   27 1593  556 -104
  -20    0 1565  613 -110
  -16  -24 1532  672 -116
  -12  -46 1495  732 -121
   -9  -65 1455  793 -126
   -6  -81 1411  854 -130
   -4  -95 1364  915 -132
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
  -2 -107 1315  975 -133
   0 -116 1262 1035 -133
   1 -123 1208 1094 -132
-128 1152 1152 -128    0
-132 1094 1208 -123    1
-133 1035 1262 -116    0
-133  975 1315 -107   -2
-132  915 1364  -95   -4
-130  854 1411  -81   -6
-126  793 1455  -65   -9
-121  732 1495  -46  -12
-116  672 1532  -24  -16
-110  613 1565    0  -20
-104  556 1593   27  -24
 -97  499 1618   57  -29
 -90  445 1638   90  -35
 -82  392 1654  125  -41
 -75  342 1665  163  -47
 -68  293 1673  204  -54
```

### ppcoef_scale_1x_ver_5tap.dat

```
5    32    11
      0       0    2048       0       0
    -40     133    1785     225     -55
    -33      91    1778     276     -64
    -27      53    1765     330     -73
    -21      18    1747     386     -82
    -15     -13    1722     445     -91
    -11     -41    1693     507    -100
     -7     -66    1660     570    -109
     -3     -88    1622     635    -118
      0    -107    1579     703    -127
      2    -122    1532     771    -135
      4    -135    1482     839    -142
      5    -145    1428     909    -149
      6    -153    1371     978    -154
      7    -158    1310    1047    -158
      7    -161    1247    1116    -161
    162    1186    1186    -162       0
    161    1116    1247    -161       7
    158    1047    1310    -158       7
    154     978    1371    -153       6
    149     909    1428    -145       5
    142     839    1482    -135       4
    135     771    1532    -122       2
    127     703    1579    -107       0
    118     635    1622     -88      -3
    109     570    1660     -66      -7
    100     507    1693     -41     -11
    -91     445    1722     -13     -15
    -82     386    1747      18     -21
    -73     330    1765      53     -27
    -64     276    1778      91     -33
    -55     225    1785     133     -40
```


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.11.4.3 VS (Bilinear Filter Coefficients)

##### 12.2.11.4.3.1 ppfcoef_scale_eq_1_32_phases_flip_PPF3_peak5_gain_eq_1_25.dat

This is not applicable for this device.

```
7  32 11
  -11 -106  190 1902  190 -106  -11
   -9 -105  153 1897  230 -105  -13
   -8 -105  121 1887  274 -105  -16
   -6 -104   91 1869  320 -104  -18
   -5 -102   65 1843  370 -102  -21
   -4 -101   42 1812  424 -101  -24
   -3  -99   20 1776  480  -99  -27
   -2  -96    3 1730  539  -96  -30
   -1  -93  -12 1679  602  -93  -34
   -1  -90  -26 1627  665  -90  -37
    0  -87  -37 1568  732  -87  -41
    0  -84  -46 1506  801  -84  -45
    0  -80  -54 1439  871  -80  -48
    0  -76  -60 1371  941  -76  -52
    1  -72  -65 1299 1013  -72  -56
    1  -68  -69 1227 1085  -68  -60
  -64  -64 1152 1152  -64  -64    0
  -60  -68 1085 1227  -69  -68    1
  -56  -72 1013 1299  -65  -72    1
  -52  -76  941 1371  -60  -76    0
  -48  -80  871 1439  -54  -80    0
  -45  -84  801 1506  -46  -84    0
  -41  -87  732 1568  -37  -87    0
  -37  -90  665 1627  -26  -90   -1
  -34  -93  602 1679  -12  -93   -1
  -30  -96  539 1730    3  -96   -2
  -27  -99  480 1776   20  -99   -3
  -24 -101  424 1812   42 -101   -4
  -21 -102  370 1843   65 -102   -5
  -18 -104  320 1869   91 -104   -6
  -16 -105  274 1887  121 -105   -8
  -13 -105  230 1897  153 -105   -9
```


Copyright © 2011–2016, Texas Instruments Incorporated

## 12.2.12 Video Compositer (VCOMP)

### 12.2.12.1 Overview

The Video Compositor module composites two sources of input video over a background color layer. Both input sources are in 4:2:2 YUV format. The output of the module is also 4:2:2 YUV.

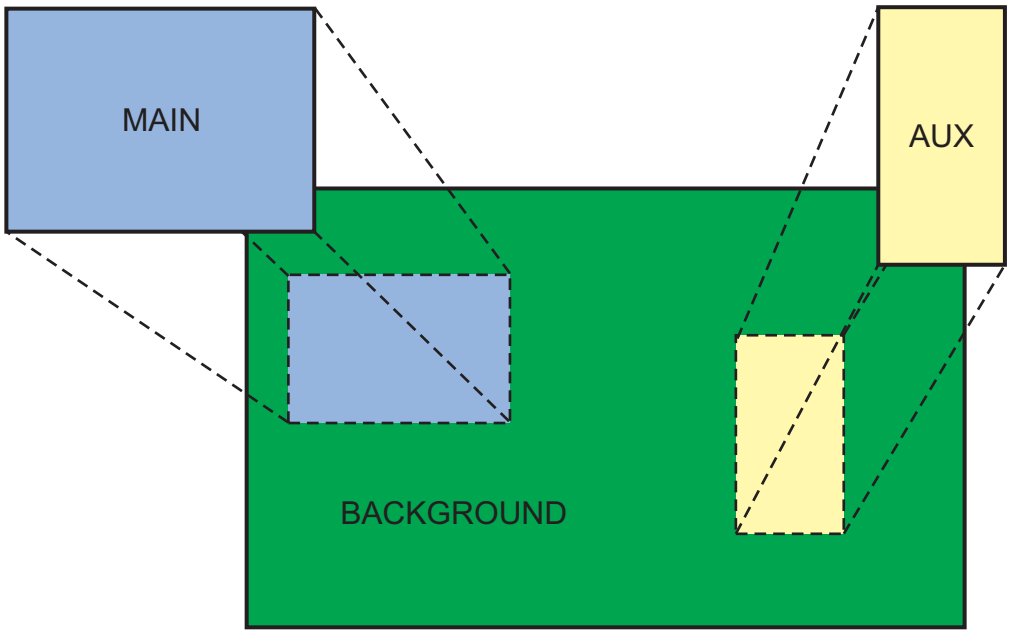
### 12.2.12.2 Features Supported

- Flexible placement of Main and Auxiliary layers on the background color
- Main and Auxiliary layers can be cropped in both horizontal and vertical directions
- Configurable background color in YUV422
- Optional disabling of Main or Auxiliary or both layers
- Alternate color selection when the input is disabled

### 12.2.12.3 Functional Description

The VCOMP module takes 4:2:2 YUV input from two sources called Main (Main) and Auxiliary (Aux) layers. The VCOMP has a default background color layer that is the size of the output. As shown in Figure 12-157, the Main and Aux layers are composited over the background color.

**Figure 12-157. Main and Auxiliary Layers Over Background**

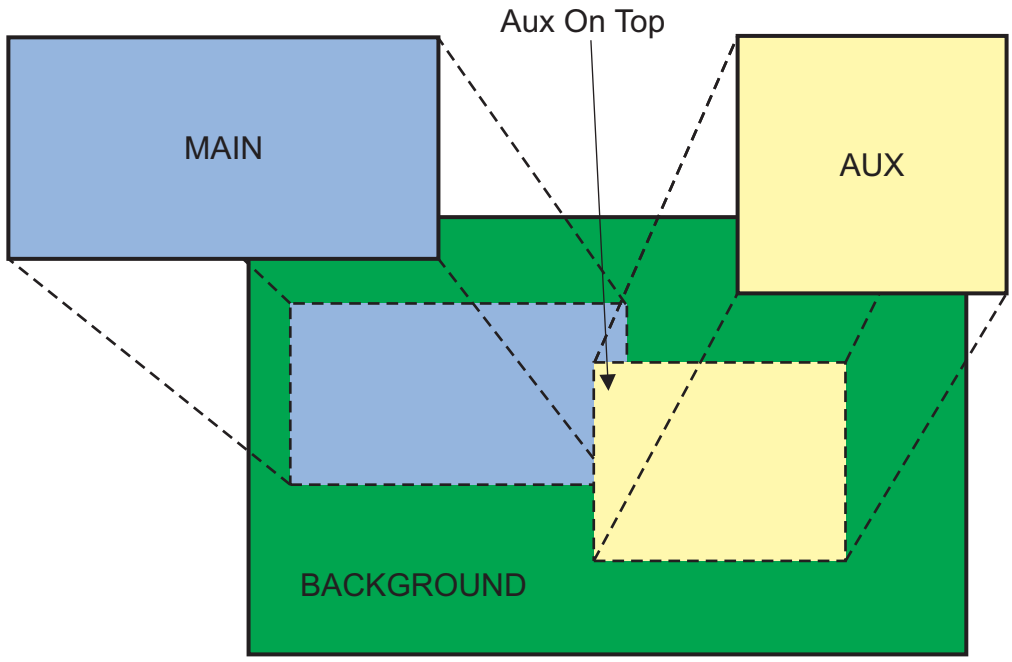




Copyright © 2011–2016, Texas Instruments Incorporated

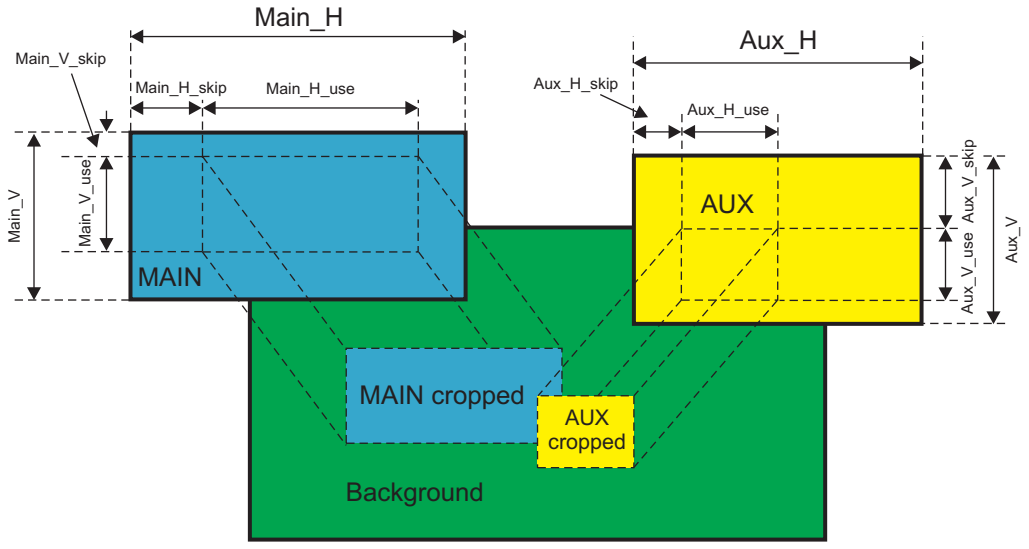
Pixels where the Main and Aux overlap are controlled by a global configuration bit (VCOMP→reg9.cfg_main_aux_n_ontop) that determines whether the Main or the Aux is on top. An example of the config bit set for Aux on Top is shown in Figure 12-158.

**Figure 12-158. Overlap Region**




The Main and Aux layers can be positioned anywhere over the background layer. The Main and Aux layers can optionally be cropped in the horizontal and vertical directions as shown in Figure 12-159.

**Figure 12-159. Cropping of Input Layers**





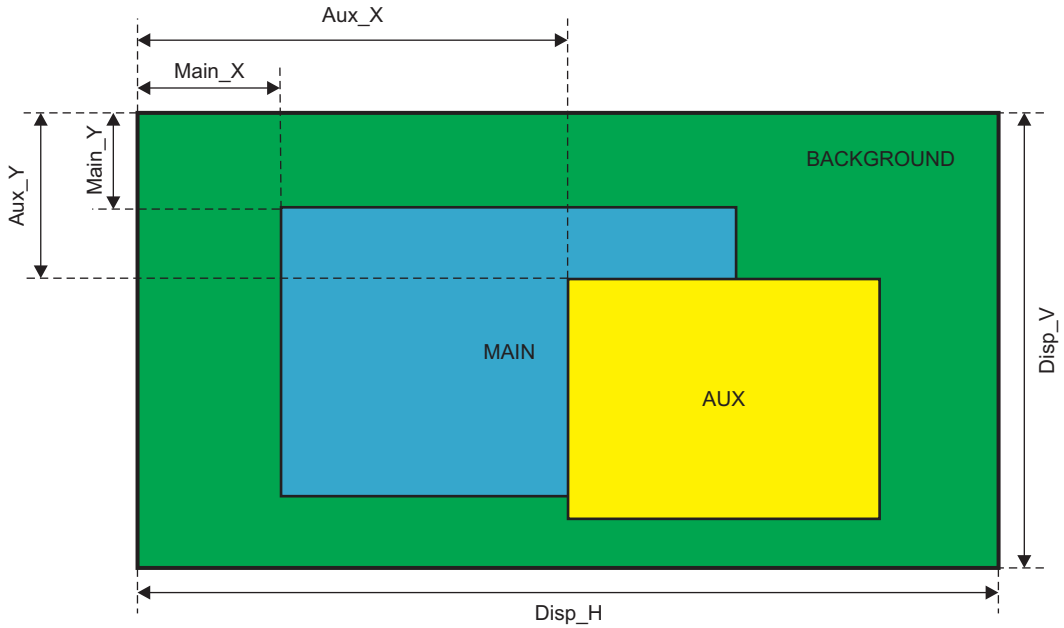
Copyright © 2011–2016, Texas Instruments Incorporated

The configuration of parameters shown in Figure 12-159 is done using the following register fields:

```
Main_H          = VCOMP->reg0.cfg_main_native_numpix_per_line
Main_H_skip     = VCOMP->reg1.cfg_main_skip_numpix
Main_H_use      = VCOMP->reg1.cfg_main_use_numpix
Man_V           = VCOMP->reg0.cfg_main_native_numlines
Main_V_skip     = VCOMP->reg2.cfg_main_skip_numlines
Main_V_use      = VCOMP->reg2.cfg_main_use_numlines
Aux_H           = VCOMP->reg3.cfg_aux_native_numpix_per_line
Aux_H_skip      = VCOMP->reg4.cfg_aux_skip_numpix
Aux_H_use       = VCOMP->reg4.cfg_aux_use_numpix
Aux_V           = VCOMP->reg3.cfg_aux_native_numlines
Aux_V_skip      = VCOMP->reg5.cfg_aux_skip_numlines
Aux_V_use       = VCOMP->reg5.cfg_aux_use_numlines
```

The output of the VCOMP is a single 4:2:2 YUV plane, as shown in Figure 12-160.

**Figure 12-160. Single Plane Output**




The configuration of parameters shown in Figure 12-160 is done using the following register fields:

```
Disp_H    = VCOMP->reg6.cfg_dsply_numpix_per_line
Disp_V    = VCOMP->reg6. cfg_dsply_numlines
Main_X    = VCOMP->reg7.cfg_dsply_main_x_origin
Main_Y    = VCOMP->reg7.cfg_dsply_main_y_origin
Aux_X     = VCOMP->reg8.cfg_dsply_aux_x_origin
Aux_Y     = VCOMP->reg8.cfg_dsply_aux_y_origin
```

Background color configuration:

```
Y         = VCOMP->reg8.cfg_dsply_bckgrnd_y_val
Cb        = VCOMP->reg8.cfg_dsply_bckgrnd_cb_val
Cr        = VCOMP->reg8.cfg_dsply_bckgrnd_cr_val
```


Copyright © 2011–2016, Texas Instruments Incorporated

In the previous examples, both Main and Aux are active. However, Main can be the only source (Figure 12-161), Aux can be the only source (Figure 12-162), or both Main and Aux can be disabled and only the Background is sent out (Figure 12-163). In the last case, either the Main or the Aux inputs must still be sending data into the VCOMP even though the picture data is not used.

**Figure 12-161. Main Layer Only**

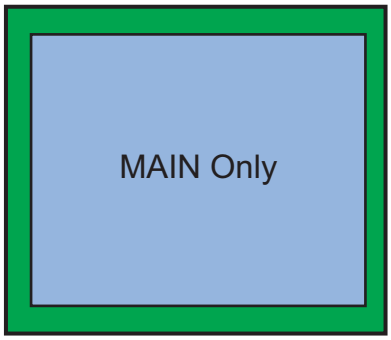



**Figure 12-162. Aux Layer Only**

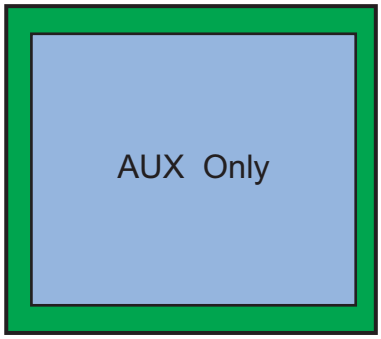
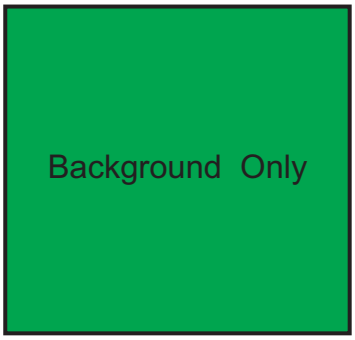



**Figure 12-163. No Layer Enabled**




To enable/disable Layers, the following register bitfields are used:

```
Main Layer    VCOMP->reg0.cfg_main_enable
Aux Layer     VCOMP->reg3.cfg_aux_enable
```

The Main and Aux sources, single or together, can completely cover the Background layer.

Alternate fixed color can be selected for main and aux layers when needed as shown below.

Main layer Fixed data enable/disable VCOMP→reg0.cfg_main_fixed_data_send


Copyright © 2011–2016, Texas Instruments Incorporated

Fixed data:

```
Y        = VCOMP->reg11.cfg_dsply_alt_main_y_val
Cb       = VCOMP->reg11.cfg_dsply_alt_main_cb_val
Cr       = VCOMP->reg11.cfg_dsply_alt_main_cr_val
```

Aux layer Fixed data enable/disable VCOMP→reg3.cfg_aux_fixed_data_send

Fixed data:

```
Y        = VCOMP->reg12.cfg_dsply_alt_aux_y_val
Cb       = VCOMP->reg12.cfg_dsply_alt_aux_cb_val
Cr       = VCOMP->reg12.cfg_dsply_alt_aux_cr_val
```

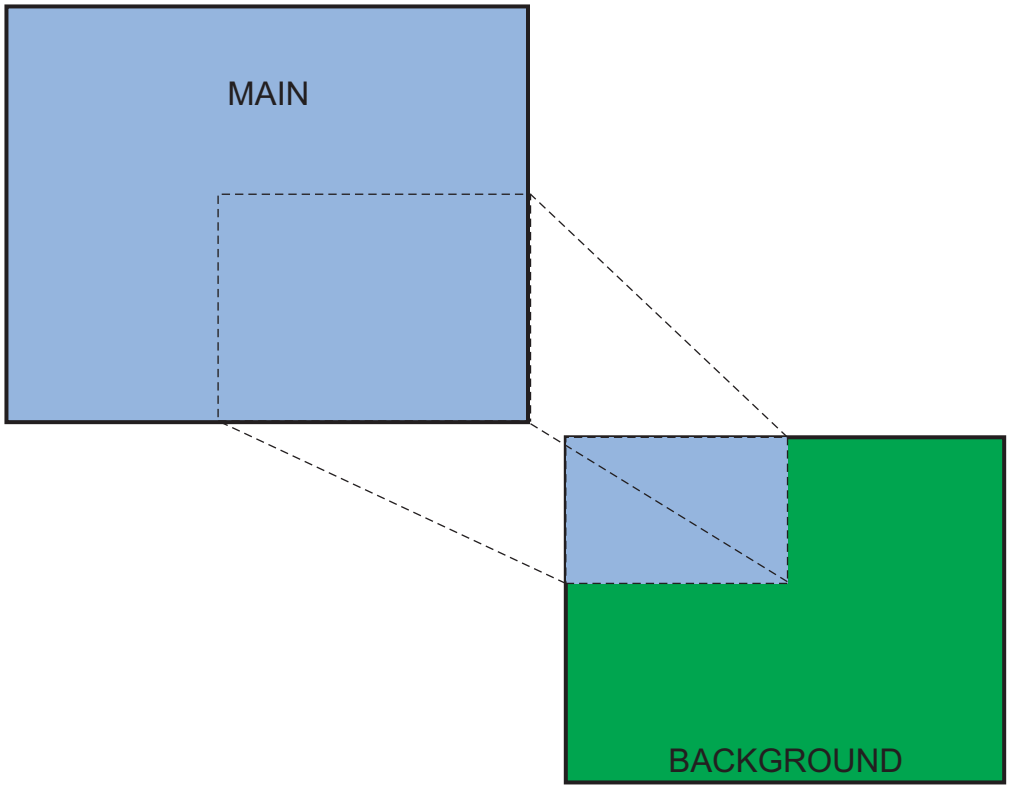
### 12.2.12.4 Exceptions

#### *12.2.12.4.1 Exception with Respect to Timing*

The VCOMP gets data from external memory and is generally used to setup HD Main Video Window for display operations through VENC. A VENC with a timebase generator determines the VCOMP's output timing requirements for when it needs to have pixel data ready. The VCOMP may not have pixel data ready for display if the input pixels have yet to arrive. The following is one scenario which can possibly create an issue.

In this case, the lower right quadrant of the Main layer is to be cropped and displayed on the top left corner of the output. This situation is shown in Figure 12-164. When the first pixel of the output is needed, the first pixel of the cropped region may not have entered the VCOMP yet. In this case, the configuration does not work for the device even though it is conceptually valid.

**Figure 12-164. Conditions Where Valid Configurations May Not Work in a System**





Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.12.4.2 Exception with Respect to Placement

Figure 12-165 contains examples that show if the placement of Aux layer with respect to Main Layer is supported or not.

Exception1: Do not start or end an inserted Main or Aux window at an odd horizontal pixel location. Item “pip034” shown in Figure 12-165 is not supported because it starts and ends at odd pixel number.

**Figure 12-165. Example**

Copyright © 2011–2016, Texas Instruments Incorporated

## 12.2.13 Video Port Direct Memory Access (VPDMA)

### 12.2.13.1 Introduction

#### 12.2.13.1.1 Overview

The primary function of the Video Port Direct Memory Access (VPDMA) is to move data between external memory and HD-VPSS processing modules that source or sink data. The modules that source or sink data are referred to as "clients". The physical interface between the processing module and external memory is called a client. A "channel (software mechanism)" is set up inside the VPDMA to connect a specific memory buffer to a specific client. The VPDMA centralizes the Direct Memory Access (DMA) control functions and buffering required to allow all the clients to minimize the effect of long latency.

The VPDMA supports descriptor based mode where lists of descriptors can be set up to configure all the channels as they become available.

#### 12.2.13.1.2 Basic Definitions

##### 12.2.13.1.2.1 Firmware

All the DMA transfers are controlled by List Manager module inside VPDMA. List Manager needs to be loaded with **FIRMWARE**, before any DMA transfer from memory, after the VPDMA reset.

The first MMR write to LIST_ADDR register after VPDMA reset should be the address of the memory buffer(128-bit aligned, that is, last four bits of the buffer address should be zero) where the firmware is stored. List Manager then schedules a DMA transaction to fetch the firmware and sets the list_attr.rdy bit after the firmware loading is complete.

The followingcode is required to be included after the VPDMA clock is enabled in HD-VPSS:

```
*(volatile Uint32*)0x4810D004 = (Uint32)(firmware_buffer_address
      & 0xFFFFFFF0); // Write buffer address of the firmware to register VPDMA-
>List_addr (offset:0x4)
while(*(volatile Uint32*)0x4810D008 != 0x00080000); // Wait for firmware to complete loading.
Wait on bit-field VPDMA->List_attr->rdy (offset: 0x4, bit 19)
```

##### 12.2.13.1.2.2 Client

The clients of the VPDMA are the physical connection to the HD-VPSS processing modules that source or sinks of the data and control when the channel can be updated. Each client can be configured to have specific start event which allows for the channel data to start flowing to or from the external client. The start event can be selected from one of the signals specified by bitfield RD_LB_CLIENT_CTL_STATUS.frame_start or RD_CLIENT_CTL_STATUS.frame_start or WR_CLIENT_CTL_STATUS.frame_start according to the type of the client. MMR of this type are mentioned in the address offset range 0x300 - 0x3f4.

##### 12.2.13.1.2.3 Channel

The VPDMA requires a channel to be setup for each group of transfers. The VPDMA supports four types of channels: YUV, RGB, Miscellaneous and Free. The type of channels is based on the client that the channel can be connected to as each client is expected to provide a specific type of data.All the channels are described through a Data Transfer Descriptor that has a common format. The client that the channel is mapped to interprets the information in the descriptor to perform the requested data transfer.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.1.2.4 Descriptor

The VPDMA needs to be programmed through DESCRIPTORS(A pre-defined structure of eight or four 32-bit words depending on type of descriptors) other than VPDMA Memory Mapped Registers (MMR). Descriptors are of three types:

1. Data Transfer Descriptors - A memory structure used to describe a desired memory transaction to or from a client.
2. Control Descriptors - A memory structure used to perform a control operation inside the DMA controller
3. Configuration Descriptors - A memory structure used to described a setup that should be applied to an processing modules like MMR write, scalar coefficient write etc.

#### 12.2.13.1.2.5 List

A list is a group of descriptors that makes up a set of DMA transfers that need to be completed.

The Regular List is just a single list that the VPDMA will execute each descriptor once and initiate an interrupt when the list has completed. A regular list can contain any kind of descriptor without limitation and be of any size.

The VPDMA Controller works on lists of descriptors. In this mode the processor writes the lists of descriptors in the order it wants them executed. It then writes the location of the list to the LIST_ADDR register, followed by writing the size, type of list and list number to the LIST_ATTR register. The List Manager module will then schedule a DMA transfer to pull in the portion of the list that it can store in internal VPDMA memory. The List Manager will sequentially process the active list of descriptors until either the descriptor requires the use of a client that is currently active or if it is waiting for the next portion of the list to be transferred from a DMA request. If there is no active list to process the list manager will go into an IDLE mode waiting for any client that is blocking a list or a list DMA transfer to complete.

#### 12.2.13.1.2.6 Data Formats Supported

Following are the data formats supported in the VPDMA:

- RGB Data Types:
  - RGB16-565
  - ARGB-1555
  - ARGB-4444
  - RGBA-5551
  - RGBA-4444
  - ARGB24-6666
  - RGB24-888
  - ARGB32-8888
  - RGBA24-6666
  - RGBA32-8888
  - Bitmap-8
  - Bitmap-4 Lower
  - Bitmap-4 Upper
  - Bitmap-2 Offset0
  - Bitmap-2 Offset1
  - Bitmap-2 Offset2
  - Bitmap-2 Offset3
  - Bitmap-1 Offset0
  - Bitmap-1 Offset1
  - Bitmap-1 Offset2
  - Bitmap-1 Offset3


Copyright © 2011–2016, Texas Instruments Incorporated

  - Bitmap-1 Offset4
  - Bitmap-1 Offset5
  - Bitmap-1 Offset6
  - Bitmap-1 Offset7
- YUV Data Types:
  - Y 4:4:4
  - Y 4:2:2
  - Y 4:2:0
  - C 4:4:4
  - C 4:2:2
  - C 4:2:0
  - YC 4:2:2
  - YC4:4:4
  - Cb 4:4:4
  - Cb 4:2:2
  - Cb 4:2:0
  - CbY 4:2:2
  - YC 4:2:2
  - YCb 4:2:2

#### 12.2.13.2 Memory Databus Write Order

The VPDMA L3 master port interfaces with the memory controller via a Quad Word (128 bit) data bus. Words are placed in the Quad Word data bus in little endian order. Table 12-72 shows the beginning or a burst on the quad word data bus and the arrangement of each word address within the burst.

**Table 12-72. Memory Databus Write Order**

| | 127 96 | 95 64 | 63 32 | 31 0 |
|---|---|---|---|---|
| Quadword Addr 0 | Word Addr 0x3 | Word Addr 0x2 | Word Addr 0x1 | Word Addr 0x0 |
| Quadword Addr 1 | Word Addr 0x7 | Word Addr 0x6 | Word Addr 0x5 | Word Addr 0x4 |
| ..... | | | | |
| ..... | | | | |

#### 12.2.13.3 Descriptor

##### *12.2.13.3.1 Data Transfer Descriptors*

In order to set up data transfers from the VPDMA, a data transfer descriptor is added into a list. The fields used for an Inbound and Outbound descriptor vary slightly. An outbound transfer can have two different outbound transfers. The two transfers are the main data transfer and a write of an Inbound Descriptor. The created inbound descriptor is formatted so it can be read back in directly to the next channel specified in the original descriptor.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 12-166. Inbound Data Transfer Descriptor Format**

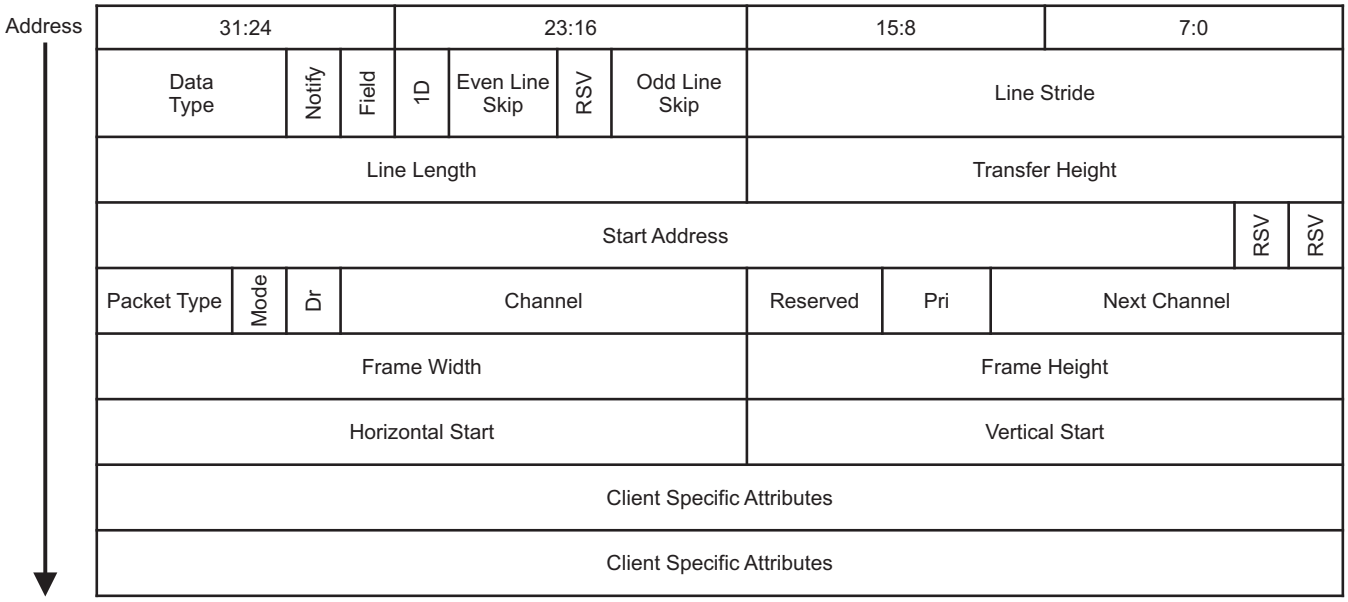
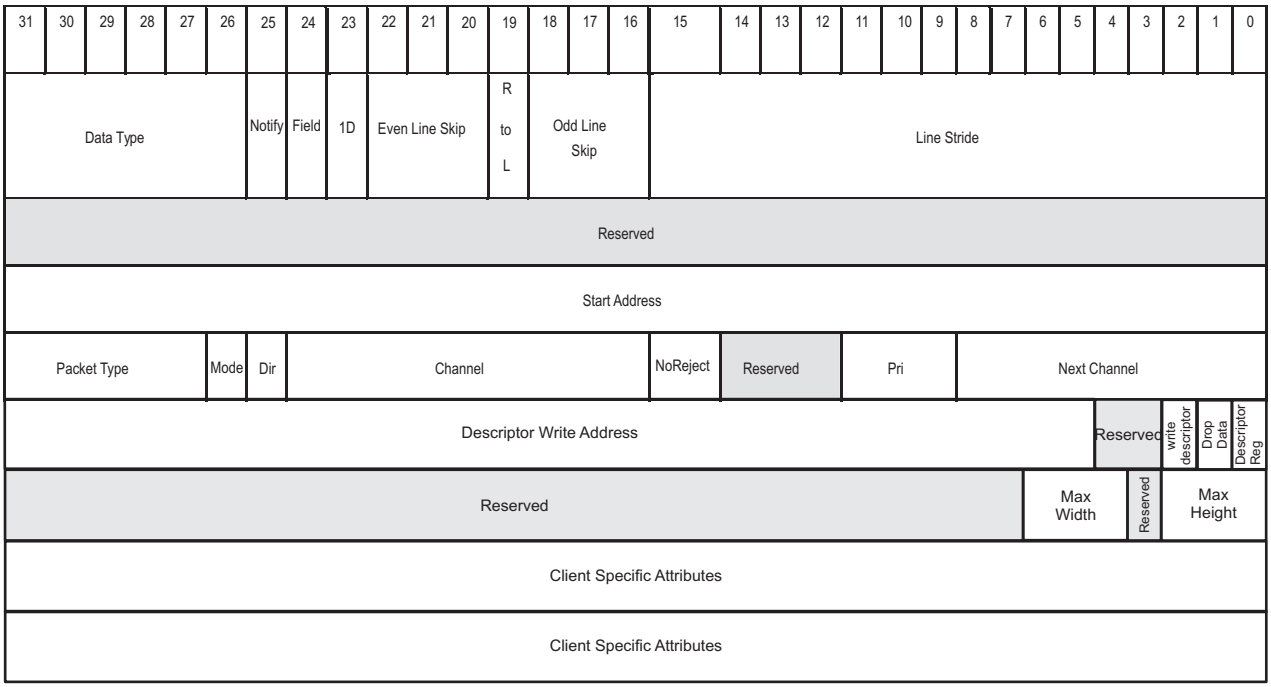



**Figure 12-167. Outbound Data Transfer Descriptor Format**




The general data transfer Descriptor formats can be seen in the above figures. The descriptor consists of 8 32 bit words. A Data Transfer Descriptor will be removed from the list when the resource specified by the Channel field is free. If the Channel is not free when the list reaches a data transfer descriptor then the list will stall until the current transfer on the channel has completed.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.1.1 Data Packet Descriptor Word 0 (Data)

**Table 12-73. Data Packet Descriptor Word 0 Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-26 | Data Type | | Miscellaneous Channel |
| | | | Sets the pixel size in bits plus 1 |
| | | | RGB Channel |
| | | 0 | RGB16-565 |
| | | 1h | ARGB-1555 |
| | | 2h | ARGB-4444 |
| | | 3h | RGBA-5551 |
| | | 4h | RGBA-4444 |
| | | 5h | ARGB24-6666 |
| | | 6h | RGB24-888 |
| | | 7h | ARGB32-8888 |
| | | 8h | RGBA24-6666 |
| | | 9h | RGBA32-8888 |
| | | 20h | Bitmap-8 |
| | | 22h | Bitmap-4 Lower |
| | | 23h | Bitmap-4 Upper |
| | | 24h | Bitmap-2 Offset0 |
| | | 25h | Bitmap-2 Offset1 |
| | | 26h | Bitmap-2 Offset2 |
| | | 27h | Bitmap-2 Offset3 |
| | | 28h | Bitmap-1 Offset0 |
| | | 29h | Bitmap-1 Offset1 |
| | | 2Ah | Bitmap-1 Offset2 |
| | | 2Bh | Bitmap-1 Offset3 |
| | | 2Ch | Bitmap-1 Offset4 |
| | | 2Dh | Bitmap-1 Offset5 |
| | | 2Eh | Bitmap-1 Offset6 |
| | | 2Fh | Bitmap-1 Offset7 |
| | | | YUV Channel |
| | | 0 | Y 4:4:4 |
| | | 1 | Y 4:2:2 |
| | | 2 | Y 4:2:0 |
| | | 4 | C 4:4:4 |
| | | 5 | C 4:2:2 |
| | | 6 | C 4:2:0 |
| | | 7 | YC 4:2:2 |
| | | 8 | YC 4:4:4 |
| | | 14h | Cb 4:4:4 |
| | | 15h | Cb 4:2:2 |
| | | 16h | Cb 4:2:0 |
| | | 17h | CbY 4:2:2 |
| | | 27h | YC 4:2:2 |
| | | 37h | YCb 4:2:2<br>The formats 14h, 15h, 16h, 17h, 27h, 37h are referred to as swap formats in this specification. |
| 25 | Notify | 0-1 | Send List Notification Interrupt upon last transfer of this channel |
| 24 | Field | 0-1 | Field Value |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-73. Data Packet Descriptor Word 0 Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 23 | 1D | 0-1 | The transfer is one dimensional. The transfer and frame sizes are combined to make a single 32-bit transfer size. For writes this value is passed to the generated descriptor. |
| 22-20 | Even Line Skip | | Field Value |
| | | 0 | +1 line |
| | | 1h | +2 lines |
| | | 2h-7h | Reserved |
| 18-16 | Odd Line Skip | | Field Value |
| | | 0 | +1 line |
| | | 1h | +2 lines |
| | | 2h-7h | Reserved |
| 15-0 | Line Stride | 0-FFFFh | Address stride between lines in bytes |

#### *12.2.13.3.1.1.1 Data Type*

Bits 31-26 indicate the type of data that is to be transferred. This value is used to compute the number of bytes per pixel so that transactions may be made for the appropriate amount of data. The types of data are dependent on the type of channel. The VPDMA descriptor data types RGB or YUV are defined associated with the VPDMA channel assignment. This helps the engine to distinguish between the overlapping values of the descriptor data types for RGB and YUV data. For more information about VPDMA channels see Table 12-105.

##### *12.2.13.3.1.1.1.1 Miscellaneous Data Type*

The Data Type selects the size in bits of the data for the Miscellaneous channel. The range is from 0 to 63 to represent the sizes from 1 to 64 bits.

MV and MVSTM clients for DEI and DEI_H use 4 bits for each pixel. Data type 3 needs to be selected for this.

##### *12.2.13.3.1.1.1.2 RGB Data Type*

The most-significant bit selects if the data uses the CLUT table to generate the final RGB data. The data in this case is used as an address to a 256 entry table that generates the final data. If the Bitmap used is less then 8 bits then the lower bits of the data type are used to generate the upper address bits of the lookup table. If the CLUT table is not used then

##### *12.2.13.3.1.1.1.3 YUV Data Type*

The Data Type for a YUV channel determines if the data channel is interleaved or color space separate. If color spaced separate it is still assumed that the two chroma pixels are interleaved.

#### *12.2.13.3.1.1.2 Notify*

The Notify bit is used in conjunction with the Notify interrupts and when the channel has completed the DMA transfer the Notify Interrupt for the list that contains the descriptor will fire. The last Notify bit set for a specific list will be used so only a single Notify should be set in a list.

#### *12.2.13.3.1.1.3 Field*

The Field bit is the field value of the data that will be passed down to the clients. If the data is interlaced, then this value should be cleared to 0.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.1.1.4 1D

This bit is set if a large one dimensional frame needs to be sent to the client. In this case, the stride is ignored and for the write the stride the generated descriptor will always be 0. If this bit is set, then the transfer length and transfer height and frame width and frame height fields are combined to form one 32-bit field with the upper 8 bits reserved and the lower 24 bits being the size of the frame in pixels.

**CAUTION**

Make sure the 1D bit is always cleared to 0 for expected functioning of the VPDMA.

#### 12.2.13.3.1.1.5 Even Line Skip

The Even Line skip is used with Line Stride to generate the next line address on an even line. All frames start on line 0. This value allows for the DMA controller to skip lines for interlaced data in a progressive frame buffer.

#### 12.2.13.3.1.1.6 Odd Line Skip

The Odd Line skip is used with Line Stride to generate the next line address on an odd line. All frames start on line 0, so this will apply starting with the second line. This value allows for the DMA controller to skip lines for interlaced data in a progressive frame buffer.

#### 12.2.13.3.1.1.7 Line Stride

Bits 15-0 are the stride between lines in bytes at the external address. This value is added or subtracted based upon an adjustment using the current skip value. Operation of the external address pointer shall load the Source Address upon start of the transfer, then at the end of each line increment or decrement by the value computed using the Line Stride and Skip value for the line. The line stride must be aligned to an 128-bit bus width i.e The lower 4 bits of the line stride (in hexa decimal) should be zero.

### 12.2.13.3.1.2 Data Packet Descriptor Word 1

**Table 12-74. Data Packet Descriptor Word 1 Field Description**

| Bits | Name | Description |
|---|---|---|
| 31-16 | Line Length | Line Length in Pixels |
| 15-0 | Transfer Height | Number of rows in transfer. |

#### 12.2.13.3.1.2.1 Line Length

Bits 31-16 are the line length in pixels of the current channel. This is ignored for the outbound transfer as the client will provide the line length with the end of line signal on the client interface. The maximum supported line length is currently 4096.

#### 12.2.13.3.1.2.2 Transfer Height

Bits 15-0 are the number of lines to be transferred in the current channel. This is ignored for outbound transfers as the client will provide the line length with the end of frame signal on the client interface. The maximum supported transfer size is currently 2048.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.1.3 Data Packet Descriptor Word 2

**Table 12-75. Data Packet Descriptor Word 2 Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Start Address | | 32-bit data source address [31:0] |
| | | | If Mode is TILED, then TILER specific ADDRESS Map is used: |
| | | | Bits 31-29: |
| | | 0 | 0-degree view |
| | | 1h | 180-degree view + mirroring |
| | | 2h | 0-degree view + mirroring |
| | | 3h | 180-degree view |
| | | 4h | 270-degree view + mirroring |
| | | 5h | 270-degree view |
| | | 6h | 90-degree view |
| | | 7h | 90-degree view + mirroring |
| | | | Bits 28-27: |
| | | 0 | 8-bit container |
| | | 1h | 16-bit container |
| | | 2h | 32-bit container |
| | | 3h | Page Mode |
| | | | If Mode is NORMAL, then bits 31-26 are the upper bits of the address. |

##### 12.2.13.3.1.3.1 Start Address

This is the byte aligned address for the first data transfer. It should be set to the leftmost address fetch if the R to L field is not set in the descriptor. If the R to L field is set then the address for the right most pixel should be specified. The lower bits will be used by the client to determine where to grab the first pixel. The address on the OCP bus will always be word aligned.

#### 12.2.13.3.1.4 Data Packet Descriptor Word 3

**Table 12-76. Data Packet Descriptor Word 3 Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor Type = 0xa |
| 26 | Mode | 0= Normal, 1=TILED |
| 25 | Direction | Inbound = 0, Outbound = 1 |
| 24-16 | Channel | Channel for which this descriptor describes |
| 15 | Reserved | Reserved for future use |
| 11-9 | Priority | Only Bit 9 and Bit 11 are used to set the priority. Bit 10 is ignored.<br>Highest = 0, Lowest = 3<br>By default, hardware assigns priority = 3. This priority level is used in the arbitration between the masters at EMIF level (for DDR access). See Section 12.2.13.3.1.4.5 for more details. |
| 8-0 | Next Channel | Next Channel to execute on a line or the channel to use in the generated write descriptor. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.1.4.1 Packet Type

Bits 31:27 are a unique code which indicates that this Descriptor is a VPDMA descriptor. This value should be 0xA for data transfer descriptor.

#### 12.2.13.3.1.4.2 Mode

Bit 26 is used to indicate if the transfer is to regular memory space or to 2D TILED memory space. The VPDMA will use this to determine if it does standard raster based addressing or if it assumes that the data is stored in TILER format. If data is stored in TILER format then the buffer is turned into 2 or 4 line buffers depending on the container type. For shared clients, such as the ancillary data and the VIP port, that use the memory only one can be active if the mode field is set.

#### 12.2.13.3.1.4.3 Direction

Bit 25 is used to indicate the direction of transfer. This bit indicates that the data flow is from an external source to an internal buffer (inbound) or data transfers form an internal buffer to an external location (outbound).

#### 12.2.13.3.1.4.4 Channel

Bits 24:16 are the Channels which is supported by the descriptor. This is the identification of the specific Channel that is controlled by the contents of the descriptor. The channel assignments can be found in the channel table.

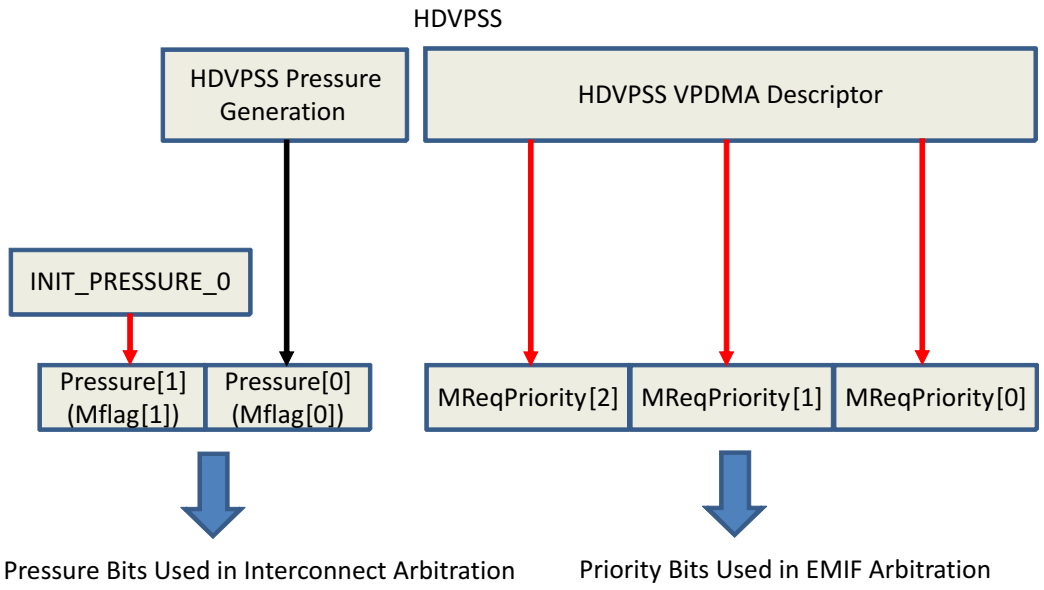
#### 12.2.13.3.1.4.5 Priority

The priority bits (Bits 11, 10 and 9) are used to set the priority at the EMIF level.

VPDMA has two master ports (mst0 and mst1) to transfer the data to external and internal memories.

The priority level for the VPDMA masters can be at two different levels: L3 level (for all the transfers), and EMIF level (only for DDR transfers), as shown in Figure 12-168.

**Figure 12-168. HDVPSS Pressure and Priority Settings**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.1.4.5.1 Priority Level Control at L3 Level

Four priority levels are supported at L3 level for all VPMDA transfers that use two bits (Pressure[1] and Pressure[0], shown in Figure 12-168). Pressure[0] is auto-controlled by VPDMA based on buffer levels. Pressure[1] can be configured in the chip level register: Bit 8 of 0x48140608 is used to set the Pressure[1] bit of the VPDMA master port0 (mst0); Bit 10 of 0x48140608 is used to set the Pressure[1] bit of the VPDMA master port1 (mst1).

It is recommended to use the same priority for both VPDMA masters.

**NOTE:** The higher the value, the higher the priority; that is, a priority level of 3 is prioritized over a priority level of 2.

#### 12.2.13.3.1.4.5.2 Priority Level Control at EMIF Level

Eight levels of priority can be set at the EMIF level using three bits (MReqPriority[2:0]).

**Table 12-77. Priority Bit Fields**

| Priority[11] | Priority[10] | Priority[9] | Priority [11:9] | Priority at EMIF |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 2 | 2 |
| 0 | 1 | 1 | 3 | 3 |
| 1 | 0 | 0 | 4 | 4 |
| 1 | 0 | 1 | 5 | 5 |
| 1 | 1 | 0 | 6 | 6 |
| 1 | 1 | 1 | 7 | 7 |

As shown in Table 12-77, Bit 10 of priority bit-field is ignored. Bits 9 and 11 are used to set the priority at EMIF level.

**NOTE:** The higher the value, the lower the priority; that is, a priority level of 2 is prioritized over a priority level of 3.

#### 12.2.13.3.1.4.6 Next Channel

Bits 8:0 give the next channel to use. This value MUST be the same as the channel.

### 12.2.13.3.1.5 Data Packet Descriptor Word 4

#### 12.2.13.3.1.5.1 Inbound data

**Table 12-78. Data Packet Descriptor Word 4 Inbound Data Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 31-16 | Frame Width | Width of the client frame. |
| 15-0 | Frame Height | Height of the client frame |

##### 12.2.13.3.1.5.1.1 Frame Width

Bits 31:16 indicate the width in pixels of the frame. This is the width of the entire frame and not just the width of the data represented by the current channel unless this channel fills the entire frame. This allows for the VPDMA to present a larger frame of data to the client while only fetching the required data. The currently supported maximum width is 4096.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.1.5.1.2 Frame Height

Bits 15:0 indicate the height in pixels of the entire frame. This is the height of the entire frame and not just the height of the data represented by the current channel unless this channel fills the entire frame. This allows for the VPDMA to present a larger frame of data to the client while only fetching the required data. The currently supported maximum height is 2048.

#### 12.2.13.3.1.5.2 Outbound data

**Table 12-79. Data Packet Descriptor Word 4 Outbound Data Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 31-5 | Descriptor Write Address | The 32 byte aligned location to write an inbound descriptor |
| 2 | Write Descriptor | If set to 1, a descriptor will be generated when the client completes a frame. |
| 1 | Drop Data | If set to 1, the data will not be written out. Also, if this is set, the write descriptor bit MUST be set.This allows for descriptors to only be written out so that software can determine the size of the required buffer before data is written out. |
| 0 | Use Descriptor Register | If set to 1, the CURRENT_DESCRIPTOR register will be used for the write address location. If set to 0, the Descriptor Write Address field in this word will be used for the Descriptor Write Address. |

#### 12.2.13.3.1.5.2.1 Descriptor Write Address

Bits 31:5 set the 32 byte address to write the generated descriptor. This address is only used if the Write Descriptor bit is set to 1 and the Use Descriptor Register bit is set to 0. If this case is met when the channel is complete a descriptor that meets the inbound descriptor format will be written to the location specified by this field. This allows for software to have a specific channel write its descriptor to a specific address no matter what order the channel completes compared to other outbound channels.

#### 12.2.13.3.1.5.2.2 Write Descriptor

Bit 2 determines if a descriptor should be written out when the client is completed. If this bit is set the descriptor will be written. The format of the descriptor will be of an Inbound Data Transfer descriptor with the LINE LENGTH and TRANSFER HEIGHT fields determined by the counters in the clients. This means that the direction bit will be the opposite of the inbound descriptor. The FRAME WIDTH and FRAME HEIGHT values will match the LINE LENGTH and TRANSFER HEIGHT fields. The CHANNEL and NEXT CHANNEL fields will match the NEXT CHANNEL field of the original descriptor. The FIELD bit will match the source field captured by the client. The NOTIFY bit will always be 0. All other fields will match the original outbound descriptor. If the Outbound descriptor MODE is set to TILED data then the descriptor address used will be in TILED data space and software must ensure that the address is in page mode for the address of the descriptor.

#### 12.2.13.3.1.5.2.3 Drop Data

Bit 1 determines if the data should be written out or not. In some cases software might only want to write the descriptor out to determine the size of the buffer that will be required to store incoming data. By setting this bit,no data will be written out. If this bit is set then the Write Descriptor bit MUST be set.

#### 12.2.13.3.1.5.2.4 Use Descriptor Register

Bit 0 determines where the descriptor should be written. This bit MUST be set to 0.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.1.6 Data Packet Descriptor Word 5

##### 12.2.13.3.1.6.1 Inbound data

**Table 12-80. Data Packet Descriptor Word 5 Inbound Data Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 31-16 | Horizontal Start | Data field start location for channel data insertion into frame/field. |
| 15-0 | Vertical Start | Horizontal displacement to start of window within the frame. |

###### 12.2.13.3.1.6.1.1 Horizontal Start

Bits 31:16 are the horizontal start location for the channel data. This value is used to control where the data starts for a channel so that blank segments may be inserted into the row. This is used for region based graphics. This field is reserved for outbound data. For non-graphics clients this MUST be set to 0.

###### 12.2.13.3.1.6.1.2 Vertical Start

Bits 15:0 are the vertical start location for the channel which allows the channel to be spaced vertically from the previous channel or the top of the frame. This is used for region-based graphics. This field is reserved for outbound data descriptors. For non-graphics clients, this MUST be set to 0.

##### 12.2.13.3.1.6.2 Outbound data

**Table 12-81. Data Packet Descriptor Word 5 Outbound Data Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 6-4 | Max Width | The maximum allowable pixels per line. 0: Unlimited Line Size 4: 352 pixels 5: 768 pixels 6: 1280 pixels 7: 1920 pixels Others: Reserved |
| 2-0 | Max Height | The maximum allowable lines per frame 0: Unlimited Frame Size 4: 288 lines 5: 576 lines 6: 720 lines 7: 1080 lines Others: Reserved |

###### 12.2.13.3.1.6.2.1 Max Width

Bits 6:4 are encoded to set the maximum transferred line size. If the field is not set to unlimited if the client sending data into the VPDMA exceeds the maximum allowed pixels the data will continue to be received from the client but will not be sent to external memory until an end of line is received from the client sending data. The outbound descriptor if created will still have the transmitted size of the last line of the frame which will match the max width. If the frame does not exceed the max width then the actual transmitted size will be placed in the descriptor. Tiled clients such as the noise filter should always set this to 0. If Max Width is 0 and the line size is larger then 4096 pixels then the address counters will overflow and the data at the start of the line will be overwritten.

###### 12.2.13.3.1.6.2.2 Max Height

Bits 2:0 are encoded to set the maximum transferred frame size. If the field is not set to unlimited if the client sending data into the VPDMA exceeds the maximum allowed lines the data will continue to be received from the client but will not be sent to external memory until an end of frame is received from the client sending data. The outbound descriptor if created will still have the transmitted number of lines received which would match the max height. If the frame does not exceed the max height then the actual transmitted size will be placed in the descriptor. . Tiled clients such as the noise filter should always set this to 0.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.1.7 Data Packet Descriptor Word 6/7 (Data)

The words 4/5 give a 64 bit of configuration that can be passed specifically to the IP that supports it. This is passed directly down to the module through a VPI Control port. Please see the section on the specific clients for the format of this data.

### 12.2.13.3.2 Configuration Descriptor

The Configuration Descriptor is used in a List to setup a client configuration. The configuration descriptor consists of a header and a payload portion. The payload can either be part of the list that the descriptor is contained in or it can be at a separate location. The VPDMA will pass the payload of the configuration descriptor down to the client over the VPI Control port interface or through the MMR configuration port depending on the destination field. A Configuration Descriptor to any single destination except Destination 0 MUST be on a single list. Configuration Descriptors to different destinations may be on different lists.

The Configuration Descriptor Header is 4 32 bit words. A configuration descriptor is not consumed until the destination specified has received the entire configuration payload. Therefore the list is expected to stall while the configuration takes place. This ensures that all descriptors after a configuration descriptor in the list will occur after the desired configuration.

#### 12.2.13.3.2.1 Configuration Descriptor Header Word0

**Table 12-82. Configuration Descriptor Header Word0 Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 31-0 | Address Offset of the Destination | This is the address offset location in the destination that the block of data in the payload should be written. This field is only used if the descriptor Class is a block type. Otherwise this field is reserved. |

#### 12.2.13.3.2.2 Configuration Descriptor Header Word1

**Table 12-83. Configuration Descriptor Header Word1 Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 15-0 | Number of Data Words | Length of First Data Packet for Class 1(block). |

##### 12.2.13.3.2.2.1 Number of Data Words

Bits 15-0 indicate the length of the first data block if the class is 1 as specified in Configuration Descriptor Header Word3. If the class is not 1 then this field should be set to 0.

#### 12.2.13.3.2.3 Configuration Descriptor Header Word2

**Table 12-84. Configuration Descriptor Header Word2 Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 31-0 | Payload Location | Pointer to the data payload |

##### 12.2.13.3.2.3.1 Payload Location

Bits 31-0 contain the pointer to the data payload if the command packet is an indirect type. This value along with the Payload Length will be combined to issue a DMA transaction that must complete before the List Manager can finish processing this descriptor. The list will be made inactive pending this DMA completion. This address should be on a 16 byte boundary so the lower 4 bits should always be 0.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.2.4 Configuration Descriptor Header Word3

**Table 12-85. Configuration Descriptor Header Word3 Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor Type = 0xb |
| 26 | Direct | 0 = Indirect Command<br>1 = Direct Command |
| 25-24 | Class | 0 = Address, Data Set<br>1 = Block 2,3 Reserved for Future Use |
| 23-16 | Destination | Destination of the configuration payload |
| 15-0 | Payload Length | Length of Payload in Words. |

##### 12.2.13.3.2.4.1 Packet Type

Bits 31-27, this field indicates the type of descriptor and should be set 0xB for configuration descriptor.

##### 12.2.13.3.2.4.2 Direct

Bit 26, this field indicates that the descriptor is a direct command or indirect command. A direct command means that the payload is contiguous with the descriptor and will be pulled in by the list manager descriptor control as it processes the list. A direct payload must end on or before a 256 byte boundary in the list. An indirect command is when the Address is located in Word2 is a pointer to the data payload. A DMA transaction will be scheduled to bring the payload into the List Manager to allow for the payload to be sent to the destination.

##### 12.2.13.3.2.4.3 Class

Bits 25-24, this field indicates the type of payload is associated with command descriptor, and thus how to handle the payload.

A 0 indicates that the payload consists of blocks of data with each block having an address and length. The first word after a sub-block contains the next address and length in bytes. This allows for writing to multiple configuration regions in a client. The Configuration Descriptor word1 is the first address and length pair. All addresses used must be larger then the previous address for all destinations except for the MMR destination. If a sub block length is not evenly divisible into 16 then zeroes should be padded in the payload so that the Next Client Address and Length field start on a 16 byte boundary.

###### 12.2.13.3.2.4.3.1 Address Data Block Format

**Table 12-86. Address Data Block Format Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 31-0 | | Next Client Address |
| 31-0 | | Configuration for Next Client Address |
| 31-0 | | Configuration for Next Client Address + 4 |
| 31-0 | | Configuration for Next Client Address + 8 |
| 31-0 | | Configuration for Next Client Address + 12 |
| 31-0 | | Configuration for Next Client Address + 16 |
| 31-0 | | Next Client Address 2 |
| 15-0 | | Sub Block Length |

A 1 indicates the payload is simply block data. The data is contiguous and starts at the offset of the destination as specified in word1.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.2.4.4 Destination

The Destination field is used to determine where the configuration payload should be sent. The values for the destination field can be seen in the table below.

**Table 12-87. Destination Field Description**

| Destination Value | Actual Destination | Description |
|---|---|---|
| 0 | mmr_client | Write to MMR registers to setup other modules |
| 1 | grpx1 | Graphics 0 Frame Configuration and Scaler Coefficients |
| 2 | grpx2 | Graphics 1 Frame Configuration and Scaler Coefficients |
| 3 | grpx3 | Graphics 2 Frame Configuration and Scaler Coefficients |
| 4 | dei | DEI Scaler Coefficient Tables |
| 5 | mq | Mid Quality DEI Scaler Coefficient Tables |
| 6 | sc_m_wrbk | Mid Quality Writeback Scaler Coefficient Tables |
| 7 | vip1 | VIP0 Scaler Coefficient Tables |
| 8 | vip2 | VIP1 Scaler Coefficient Tables |

#### 12.2.13.3.2.4.5 Payload Length

Bits 15-0, this field indicates the size of the payload in words for the command. This is 128 bit words. The maximum number of words is 1023 words in a single payload.

### 12.2.13.3.3 Control Descriptor

#### 12.2.13.3.3.1 Generic Control Descriptor Format

A control descriptor is the mechanism that is used in a list to give commands to the List Manager. These descriptors are not considered consumed until the List Manager has done the instruction required by the descriptor. All control descriptors are four 32bit words long. All control descriptors have a common Header located at Word 3. But, Word 0, Word 1 and Word 2 are based on the specific control descriptor.

#### 12.2.13.3.3.2 Control Descriptor Header Description

**Table 12-88. Control Descriptor Header Description**

| Bits | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor Type = 0xc |
| 26-25 | Reserved | Reserved |
| 24-16 | Source | Specifies the source used for the control event. |
| 15-4 | Reserved | Reserved for future use |
| 3-0 | Control | The type of control descriptor that should be run by the List Manager |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.3.2.1 Packet Type

Bits 31-27, this field indicates the type of descriptor and should be set 0xB for configuration descriptor.

#### 12.2.13.3.3.2.2 Source

The source is combined with the control field to determine what the control descriptor should use as the source for its synchronization event.

#### 12.2.13.3.3.2.3 Control

The Control field defines the specific function of the descriptor. The table below lists the different control descriptors.

### 12.2.13.3.3.3 Control Descriptor Types

**Table 12-89. Control Descriptor Types Summary**

| Control Descriptor | Control Field Value | Description |
|---|---|---|
| Sync on Client | 0 | Wait for the client attached to the channel specified to reach a certain event before proceeding. |
| Sync on List | 1h | Wait for another list(s) to reach its Sync on List Descriptor |
| Sync on External Event | 2h | Wait for a Register write to the LIST_STAT_SYNC bit specified or for an external event. |
| Sync on Channel | 4h | Wait for the channel specified to complete |
| Change Client Interrupt | 5h | Change the interrupt event for a client interrupt but do not wait for the event to occur. |
| Send Interrupt | 6h | Generate an interrupt event on one of the Control Descriptor Interrupts |
| Reload List | 7h | Reload the list from the location and size specified. |
| Abort Channel | 8h | Abort the transfer in the channel specified. |
| Toggle LM Field | 9h | Causes the internal field signal that can be selected by each client to toggle. This can be used so that a group of clients can all start simultaneously and not be tied to one of the specific output ports. |

#### 12.2.13.3.3.3.1 Sync on Client

A Sync on Client Control descriptor uses the source field to select a channel to modify the attached clients interrupt generation event. After configuring the interrupt generation event the list will then stall until that event has occurred. Word 0 and Word 2 are reserved.

**Table 12-90. Sync on Client Field Descriptions (Word - 1)**

| Bits | Name | Description |
|---|---|---|
| 31-16 | PIXEL_COUNT | Specify the pixel position on a line specified by LINE_COUNT where a pixel based event would occur. |
| 15-0 | LINE_COUNT | Specify the line where a line based event would trigger |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-91. Sync on Client Field Descriptions (Word - 3)**

| Bits | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor Type = Ch |
| 26-24 | Reserved | Reserved |
| 23-16 | Source | Specifies the source used for the control event. |
| 15-4 | Reserved | Reserved for future use |
| 3-0 | Control | Control type = 0 |

#### *12.2.13.3.3.3.2 Sync on List*

The Sync on List Control descriptor is a mechanism to ensure that multiple lists have all reached a common point. The Sync on List descriptor uses the Source field as a mask for each list that should be synchronized which must include the list where the control descriptor resides. Once all lists specified in the source field have reached their Sync on List descriptor all lists will resume with the lowest list number resuming first. All Sync on List Control descriptors in each of the lists must have the same source field so that all lists will synchronize upon reaching that point.

If it is desired to synchronize list 0 and list 1 then the source field would be 0x3. If it is desired to synchronize list 1, list 3, and list 4 then the source field would be 0x1a. Word 0, Word 1 and Word 2 are reserved.

**Table 12-92. Sync on List Field Descriptions (Word - 3)**

| Bits | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor Type = Ch |
| 26-24 | Reserved | Reserved |
| 23-16 | Source | Specifies the source used for the control event. |
| 15-4 | Reserved | Reserved for future use |
| 3-0 | Control | Control type = 1h |

#### *12.2.13.3.3.3.3 Sync on External Event*

A Sync on External Event descriptor ensures an external event has occurred before proceeding. The external event can be a write to a bit of the LIST_STAT_SYNC register or other external events that are determined at VPDMA elaboration to have occurred. This descriptor can be used to allow for external software to control progression of the list. The descriptor can also be used to select external signals that are brought into the VPDMA. Source values 0-7 are for the respective bits in the LIST_STAT_SYNC register. The processing of a Sync on External Event descriptor will result in a handshake the clears that bit of the LIST_STAT_SYNC register. Word 0, Word 1 and Word 2 are reserved.

**Table 12-93. Sync on External Event Field Descriptions (Word - 3)**

| Bits | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor Type = Ch |
| 26-24 | Reserved | Reserved |
| 23-16 | Source | Specifies the source used for the control event. |
| 15-4 | Reserved | Reserved for future use |
| 3-0 | Control | Control type = 2h |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.3.3.4 Sync on Channel

A Sync on Channel descriptor stalls the list until the channel specified is free. If the channel is already free then the descriptor will not cause the list to stall. Word 0, Word 1 and Word 2 are reserved.

**Table 12-94. Sync on Channel Field Descriptions (Word - 3)**

| Bits | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor Type = Ch |
| 26-24 | Reserved | Reserved |
| 23-16 | Source | Specifies the source used for the control event. |
| 15-4 | Reserved | Reserved for future use |
| 3-0 | Control | Control type = 4h |

#### 12.2.13.3.3.3.5 Change Client Interrupt

A Change Channel Interrupt Source descriptor uses the source field to select a channel to modify the attached clients interrupt generation event. The list will not stall on this descriptor. The format of the fields is identical to the Sync on Client Descriptor. Word 0 is reserved.

**Table 12-95. Change Client Interrupt Field Descriptions (Word - 1)**

| Bits | Name | Description |
|---|---|---|
| 31-16 | PIXEL_COUNT | Specify the pixel position on a line specified by LINE_COUNT where a pixel based event would occur. |
| 15-0 | LINE_COUNT | Specify the line where a line based event would trigger. |

**Table 12-96. Change Client Interrupt Field Descriptions (Word - 2)**

| Bits | Name | Description |
|---|---|---|
| 31-4 | Reserved | Reserved |
| 3-0 | Event | Specify the event which should trigger the client interrupt. |

**Table 12-97. Change Client Interrupt Field Descriptions (Word - 3)**

| Bits | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor Type = Ch |
| 26-24 | Reserved | Reserved |
| 23-16 | Source | Specifies the source used for the control event. |
| 15-4 | Reserved | Reserved for future use |
| 3-0 | Control | Control type = 5h |

#### 12.2.13.3.3.3.6 Send Interrupt

A Send Interrupt descriptor will cause the VPDMA to generate an interrupt on the list manager controlled interrupts as specified by the Source Field. The list will not stall on this descriptor. For example if source is 0 then vpdma_descriptor_interrupt0 will fire. If source is 12 then vpdma_descriptor_interrupt12 will fire. Word 0, Word 1 and Word 2 are reserved.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-98. Send Interrupt Field Descriptions (Word - 3)**

| Bits | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor Type = Ch |
| 26-24 | Reserved | Reserved |
| 23-16 | Source | Specifies the source used for the control event. |
| 15-4 | Reserved | Reserved for future use |
| 3-0 | Control | Control type = 6h |

#### *12.2.13.3.3.3.7 Reload List*

A Reload List descriptor causes ending descriptors after this descriptor in the original list to be dropped and a new list at the location and of the size specified in the descriptor. This descriptor can be used to allow for linked lists in the VPDMA as the list will continue from this point and fetch the list specified and it does not have to be continuous with the current list. The NULL bit in the first word determines if it is a valid address. If the NULL bit is set then the list will not fetch that address until a doorbell is received. Upon receiving the doorbell the reload list descriptor will be refetched from memory and then the new list address will be used along with list size to fetch the next section of the list. This allows for the list to be dynamically adding new sections of a list.

The procedure to use a reload list descriptor as a mechanism to allow dynamic additions to a list is as follows. A list section is created and the last descriptor in the section is a reload list with a NULL bit set. When a new section of the list is created the last descriptor in the list will be a Reload List with the NULL bit set. The software then updates the address and size in the Reload List descriptor in the previous section and then writes to the LIST_ATTRIBUTE register with the list number and the list type set to doorbell. This will create an immediate pulse that the hardware will latch until it does another list fetch. Software can do this as many times as it wants to keep adding to the list. Word 2 is reserved.

**Table 12-99. Reload List Field Descriptions (Word - 0)**

| Bits | Name | Description |
|---|---|---|
| 31-1 | LIST_ADDRESS | 31 most-significant Bits of the memory address where the descriptors to be loaded are stored. Address must be 16 byte aligned. |
| 0 | NULL | Specify whether “doorbell” SW trigger is used. |

**Table 12-100. Reload List Field Descriptions (Word - 1)**

| Bits | Name | Description |
|---|---|---|
| 31-16 | Reserved | Reserved |
| 15-0 | LIST_SIZE | Size of the list to load |

**Table 12-101. Reload List Field Descriptions (Word - 3)**

| Bits | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor Type = Ch |
| 26-4 | Reserved | Reserved |
| 3-0 | Control | Control type = 7h |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.3.3.3.8 Abort Channel

An Abort Channel descriptor is used to clear a channel. This clears the channel from issuing any more requests. Any outstanding requests for that channel will complete as originally scheduled. All data inside the client will be flushed. For Tiled Clients such as the noise filter it is required to issue two consecutive abort channel descriptors to ensure the channel is aborted for both the current tile and next tile. Word 0, Word 1 and Word 2 are reserved.

**Table 12-102. Abort Channel Field Descriptions (Word - 3)**

| Bit | Name | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor type = 0xC |
| 26-24 | Reserved | Reserved |
| 23-16 | Source | VPDMA Channel Number whose transfers are to be aborted |
| 15-4 | Reserved | Reserved |
| 3-0 | Control | Control type = 9h |

#### 12.2.13.3.3.3.9 Toggle LM Field

A Toggle LM Field descriptor is used if the clients set their frame source to LM FID. The read clients will start transmitting data upon the FID signal inside the LM changing value. This descriptor will cause the LM to toggle the value of the internal FIDs. The descriptor allows to either toggle the existing value of the FID or to set it. Note that clients wait for changes in FID value to generate the new frame so if the value is set to the current value no event will occur for the client. Word 0 and Word 2 are reserved.

**Table 12-103. Toggle LM Field Field Descriptions (Word - 2)**

| Bit | Name | Description |
|---|---|---|
| 31-6 | Reserved | Reserved |
| 5-4 | FID2 CTL | This field controls the behavior of FID2<br>0h: FID2 remains unchanged<br>1h: FID2 toggles<br>2h: FID2 becomes 0<br>3h: FID2 becomes 1 |
| 3-2 | FID1 CTL | This field controls the behavior of FID1<br>0h: FID1 remains unchanged<br>1h: FID1 toggles<br>2h: FID1 becomes 0<br>3h: FID1 becomes 1 |
| 1-0 | FID0 CTL | This field controls the behavior of FID0<br>0h: FID0 remains unchanged<br>1h: FID0 toggles<br>2h: FID0 becomes 0<br>3h: FID0 becomes 1 |

**Table 12-104. Toggle LM Field Field Descriptions (Word - 3)**

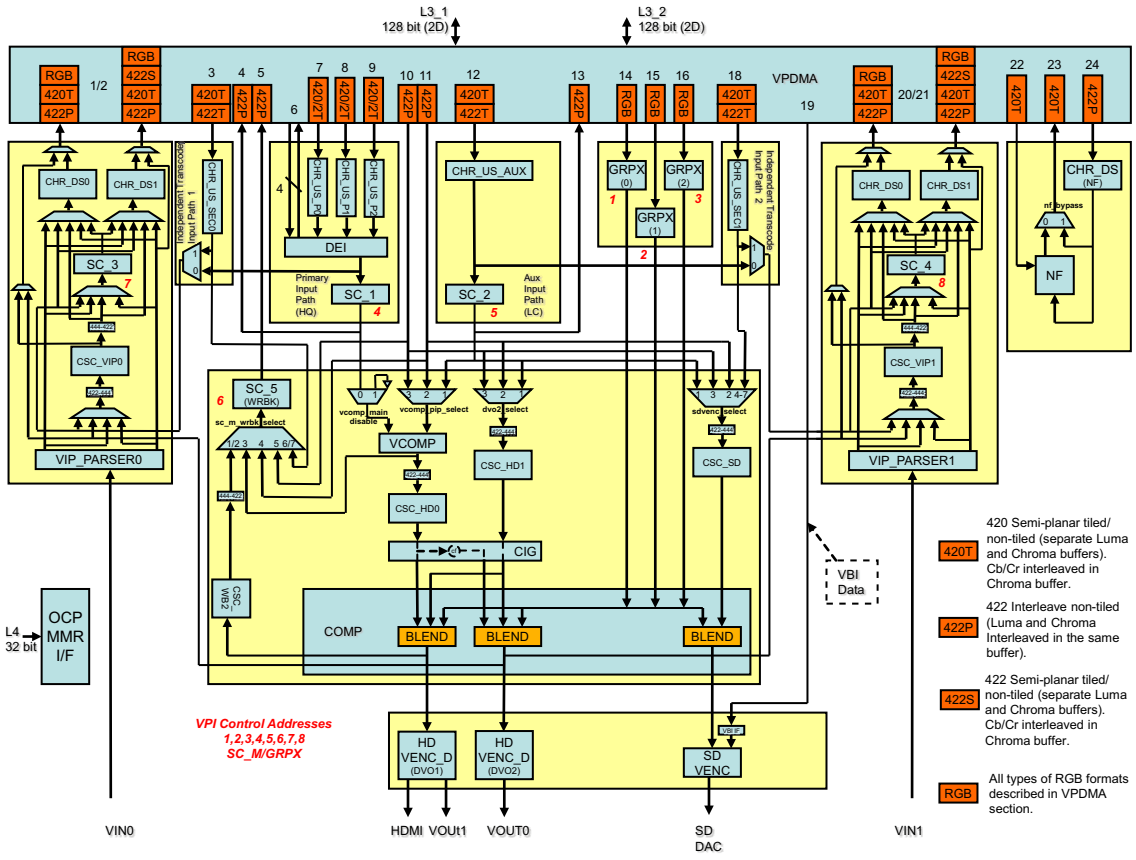
| Bit | Field | Description |
|---|---|---|
| 31-27 | Packet Type | Host Packet Descriptor type = 0xC |
| 26-4 | Reserved | This field controls the behavior of FID2 |
| 3-0 | Control | Control type = 9h |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.13.4 VPDMA and HDVPSS

The HDVPSS is shown in Figure 12-169 with each of the VPDMA clients numbered at the top.

**Figure 12-169. Block Diagram**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.4.1 VPDMA Channels in HDVPSS

Table 12-105 lists all of the channels in the VPDMA and its base attributes. The Data Type column states what type of data YUV,RGB or OTHER the channel handles and in parentheses are the legal data type values that can be entered into a data transfer descriptor. The Client field states the name of the Client and in parentheses it states the reference number shown in Figure 12-169.

**Table 12-105. VPDMA Channels**

| Channel | Description | Channel Number | Data Type | Client |
|---|---|---|---|---|
| dei_vid1_luma | DEI Video 420 Luma Data/ 422 Interleaved Data | 0 | YUV (0x1, 0x2, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | dei_1_luma (7) |
| dei_vid1_chroma | DEI Video 420 Chroma Data | 1 | YUV (0x5, 0x6, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | dei_1_chroma (7) |
| dei_vid2_luma | DEI Field Minus 1420 Luma Data | 2 | YUV (0x1, 0x2, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | dei_2_luma (8) |
| dei_vid2_chroma | DEI Field Minus 1 420 Chroma Data | 3 | YUV (0x5, 0x6, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | dei_2_chroma (8) |
| dei_vid3_luma | DEI Field Minus 2 420 Luma Data | 4 | YUV (0x1, 0x2, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | dei_3_luma (9) |
| dei_vid3_chroma | DEI Field Minus 2 420 Chroma Data | 5 | YUV (0x5, 0x6, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | dei_3_chroma (9) |
| dei_mv | DEI Motion Vector | 12 | OTHER (4) | dei_mv_in (6) |
| dei_mv_out | DEI Motion Vector Write | 15 | OTHER (4) | dei_mv_out (6) |
| dei_scaler | DEI Scaler Write to Memory | 17 | YUV (0x7) | dei_sc_out (4) |
| scaler_luma | Scaler In Luma 420/422 Data | 18 | YUV (0x1, 0x2, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | sc_in_luma (12) |
| scaler_chroma | Scaler in Chroma 420/422 Data | 19 | YUV (0x5, 0x6, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | sc_in_chroma (12) |
| scaler_out | Aux Scaler Write to Memory | 28 | YUV (0x7) | sc_out (13) |
| grpx1 | Graphics 0 Data | 29 | RGB (0x0-0x9, 0x20, 0x22-0x2f) | grpx1_data (14) |
| grpx2 | Graphics 1 Data | 30 | RGB (0x0-0x9, 0x20, 0x22-0x2f) | grpx2_data (15) |
| grpx3 | Graphcis 2 Data | 31 | RGB (0x0-0x9, 0x20, 0x22-0x2f) | grpx3_data (16) |
| grpx1_stencil | Graphics 0 Stencil | 32 | OTHER (1) | grpx1_st (2) |
| grpx2_stencil | Graphics 1 Stencil | 33 | OTHER (1) | grpx2_st (2) |
| grpx3_stencil | Graphics 2 Stencil | 34 | OTHER (1) | grpx3_st (2) |
| grpx1_clut | Graphics 0 Color Lookup Table Load from Memory | 35 | OTHER () | grpx1_clut_clt (2) |
| grpx2_clut | Graphics 1 Color Lookup Table Load from Memory | 36 | OTHER () | grpx2_clut_clt (2) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-105. VPDMA Channels (continued)**

| Channel | Description | Channel Number | Data Type | Client |
|---|---|---|---|---|
| grpx3_clut | Graphics 2 Color Lookup Table Load from Memory | 37 | OTHER () | grpx3_clut_clt (2) |
| vip1_mult_porta_src0 | Video Input 1 Port A Channel 0 | 38 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src1 | Video Input 1 Port A Channel 1 | 39 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src2 | Video Input 1 Port A Channel 2 | 40 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src3 | Video Input 1 Port A Channel 3 | 41 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src4 | Video Input 1 Port A Channel 4 | 42 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src5 | Video Input 1 Port A Channel 5 | 43 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src6 | Video Input 1 Port A Channel 6 | 44 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src7 | Video Input 1 Port A Channel 7 | 45 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src8 | Video Input 1 Port A Channel 8 | 46 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src9 | Video Input 1 Port A Channel 9 | 47 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src10 | Video Input 1 Port A Channel 10 | 48 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src11 | Video Input 1 Port A Channel 11 | 49 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src12 | Video Input 1 Port A Channel 12 | 50 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src13 | Video Input 1 Port A Channel 13 | 51 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src14 | Video Input 1 Port A Channel 14 | 52 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_porta_src15 | Video Input 1 Port A Channel 15 | 53 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_mult_portb_src0 | Video Input 1 Port B Channel 0 | 54 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src1 | Video Input 1 Port B Channel 1 | 55 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src2 | Video Input 1 Port B Channel 2 | 56 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src3 | Video Input 1 Port B Channel 3 | 57 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src4 | Video Input 1 Port B Channel 4 | 58 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src5 | Video Input 1 Port B Channel 5 | 59 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src6 | Video Input 1 Port B Channel 6 | 60 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src7 | Video Input 1 Port B Channel 7 | 61 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src8 | Video Input 1 Port B Channel 8 | 62 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src9 | Video Input 1 Port B Channel 9 | 63 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src10 | Video Input 1 Port B Channel 10 | 64 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-105. VPDMA Channels (continued)**

| Channel | Description | Channel Number | Data Type | Client |
|---|---|---|---|---|
| vip1_mult_portb_src11 | Video Input 1 Port B Channel 11 | 65 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src12 | Video Input 1 Port B Channel 12 | 66 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src13 | Video Input 1 Port B Channel 13 | 67 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src14 | Video Input 1 Port B Channel 14 | 68 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_portb_src15 | Video Input 1 Port B Channel 15 | 69 | YUV (0x7, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_mult_anca_src0 | Video Input 1 Port A Ancillary Data Channel 0 | 70 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src1 | Video Input 1 Port A Ancillary Data Channel 1 | 71 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src2 | Video Input 1 Port A Ancillary Data Channel 2 | 72 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src3 | Video Input 1 Port A Ancillary Data Channel 3 | 73 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src4 | Video Input 1 Port A Ancillary Data Channel 4 | 74 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src5 | Video Input 1 Port A Ancillary Data Channel 5 | 75 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src6 | Video Input 1 Port A Ancillary Data Channel 6 | 76 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src7 | Video Input 1 Port A Ancillary Data Channel 7 | 77 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src8 | Video Input 1 Port A Ancillary Data Channel 8 | 78 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src9 | Video Input 1 Port A Ancillary Data Channel 9 | 79 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src10 | Video Input 1 Port A Ancillary Data Channel 10 | 80 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src11 | Video Input 1 Port A Ancillary Data Channel 11 | 81 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src12 | Video Input 1 Port A Ancillary Data Channel 12 | 82 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src13 | Video Input 1 Port A Ancillary Data Channel 13 | 83 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src14 | Video Input 1 Port A Ancillary Data Channel 14 | 84 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_anca_src15 | Video Input 1 Port A Ancillary Data Channel 15 | 85 | OTHER (8) | vip1_anc_a (2) |
| vip1_mult_ancb_src0 | Video Input 1 Port B Ancillary Data Channel 0 | 86 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src1 | Video Input 1 Port B Ancillary Data Channel 1 | 87 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src2 | Video Input 1 Port B Ancillary Data Channel 2 | 88 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src3 | Video Input 1 Port B Ancillary Data Channel 3 | 89 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src4 | Video Input 1 Port B Ancillary Data Channel 4 | 90 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src5 | Video Input 1 Port B Ancillary Data Channel 5 | 91 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src6 | Video Input 1 Port B Ancillary Data Channel 6 | 92 | OTHER (8) | vip1_anc_b (2) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-105. VPDMA Channels (continued)**

| Channel | Description | Channel Number | Data Type | Client |
|---|---|---|---|---|
| vip1_mult_ancb_src7 | Video Input 1 Port B Ancillary Data Channel 7 | 93 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src8 | Video Input 1 Port B Ancillary Data Channel 8 | 94 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src9 | Video Input 1 Port B Ancillary Data Channel 9 | 95 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src10 | Video Input 1 Port B Ancillary Data Channel 10 | 96 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src11 | Video Input 1 Port B Ancillary Data Channel 11 | 97 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src12 | Video Input 1 Port B Ancillary Data Channel 12 | 98 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src13 | Video Input 1 Port B Ancillary Data Channel 13 | 99 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src14 | Video Input 1 Port B Ancillary Data Channel 14 | 100 | OTHER (8) | vip1_anc_b (2) |
| vip1_mult_ancb_src15 | Video Input 1 Port B Ancillary Data Channel 15 | 101 | OTHER (8) | vip1_anc_b (2) |
| vip1_porta_luma | Video Input 1 Port A 420 Data Luma | 102 | YUV (0x1, 0x2,0x7, 0x17, 0x27, 0x37) | vip1_up_y (1) |
| vip1_porta_chroma | Video Input 1 Port A 420 Data Chroma | 103 | YUV (0x5, 0x6, 0x7, 0x15, 0x16, 0x17, 0x27, 0x37) | vip1_up_uv (1) |
| vip1_portb_luma | Video Input 1 Port B 420 Data Luma | 104 | YUV (0x1, 0x2,0x7, 0x17, 0x27, 0x37) | vip1_lo_y (2) |
| vip1_portb_chroma | Video Input 1 Port B 420 Data Chroma | 105 | YUV (0x5, 0x6, 0x7, 0x15, 0x16, 0x17, 0x27, 0x37) | vip1_lo_uv (1) |
| vip1_porta_rgb | Video Input 1 Port A RGB Data | 106 | RGB (0x0-0x19) | vip1_up_y (1) |
| vip1_portb_rgb | Video Input 1 Port B RGB Data | 107 | RGB (0x0-0x19) | vip1_lo_y (2) |
| vip2_mult_porta_src0 | Video Input 2 Port A Channel 0 | 108 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src1 | Video Input 2 Port A Channel 1 | 109 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src2 | Video Input 2 Port A Channel 2 | 110 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src3 | Video Input 2 Port A Channel 3 | 111 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src4 | Video Input 2 Port A Channel 4 | 112 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src5 | Video Input 2 Port A Channel 5 | 113 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src6 | Video Input 2 Port A Channel 6 | 114 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src7 | Video Input 2 Port A Channel 7 | 115 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src8 | Video Input 2 Port A Channel 8 | 116 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src9 | Video Input 2 Port A Channel 9 | 117 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src10 | Video Input 2 Port A Channel 10 | 118 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src11 | Video Input 2 Port A Channel 11 | 119 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-105. VPDMA Channels (continued)**

| Channel | Description | Channel Number | Data Type | Client |
|---|---|---|---|---|
| vip2_mult_porta_src12 | Video Input 2 Port A Channel 12 | 120 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src13 | Video Input 2 Port A Channel 13 | 121 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src14 | Video Input 2 Port A Channel 14 | 122 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_porta_src15 | Video Input 2 Port A Channel 15 | 123 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |
| vip2_mult_portb_src0 | Video Input 2 Port B Channel 0 | 124 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src1 | Video Input 2 Port B Channel 1 | 125 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src2 | Video Input 2 Port B Channel 2 | 126 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src3 | Video Input 2 Port B Channel 3 | 127 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src4 | Video Input 2 Port B Channel 4 | 128 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src5 | Video Input 2 Port B Channel 5 | 129 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src6 | Video Input 2 Port B Channel 6 | 130 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src7 | Video Input 2 Port B Channel 7 | 131 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src8 | Video Input 2 Port B Channel 8 | 132 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src9 | Video Input 2 Port B Channel 9 | 133 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src10 | Video Input 2 Port B Channel 10 | 134 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src11 | Video Input 2 Port B Channel 11 | 135 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src12 | Video Input 2 Port B Channel 12 | 136 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src13 | Video Input 2 Port B Channel 13 | 137 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src14 | Video Input 2 Port B Channel 14 | 138 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_portb_src15 | Video Input 2 Port B Channel 15 | 139 | YUV (0x7, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_mult_anca_src0 | Video Input 2 Port A Ancillary Data Channel 0 | 140 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src1 | Video Input 2 Port A Ancillary Data Channel 1 | 141 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src2 | Video Input 2 Port A Ancillary Data Channel 2 | 142 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src3 | Video Input 2 Port A Ancillary Data Channel 3 | 143 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src4 | Video Input 2 Port A Ancillary Data Channel 4 | 144 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src5 | Video Input 2 Port A Ancillary Data Channel 5 | 145 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src6 | Video Input 2 Port A Ancillary Data Channel 6 | 146 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src7 | Video Input 2 Port A Ancillary Data Channel 7 | 147 | OTHER (8) | vip2_anc_a (21) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-105. VPDMA Channels (continued)**

| Channel | Description | Channel Number | Data Type | Client |
|---|---|---|---|---|
| vip2_mult_anca_src8 | Video Input 2 Port A Ancillary Data Channel 8 | 148 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src9 | Video Input 2 Port A Ancillary Data Channel 9 | 149 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src10 | Video Input 2 Port A Ancillary Data Channel 10 | 150 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src11 | Video Input 2 Port A Ancillary Data Channel 11 | 151 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src12 | Video Input 2 Port A Ancillary Data Channel 12 | 152 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src13 | Video Input 2 Port A Ancillary Data Channel 13 | 153 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src14 | Video Input 2 Port A Ancillary Data Channel 14 | 154 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_anca_src15 | Video Input 2 Port A Ancillary Data Channel 15 | 155 | OTHER (8) | vip2_anc_a (21) |
| vip2_mult_ancb_src0 | Video Input 2 Port B Ancillary Data Channel 0 | 156 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src1 | Video Input 2 Port B Ancillary Data Channel 1 | 157 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src2 | Video Input 2 Port B Ancillary Data Channel 2 | 158 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src3 | Video Input 2 Port B Ancillary Data Channel 3 | 159 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src4 | Video Input 2 Port B Ancillary Data Channel 4 | 160 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src5 | Video Input 2 Port B Ancillary Data Channel 5 | 161 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src6 | Video Input 2 Port B Ancillary Data Channel 6 | 162 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src7 | Video Input 2 Port B Ancillary Data Channel 7 | 163 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src8 | Video Input 2 Port B Ancillary Data Channel 8 | 164 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src9 | Video Input 2 Port B Ancillary Data Channel 9 | 165 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src10 | Video Input 2 Port B Ancillary Data Channel 10 | 166 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src11 | Video Input 2 Port B Ancillary Data Channel 11 | 167 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src12 | Video Input 2 Port B Ancillary Data Channel 12 | 168 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src13 | Video Input 2 Port B Ancillary Data Channel 13 | 169 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src14 | Video Input 2 Port B Ancillary Data Channel 14 | 170 | OTHER (8) | vip2_anc_b (21) |
| vip2_mult_ancb_src15 | Video Input 2 Port B Ancillary Data Channel 15 | 171 | OTHER (8) | vip2_anc_b (21) |
| vip2_porta_luma | Video Input 2 Port A 420 Data Luma | 172 | YUV (0x1, 0x2, 0x7, 0x17, 0x27, 0x37) | vip2_up_y (20) |
| vip2_porta_chroma | Video Input 2 Port A 420 Data Chroma | 173 | YUV (0x5, 0x6, 0x7, 0x15, 0x16, 0x17, 0x27, 0x37) | vip2_up_uv (20) |
| vip2_portb_luma | Video Input 2 Port B 420 Data Luma | 174 | YUV (0x1, 0x2, 0x7, 0x17, 0x27, 0x37) | vip2_lo_y (21) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-105. VPDMA Channels (continued)**

| Channel | Description | Channel Number | Data Type | Client |
|---|---|---|---|---|
| vip2_portb_chroma | Video Input 2 Port B 420 Data Chroma | 175 | YUV (0x5, 0x6, 0x7, 0x15, 0x16, 0x17, 0x27, 0x37) | vip2_lo_uv (20) |
| vip2_porta_rgb | Video Input 2 Port A RGB Data | 176 | RGB (0x0-0x19) | vip2_up_y (20) |
| vip2_portb_rgb | Video Input 2 Port B RGB Data | 177 | RGB (0x0-0x19) | vip2_lo_y (21) |
| nf_read | Noise Filter Input Data 422 Interleaved | 178 | YUV (0x7, 0x17, 0x27, 0x37) | nf_422_in (24) |
| nf_write_luma | Noise Filter Data Write 420 Luma | 179 | YUV (0x1, 0x2, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | nf_420_y_out (23) |
| nf_write_chroma | Noise Filter Data Write 420 Chroma | 180 | YUV (0x6, 0x5) | nf_420_uv_out (23) |
| nf_last_luma | Noise Filter Previous Frame 420 Luma | 181 | YUV (0x2) | nf_420_y_in (22) |
| nf_last_chroma | Noise Filter Previous Frame 420 Chroma | 182 | YUV (0x6, 0x5) | nf_420_uv_in (22) |
| vbi_sd_venc | SD Video Encoder VBI Data | 184 | OTHER (8) | vbi_sdvenc (19) |
| post_comp_wr | Scaled Compositor or Transcode Writeback to Memory | 185 | YUV (0x7) | hdmi_wrbk_out (5) |
| pip_frame | 422I Video Input to Compositor (no scaling) | 186 | YUV (0x7) | pip_wrbk (10) |
| aux_in | 422I Video Input to Compositor (no scaling) | 187 | YUV (0x7) | comp_wrbk (11) |
| transcode1_luma | Independent Transcode Path 0 Luma | 188 | YUV (0x1, 0x2, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | trans1_luma (3) |
| transcode1_chroma | Independent Transcode Path 0 Chroma | 189 | YUV (0x5, 0x6, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | trans1_chroma (3) |
| transcode2_luma | Independent Transcode Path 0 Luma | 190 | YUV (0x1, 0x2, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | trans2_luma (18) |
| transcode2_chroma | Independent Transcode Path 0 Chroma | 191 | YUV (0x5, 0x6, 0x7 If data type 0x7 is used half the data fetched will be thrown out.) | trans2_chroma (18) |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.4.2 hd_dss_vpdma Client Buffering

Table 12-106 lists for each client the channels used, amount of buffering allocated for it, and the shared buffer used for its memory.

**Table 12-106. VPDMA Client Buffering**

| Client | Channel(s) | Buffering | Shared Buffer |
|---|---|---|---|
| dei_hq_1_chroma | hq_vid1_chroma | 11520 | DEI_MQ_VID |
| dei_hq_1_luma | hq_vid1_luma | 7680 | HD_DEI_VID |
| dei_hq_2_luma | hq_vid2_luma | 7680 | DEI_MQ_VID |
| dei_hq_2_chroma | hq_vid2_chroma | 11520 | HD_DEI_VID |
| dei_hq_3_luma | hq_vid3_luma | 7680 | DEI_MQ_VID1 |
| dei_hq_3_chroma | hq_vid3_chroma | 11520 | HD_DEI_VID1 |
| dei_hq_mv_in | hq_mv | 4096 | MV |
| dei_hq_mv_out | hq_mv_out | 4096 | MV |
| dei_sc_out | hq_scaler | 4096 | MEM_TO_MEM |
| pip_wrbk | pip_frame | 4096 | MEM_TO_MEM1 |
| sc_in_chroma | scaler_chroma | 11520 | DEI_MQ_VID1 |
| sc_in_luma | scaler_luma | 7680 | TRANS_VID0 |
| sc_out | scaler_out | 4096 | MEM_TO_MEM |
| comp_wrbk | aux_in | 4096 | MEM_TO_MEM1 |
| grpx1_data | grpx1 | 4096 | GRPX_BUF |
| grpx2_data | grpx2 | 4096 | GRPX_BUF1 |
| grpx3_data | grpx3 | 4096 | MV |
| vip1_lo_y | vip1_mult_porta_src0<br>vip1_mult_porta_src1<br>vip1_mult_porta_src2<br>vip1_mult_porta_src3<br>vip1_mult_porta_src4<br>vip1_mult_porta_src5<br>vip1_mult_porta_src6<br>vip1_mult_porta_src7<br>vip1_mult_porta_src8<br>vip1_mult_porta_src9<br>vip1_mult_porta_src10<br>vip1_mult_porta_src11<br>vip1_mult_porta_src12<br>vip1_mult_porta_src13<br>vip1_mult_porta_src14<br>vip1_mult_porta_src15<br>vip1_portb_luma<br>vip1_portb_rgb | 11520 | VP_WR |
| vip1_lo_uv | vip1_mult_portb_src0<br>vip1_mult_portb_src1<br>vip1_mult_portb_src2<br>vip1_mult_portb_src3<br>vip1_mult_portb_src4<br>vip1_mult_portb_src5<br>vip1_mult_portb_src6<br>vip1_mult_portb_src7<br>vip1_mult_portb_src8<br>vip1_mult_portb_src9<br>vip1_mult_portb_src10<br>vip1_mult_portb_src11<br>vip1_mult_portb_src12<br>vip1_mult_portb_src13<br>vip1_mult_portb_src14<br>vip1_mult_portb_src15<br>vip1_portb_chroma | 11520 | VP_WR |
| vip1_up_y | vip1_porta_luma<br>vip1_porta_rgb | 11520 | VP_WR |
| vip1_up_uv | vip1_porta_chroma | 11520 | VP_WR |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-106. VPDMA Client Buffering (continued)**

| Client | Channel(s) | Buffering | Shared Buffer |
|---|---|---|---|
| vip2_lo_y | vip2_mult_porta_src0<br>vip2_mult_porta_src1<br>vip2_mult_porta_src2<br>vip2_mult_porta_src3<br>vip2_mult_porta_src4<br>vip2_mult_porta_src5<br>vip2_mult_porta_src6<br>vip2_mult_porta_src7<br>vip2_mult_porta_src8<br>vip2_mult_porta_src9<br>vip2_mult_porta_src10<br>vip2_mult_porta_src11<br>vip2_mult_porta_src12<br>vip2_mult_porta_src13<br>vip2_mult_porta_src14<br>vip2_mult_porta_src15<br>vip2_portb_luma<br>vip2_portb_rgb | 11520 | VP_WR2 |
| vip2_lo_uv | vip2_mult_portb_src0<br>vip2_mult_portb_src1<br>vip2_mult_portb_src2<br>vip2_mult_portb_src3<br>vip2_mult_portb_src4<br>vip2_mult_portb_src5<br>vip2_mult_portb_src6<br>vip2_mult_portb_src7<br>vip2_mult_portb_src8<br>vip2_mult_portb_src9<br>vip2_mult_portb_src10<br>vip2_mult_portb_src11<br>vip2_mult_portb_src12<br>vip2_mult_portb_src13<br>vip2_mult_portb_src14<br>vip2_mult_portb_src15<br>vip2_portb_chroma | 11520 | VP_WR2 |
| vip2_up_y | vip2_porta_luma<br>vip2_porta_rgb | 11520 | VP_WR2 |
| vip2_up_uv | vip2_porta_chroma | 11520 | VP_WR2 |
| grpx1_st | grpx1_stencil | 1024 | GRPX_BUF |
| grpx2_st | grpx2_stencil | 1024 | GRPX_BUF1 |
| grpx3_st | grpx3_stencil | 1024 | MV |
| nf_422_in | nf_read | 4096 | NF_BUF |
| nf_420_y_in | nf_last_luma | 4096 | NF_BUF1 |
| nf_420_uv_in | nf_last_chroma | 4096 | NF_BUF |
| nf_420_y_out | nf_write_luma | 4096 | NF_BUF1 |
| nf_420_uv_out | nf_write_chroma | 4096 | NF_BUF |
| vbi_sdvenc | vbi_sd_venc | 256 | MEM_TO_MEM |
| vpi_ctl | | 1024 | MEM_TO_MEM1 |
| hdmi_wrbk_out | post_comp_wr | 4096 | MEM_TO_MEM |
| trans1_chroma | transcode1_chroma | 11520 | TRANS_VID0 |
| trans1_luma | transcode1_luma | 7680 | TRANS_VID1 |
| trans2_chroma | transcode2_chroma | 11520 | TRANS_VID2 |
| trans2_luma | transcode2_luma | 7680 | TRANS_VID3 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-106. VPDMA Client Buffering (continued)**

| Client | Channel(s) | Buffering | Shared Buffer |
|---|---|---|---|
| vip1_anc_a | vip1_mult_anca_src0<br>vip1_mult_anca_src1<br>vip1_mult_anca_src2<br>vip1_mult_anca_src3<br>vip1_mult_anca_src4<br>vip1_mult_anca_src5<br>vip1_mult_anca_src6<br>vip1_mult_anca_src7<br>vip1_mult_anca_src8<br>vip1_mult_anca_src9<br>vip1_mult_anca_src10<br>vip1_mult_anca_src11<br>vip1_mult_anca_src12<br>vip1_mult_anca_src13<br>vip1_mult_anca_src14<br>vip1_mult_anca_src15 | 0 | VP_WR |
| vip1_anc_b | vip1_mult_ancb_src0<br>vip1_mult_ancb_src1<br>vip1_mult_ancb_src2<br>vip1_mult_ancb_src3<br>vip1_mult_ancb_src4<br>vip1_mult_ancb_src5<br>vip1_mult_ancb_src6<br>vip1_mult_ancb_src7<br>vip1_mult_ancb_src8<br>vip1_mult_ancb_src9<br>vip1_mult_ancb_src10<br>vip1_mult_ancb_src11<br>vip1_mult_ancb_src12<br>vip1_mult_ancb_src13<br>vip1_mult_ancb_src14<br>vip1_mult_ancb_src15 | 0 | VP_WR |
| vip2_anc_a | vip2_mult_anca_src0<br>vip2_mult_anca_src1<br>vip2_mult_anca_src2<br>vip2_mult_anca_src3<br>vip2_mult_anca_src4<br>vip2_mult_anca_src5<br>vip2_mult_anca_src6<br>vip2_mult_anca_src7<br>vip2_mult_anca_src8<br>vip2_mult_anca_src9<br>vip2_mult_anca_src10<br>vip2_mult_anca_src11<br>vip2_mult_anca_src12<br>vip2_mult_anca_src13<br>vip2_mult_anca_src14<br>vip2_mult_anca_src15 | 0 | VP_WR2 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.4.3 hd_dss_vpdma Client Functionality

Table 12-107 lists for each client the sizes it handles for tiled and non-tiled memory spaces as well as any additional features it supports such as Virtual Video Buffering, 1D descriptor field.

**Table 12-107. HDVPSS Client Functionality**

| Client | Tiled Memory Max Line Size | Non-Tiled Memory Max Line Size | Additional Features |
|---|---|---|---|
| dei_hq_1_chroma | 1920 | 1920 | Virtual Video Buffer with line buffer limitations, TILED |
| dei_hq_1_luma | 1920 | 4096 | Virtual Video Buffer, TILED |
| dei_hq_2_luma | 1920 | 4096 | Virtual Video Buffer, TILED |
| dei_hq_2_chroma | 1920 | 1920 | Virtual Video Buffer with line buffer limitations, TILED |
| dei_hq_3_luma | 1920 | 4096 | Virtual Video Buffer, TILED |
| dei_hq_3_chroma | 1920 | 1920 | Virtual Video Buffer with line buffer limitations, TILED |
| dei_hq_mv_in | Tiled Data Not Supported | 4096 | |
| dei_hq_mv_out | Tiled Data Not Supported | 4096 | |
| dei_sc_out | Tiled Data Not Supported | 4096 | |
| pip_wrbk | Tiled Data Not Supported | 4096 | Virtual Video Buffer |
| sc_in_chroma | 1920 | 1920 | Virtual Video Buffer with line buffer limitations, TILED |
| sc_in_luma | 1920 | 4096 | Virtual Video Buffer, TILED |
| sc_out | Tiled Data Not Supported | 4096 | |
| comp_wrbk | Tiled Data Not Supported | 4096 | Virtual Video Buffer |
| grpx1_data | Tiled Data Not Supported | 4096 | |
| grpx2_data | Tiled Data Not Supported | 4096 | |
| grpx3_data | Tiled Data Not Supported | 4096 | |
| vip1_lo_y | 1920 | 4096 | TILED |
| vip1_lo_uv | 1920 | 4096 | TILED |
| vip1_up_y | 1920 | 4096 | TILED |
| vip1_up_uv | 1920 | 4096 | TILED |
| vip2_lo_y | 1920 | 4096 | TILED |
| vip2_lo_uv | 1920 | 4096 | TILED |
| vip2_up_y | 1920 | 4096 | TILED |
| vip2_up_uv | 1920 | 4096 | TILED |
| grpx1_st | Tiled Data Not Supported | 4096 | |
| grpx2_st | Tiled Data Not Supported | 4096 | |
| grpx3_st | Tiled Data Not Supported | 4096 | |
| nf_422_in | Tiled Data Not Supported | 4096 | |
| nf_420_y_in | 1920 | 4096 | TILED |
| nf_420_uv_in | 1920 | 4096 | TILED |
| nf_420_y_out | 1920 | 4096 | TILED |
| nf_420_uv_out | 1920 | 4096 | TILED |
| vbi_sdvenc | Tiled Data Not Supported | 4096 | |
| vpi_ctl | Tiled Data Not Supported | 4096 | |
| hdmi_wrbk_out | Tiled Data Not Supported | 4096 | |
| trans1_chroma | 1920 | 1920 | Virtual Video Buffer with line buffer limitations, TILED |
| trans1_luma | 1920 | 4096 | Virtual Video Buffer, TILED |
| trans2_chroma | 1920 | 1920 | Virtual Video Buffer with line buffer limitations, TILED |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-107. HDVPSS Client Functionality (continued)**

| Client | Tiled Memory Max Line Size | Non-Tiled Memory Max Line Size | Additional Features |
|---|---|---|---|
| trans2_luma | 1920 | 4096 | Virtual Video Buffer, TILED |
| vip1_anc_a | Tiled Data Not Supported | 4096 | |
| vip1_anc_b | Tiled Data Not Supported | 4096 | |
| vip2_anc_a | Tiled Data Not Supported | 4096 | |

#### *12.2.13.4.4 VPDMA Interrupts to HDVPSS*

The VPDMA has 100 interrupts that it provides to the HDVPSS. These 100 interrupts are actually four groups of 25 interrupts, where the interrupts within each group are identical in terms of their contents, but can be masked independently within the HDVPSS to allow different processors to see different interrupts. Each group of 25 interrupts are mapped to each of the four interrupt outputs from the HDVPSS. The interrupts can fire for the completion of a channel, the completion of a client, completion of a list, a list channel notification completion, or a special event received by the List Manager. These 25 interrupts are additionally grouped by the VPDMA from a total of 251 interrupt sources.

Each of these 25 interrupts can have multiple sources within the VPDMA, and within the VPDMA there are separate status and masking controls to provide the additional layer of register reads to determine the exact source of the interrupt.

Table 12-108 shows all interrupt sources received by the HDVPSS from the VPDMA.

**Table 12-108. HDVPSS Interrupt from VPDMA**

| Interrupt | Description |
|---|---|
| vpdma_int_channel_group0 | An unmasked channel interrupt for interrupt group 0 in channel register 0 has fired. |
| vpdma_int_channel_group1 | An unmasked channel interrupt for interrupt group 1 in channel register 0 has fired. |
| vpdma_int_channel_group2 | An unmasked channel interrupt for interrupt group 2 in channel register 0 has fired. |
| vpdma_int_channel_group3 | An unmasked channel interrupt for interrupt group 3 in channel register 0 has fired. |
| vpdma_int_channel_group4 | An unmasked channel interrupt for interrupt group 4 in channel register 0 has fired. |
| vpdma_int_channel_group5 | An unmasked channel interrupt for interrupt group 5 in channel register 0 has fired. |
| vpdma_int_channel_group6 | An unmasked channel interrupt for interrupt group 6 in channel register 0 has fired. |
| vpdma_int_list0_complete | List 0 has completed |
| vpdma_int_list0_notify | The data transfer in list 0 with the Notify Field set in the descriptor has completed |
| vpdma_int_list1_complete | List 1 has completed |
| vpdma_int_list1_notify | The data transfer in list 1 with the Notify Field set in the descriptor has completed |
| vpdma_int_list2_complete | List 2 has completed |
| vpdma_int_list2_notify | The data transfer in list 2 with the Notify Field set in the descriptor has completed |
| vpdma_int_list3_complete | List 3 has completed |
| vpdma_int_list3_notify | The data transfer in list 3 with the Notify Field set in the descriptor has completed |
| vpdma_int_list4_complete | List 4 has completed |
| vpdma_int_list4_notify | The data transfer in list 4 with the Notify Field set in the descriptor has completed |
| vpdma_int_list5_complete | List 5 has completed |
| vpdma_int_list5_notify | The data transfer in list 5 with the Notify Field set in the descriptor has completed |
| vpdma_int_list6_complete | List 6 has completed |
| vpdma_int_list6_notify | The data transfer in list 6 with the Notify Field set in the descriptor has completed |
| vpdma_int_list7_complete | List 7 has completed |
| vpdma_int_list7_notify | The data transfer in list 7 with the Notify Field set in the descriptor has completed |
| vpdma_int_client | Client Interrupt |
| vpdma_int_descriptor | Descriptor Interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

As stated, the above 25 interrupts are replicated four times. Each group has a separate mask register and status register (described in Section 12.3.8.13 to Section 12.3.8.92). At HDVPSS level, these four groups are connected to the four external interrupts (described in Section 12.3.7.3 to Section 12.3.7.31). As such, the overall interrupt mechanism for software is two-tierd.

In the previous table, the “channel_group”, “client” and “descriptor” interrupts are actually a set of additional interrupts. When software receives an interrupt from a “channel_group,” “client,” or “descriptor” it must read the appropriate register within the VPDMA (described in Section 12.3.8.13 to Section 12.3.8.92 and use Table 12-109 to determine the actual interrupt.

**Table 12-109. HDVPSS Interrupt Sources**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_aux_in | channel_group5 | The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. |
| channel_grpx1 | channel_group0 | The last read DMA transaction has occurred for channel grpx1 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_data will now accept a new descriptor from the List Manager. |
| channel_grpx1_clut | channel_group1 | The last write DMA transaction has completed for channel grpx1_clut. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client grpx1_clut_clt then the client will be fully empty at this point. |
| channel_grpx1_stencil | channel_group1 | The last read DMA transaction has occurred for channel grpx1_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_st will now accept a new descriptor from the List Manager. |
| channel_grpx2 | channel_group0 | The last read DMA transaction has occurred for channel grpx2 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_data will now accept a new descriptor from the List Manager. |
| channel_grpx2_clut | channel_group1 | The last write DMA transaction has completed for channel grpx2_clut. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client grpx2_clut_clt then the client will be fully empty at this point. |
| channel_grpx2_stencil | channel_group1 | The last read DMA transaction has occurred for channel grpx2_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_st will now accept a new descriptor from the List Manager. |
| channel_grpx3 | channel_group0 | The last read DMA transaction has occurred for channel grpx3 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_data will now accept a new descriptor from the List Manager. |
| channel_grpx3_clut | channel_group1 | The last write DMA transaction has completed for channel grpx3_clut. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client grpx3_clut_clt then the client will be fully empty at this point. |
| channel_grpx3_stencil | channel_group1 | The last read DMA transaction has occurred for channel grpx3_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_st will now accept a new descriptor from the List Manager. |
| channel_hq_mv | channel_group0 | The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. |
| channel_hq_mv_out | channel_group0 | The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_hq_scaler | channel_group0 | The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. |
| channel_hq_vid1_chroma | channel_group0 | The last write DMA transaction has completed for channel hq_vid1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_hq_vid1_luma | channel_group0 | The last write DMA transaction has completed for channel hq_vid1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_hq_vid2_chroma | channel_group0 | The last write DMA transaction has completed for channel hq_vid2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_hq_vid2_luma | channel_group0 | The last write DMA transaction has completed for channel hq_vid2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_hq_vid3_chroma | channel_group0 | The last write DMA transaction has completed for channel hq_vid3_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_hq_vid3_luma | channel_group0 | The last write DMA transaction has completed for channel hq_vid3_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_nf_last_chroma | channel_group5 | The last write DMA transaction has completed for channel nf_last_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_nf_last_luma | channel_group5 | The last write DMA transaction has completed for channel nf_last_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_nf_read | channel_group5 | The last read DMA transaction has occurred for channel nf_read and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_422_in will now accept a new descriptor from the List Manager. |
| channel_nf_write_chroma | channel_group5 | The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_nf_write_luma | channel_group5 | The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_pip_frame | channel_group5 | The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. |
| channel_post_comp_wr | channel_group5 | The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. |
| channel_scaler_chroma | channel_group0 | The last write DMA transaction has completed for channel scaler_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_scaler_luma | channel_group0 | The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_scaler_out | channel_group0 | The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. |
| channel_transcode1_chroma | channel_group5 | The last write DMA transaction has completed for channel transcode1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_transcode1_luma | channel_group5 | The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_transcode2_chroma | channel_group5 | The last write DMA transaction has completed for channel transcode2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_transcode2_luma | channel_group5 | The last write DMA transaction has completed for channel transcode2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_vbi_sd_venc | channel_group5 | The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. |
| channel_vip1_mult_anca_src0 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src1 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src10 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src11 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src12 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src13 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src14 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src15 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_vip1_mult_anca_src2 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src3 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src4 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src5 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src6 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src7 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src8 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_anca_src9 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src0 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src1 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src10 | channel_group3 | The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src11 | channel_group3 | The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src12 | channel_group3 | The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src13 | channel_group3 | The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src14 | channel_group3 | The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_vip1_mult_ancb_src15 | channel_group3 | The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src2 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src3 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src4 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src5 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src6 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src7 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src8 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_ancb_src9 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src0 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src1 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src10 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src11 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src12 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src13 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_vip1_mult_porta_src14 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src15 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src2 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src3 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src4 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src5 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src6 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src7 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src8 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_porta_src9 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src0 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src1 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src10 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src11 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src12 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_vip1_mult_portb_src13 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src14 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src15 | channel_group2 | The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src2 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src3 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src4 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src5 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src6 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src7 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src8 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_mult_portb_src9 | channel_group1 | The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. |
| channel_vip1_porta_chroma | channel_group3 | The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_vip1_porta_luma | channel_group3 | The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_vip1_porta_rgb | channel_group3 | The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. |
| channel_vip1_portb_chroma | channel_group3 | The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_vip1_portb_luma | channel_group3 | The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_vip1_portb_rgb | channel_group3 | The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src0 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src1 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src10 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src11 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src12 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src13 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src14 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src15 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src2 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src3 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src4 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src5 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src6 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_vip2_mult_anca_src7 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src8 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_anca_src9 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src0 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src1 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src10 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src11 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src12 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src13 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src14 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src15 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src2 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src3 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src4 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src5 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_vip2_mult_ancb_src6 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src7 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src8 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_ancb_src9 | channel_group5 | The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src0 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src1 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src10 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src11 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src12 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src13 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src14 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src15 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src2 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src3 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src4 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_vip2_mult_porta_src5 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src6 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src7 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src8 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_porta_src9 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src0 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src1 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src10 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src11 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src12 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src13 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src14 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src15 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src2 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src3 | channel_group3 | The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| channel_vip2_mult_portb_src4 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src5 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src6 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src7 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src8 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_mult_portb_src9 | channel_group4 | The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. |
| channel_vip2_porta_chroma | channel_group5 | The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_vip2_porta_luma | channel_group5 | The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_vip2_porta_rgb | channel_group5 | The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. |
| channel_vip2_portb_chroma | channel_group5 | The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_vip2_portb_luma | channel_group5 | The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. |
| channel_vip2_portb_rgb | channel_group5 | The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. |
| client_comp_wrbk | client | The client interface comp_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_dei_hq_1_chroma | client | The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_dei_hq_1_luma | client | The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| client_dei_hq_2_chroma | client | The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_dei_hq_2_luma | client | The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_dei_hq_3_chroma | client | The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_dei_hq_3_luma | client | The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_dei_hq_mv_in | client | The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_dei_hq_mv_out | client | The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_dei_sc_out | client | The client interface dei_sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_grpx1_data | client | The client interface grpx1_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_grpx1_st | client | The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_grpx2_data | client | The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_grpx2_st | client | The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_grpx3_data | client | The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_grpx3_st | client | The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_hdmi_wrbk_out | client | The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_nf_420_uv_in | client | The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| client_nf_420_uv_out | client | The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_nf_420_y_in | client | The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_nf_420_y_out | client | The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_nf_422_in | client | The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_pip_wrbk | client | The client interface pip_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_sc_in_chroma | client | The client interface sc_in_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_sc_in_luma | client | The client interface sc_in_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_sc_out | client | The client interface sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_trans1_chroma | client | The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_trans1_luma | client | The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_trans2_chroma | client | The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_trans2_luma | client | The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_vbi_sdvenc | client | The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| client_vip1_anc_a | client | The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vip1_anc_b | client | The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| client_vip1_lo_uv | client | The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vip1_lo_y | client | The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vip1_up_uv | client | The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vip1_up_y | client | The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vip2_anc_a | client | The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vip2_anc_b | client | The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vip2_lo_uv | client | The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vip2_lo_y | client | The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vip2_up_uv | client | The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vip2_up_y | client | The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. |
| client_vpi_ctl | client | The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. |
| control_descriptor_int0 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. |
| control_descriptor_int1 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. |
| control_descriptor_int10 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. |
| control_descriptor_int11 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. |
| control_descriptor_int12 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. |
| control_descriptor_int13 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. |
| control_descriptor_int14 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. |
| control_descriptor_int15 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-109. HDVPSS Interrupt Sources (continued)**

| Interrupt | Interrupt Group | Description |
|---|---|---|
| control_descriptor_int2 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. |
| control_descriptor_int3 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. |
| control_descriptor_int4 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. |
| control_descriptor_int5 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. |
| control_descriptor_int6 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. |
| control_descriptor_int7 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. |
| control_descriptor_int8 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. |
| control_descriptor_int9 | descriptor | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. |

### 12.2.13.5 VPDMA Configuration

The following section describes the different ways of configuring VPDMA for data transfers.

#### *12.2.13.5.1 Regular List*

A regular list executes each descriptor in order until the end of the list is reached. When the end of the list is reached an interrupt is sent and the list can be reused by software. A regular list can contain any descriptor types. Software creates the list at some location in external memory. After completing writing the list software then writes the location of the list to the LIST_ADDRESS register and then writes the LIST_ATTRIBUTE register. If the NUMBER in the LIST_ATTRIBUTE is not an active list then the List will be loaded and begin to execute the next time the List Manager gets to IDLE after processing previous loaded lists. If the NUMBER in the LIST_ATTRIBUTE is busy then the LIST_ADDRESS and LIST_ATTRIBUTE registers will be locked until the active list specified by NUMBER completes. Any writes to the LIST_ADDRESS or LIST_ATTRIBUTE will result in mmr_wready being held inactive until the list is freed.

The different ports inside VPDMA requires different list setup.

##### *12.2.13.5.1.1 Video Input Ports*

The Video Input Ports (VIP) can be used in multiple ways. Depending on how the input ports are configured will determine how the lists to manage them need to be setup. The ways in which the ports can work are multiplexed data stream, single YUV color separate stream, dual YUV interleaved or single RGB stream. Each port also has an ancillary data port that can run either multiplexed or single data stream and shares the buffering with the main data stream. For all cases the descriptor must be loaded into the client before the vertical sync is received on the VIP port or the entire frame will be dropped. As with all write clients the VIP channels can be shadowed so the next frame/field descriptor can be loaded while the previous frame is running without stalling the list.

###### *12.2.13.5.1.1.1 Multiplexed Data Streams*

In the case of a multiplexed data stream input the channels that should be used are VIPX_PORTY_MULT_SRCZ. Where X is the specific VIP port that wants to be used and port Y is the port A or port B that is receiving the data. Finally Z is the channel number. For Split line mux mode the LSB of the channel will determine if the line is a split line or a complete line. This is required so that the data streams do not get mixed when a channel ends without completing a line. In this mode the data will always be sent out as 422 Interleaved data to the destination specified in the descriptor.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.5.1.1.2 Single YUV Color Separate

If the port is configured to be used as sending out color separated YUV data then the channels that must be used are VIPX_PORTY_LUMA and VIPX_PORTY_CHROMA for the luma and chroma components respectively. The data type can be either 420 or 422 color separate in this case.

#### 12.2.13.5.1.1.3 Dual YUV Interleaved

If the port is configured to be used to send out 422 interleaved data from a non-multiplexed source then the channels that must be used are VIPX_PORTY_LUMA or VIPX_PORTY_CHROMA depending on if the data stream is connected to the luma or chroma port. The data type must be 422 interleaved in this case.

#### 12.2.13.5.1.1.4 Single RGB Stream

If the port is configured to be used to send out RGB stream then the channel VIPX_PORTY_RGB must be used. The incoming data is then assumed to be RGB 888 data and this is combined with the Background Color Alpha register field to make ARGB8888 data that will then be sent out based on the data type. If the data type uses less then the full 8 bits the lower bits are dropped and just the upper bits are sent to get the correct data format.

### 12.2.13.5.1.2 De-Interlacers (DEIs)

The DEIs have two areas that require special handling in setting up the descriptors. The first is with the line buffers. The line buffer must be set to match the mode that the chroma upsampler and the DEI expect to receive for chroma and luma respectively. Refer to the Chroma Upsampler specification for the settings for the chroma client. The luma client should use the same settings for the chroma assuming that the data type is 422. The line buffer mode is set by the CSTAT register for the client. This value is not shadowed and must be set before the descriptor is loaded and can not be changed until the client has finished. This can be accomplished by using a Sync on Client Not Active control descriptor prior to a configuration descriptor to change the line mode. The video clients also must have the 1D descriptor field set to 1 if compression and decompression is used for the field data. Also, if the compression and decompression are set then the write channel descriptors must also generate the next inbound descriptor as this is the only way to know the size of the outbound data that will be the inbound data for the next frame.

### 12.2.13.5.1.3 Transcode Paths

The transcode paths must set the chroma line buffer to match the chroma upsampler in the transcode path. Refer to the Chroma Upsampler specification for the settings for the chroma client. The line buffer mode is set by the CSTAT register for the client. This value is not shadowed and must be set before the descriptor is loaded and can not be changed until the client has finished. This can be accomplished by using a Sync on Client Not Active control descriptor prior to a configuration descriptor to change the line mode.

### 12.2.13.5.1.4 Noise Filter (NF)

The noise filter clients break the frame into 32 by 32 tiles for 422 data and 420 Luma data. The noise filter clients break 420 chroma data into 32x16 tiles. The incoming 422 frame should be evenly divisible by 32 by 32. The outgoing frame buffers should leave enough space for a frame that is evenly divisible by 32 by 32 for luma and 32 by 16 for chroma.

### 12.2.13.5.1.5 Graphics

The graphics ports in this device are tightly coupled with the VPDMA clients that service them. Each graphics port expects a list of data descriptors to describe each region that make up the frame.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.5.1.5.1 Region Descriptor

The Graphics module uses the standard Inbound data transfer descriptor as all other channel but it has additional information in the Client Specific Attributes field that are passed down to the Graphics processing module.

**Table 12-110. Graphics Region Descriptor Format**

| Attribute | Location | Description |
|---|---|---|
| Region_Height | Word 4 Bits 15:0 | Height of the region |
| Region Width | Word 4 Bits 31:16 | The number of pixels wide for the region |
| Vertical Start | Word 5 Bits 15:0 | position of region (Y origin) 0 is top of frame |
| Horizontal Start | Word 5 Bits 31:16 | position of region (X origin) 0 is left side of frame |
| Disp_pri | Word 6 Bits 3:0 | Region display priority 0x0 is on the bottom 0xF is on the top |
| First Region | Word 6 Bit 7 | Marks the region as the first region in a new frame. This bit must be set to 1 on the first region only |
| Last Region | Word 6 Bit 8 | Marks the last region in a frame. This bit must be set only on the last region. If only 1 region exists then both this and the First Region bit should be set |
| Scaler | Word 6 Bit 9 | The region should be scaled using the current frames resizer configuration |
| AF | Word 6 Bit 10 | The region should be passed through the scaler to Anti-flicker the data |
| Stencil | Word 6 Bit 11 | The region should use the stencil client applied to this data. A separate data descriptor for the stencil data must be created and should have been the data descriptor in the list prior to the current region descriptor. |
| Reserved | [76:78] | Reserved |
| bbox | Word 6 Bit 15 | A single pixel bounding box should be applied along the region with the alpha value of the data replaced with the alpha data in the bb_alpha field. |
| bb_alpha | Word 6 Bit 23:16 | The alpha value to use for the bounding box if it is enabled. |
| blend_alpha | Word 6 Bit 31:24 | The alpha value to use if the blend_type field is set for Region global blending. |
| blend_type | Word 7 Bits 1:0 | The type of blending to perform on the region. This is used to replace the alpha value presented to the Graphics module. 0 — no blending 1 — Region global blending 2 — reserved. (color (palette) Blending) 3 — Pixel (embedded alpha) Blending |
| trans_enable | Word 7 Bit 5 | Enable transparency for the region. If this is set the trans_cfg and trans_color fields are used to replace the alpha of a specific color to make it transparent |
| tr_lsb_mask | Word 7 Bits 7-6 | Transparency LSB bit masking 0<br>0 : No masking 01 : Mask [0] during pixel data comparison to TRANS_COLOR 1<br>0 : Mask[1:0] 1<br>1 : Mask[2:0] ex)., if TR_LSB_MASK is set to 2'b11, top 5 bits of R/G/B component data are compared to the corresponding 5 bits of TRANS_COLOR R/G/B color. |
| trans color | Word 7 Bits 31:8 | Transparency Color (RGB24) |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.5.1.5.2 Frame Configuration Descriptors

The graphics module also requires its configuration through the use of Configuration descriptors. The Graphics module can accept a frame configuration packet with or without the resizer settings. The frame must be configured before starting the graphics port for its first frame but a new configuration descriptor is not required unless either the frame attributes or the resizer attributes are changing from the previous frame.

**Table 12-111. Frame Configuration Descriptor Format**

| Attribute | Location | Value |
|---|---|---|
| Packet Type | Header Word 3 Bits 31:27 | 0xb |
| Direct | Header Word 3 Bit 26 | Direct Command = 1<br>Indirect Command = 0 |
| Class | Header Word 3 Bits 25:24 | 1 = Block |
| Destination | Header Word 3 Bits 23:16 | Graphics port that the frame is being configured. |
| Payload Length | Header Word 3 Bits 15:0 | 1 |
| Length | Header Word 2 Bits 31:24 | 1 |
| Address | Header Word 2 Bits 23:0 | 1 |
| Payload Location | Header Word 1 Bits 31:0 | Address of the Payload if Indirect Command |
| frame_height | Payload Word 1 Bits 15:0 | Frame height |
| frame_width | Payload Word 1 Bits 31:16 | Frame width |
| src_fmt_interlaced | Payload Word 3 Bit 30 | Format of the source image: 0 = Progressive 1 = Interlaced |
| Soft Reset | Payload Word 3 Bit 31 | Software reset of GRPX data pipeline |

**Table 12-112. Frame with Resizer Configuration Descriptor Format**

| Attribute | Location | Value |
|---|---|---|
| Packet Type | Header Word 3 Bits 31:27 | 0xb |
| Direct | Header Word 3 Bit 26 | Direct Command = 1<br>Indirect Command = 0 |
| Class | Header Word 3 Bits 25:24 | 1 = Block |
| Destination | Header Word 3 Bits 23:16 | Graphics port that the frame is being configured. |
| Payload Length | Header Word 3 Bits 15:0 | 12 |
| Length | Header Word 2 Bits 31:24 | 12 |
| Address | Header Word 2 Bits 23:0 | 1 |
| Payload Location | Header Word 1 Bits 31:0 | Address of the Payload if Indirect Command |
| frame_height | Payload Word 1 Bits 15:0 | Frame height |
| frame_width | Payload Word 1 Bits 31:16 | Frame width |
| src_fmt_interlaced | Payload Word 3 Bit 30 | Format of the source image: 0 = Progressive 1 = Interlaced |
| Soft Reset | Payload Word 3 Bit 31 | Software reset of GRPX data pipeline |
| coefh0_p0 | Payload Word 4 Bits 9:0 | Coefficient for Horizontal TAP 0 Phase 0 |
| coefh0_p1 | Payload Word 4 Bits 25:16 | Coefficient for Horizontal TAP 0 Phase 1 |
| coefh0_p2 | Payload Word 5 Bits 9:0 | Coefficient for Horizontal TAP 0 Phase 2 |
| coefh0_p3 | Payload Word 5 Bits 25:16 | Coefficient for Horizontal TAP 0 Phase 3 |
| coefh0_p4 | Payload Word 6 Bits 9:0 | Coefficient for Horizontal TAP 0 Phase 4 |
| coefh0_p5 | Payload Word 6 Bits 25:16 | Coefficient for Horizontal TAP 0 Phase 5 |
| coefh0_p6 | Payload Word 7 Bits 9:0 | Coefficient for Horizontal TAP 0 Phase 6 |
| coefh0_p7 | Payload Word 7 Bits 25:16 | Coefficient for Horizontal TAP 0 Phase 7 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-112. Frame with Resizer Configuration Descriptor Format (continued)**

| Attribute | Location | Value |
|---|---|---|
| coefh1_p0 | Payload Word 8 Bits 9:0 | Coefficient for Horizontal TAP 1 Phase 0 |
| coefh1_p1 | Payload Word 8 Bits 25:16 | Coefficient for Horizontal TAP 1 Phase 1 |
| coefh1_p2 | Payload Word 9 Bits 9:0 | Coefficient for Horizontal TAP 1 Phase 2 |
| coefh1_p3 | Payload Word 9 Bits 25:16 | Coefficient for Horizontal TAP 1 Phase 3 |
| coefh1_p4 | Payload Word 10 Bits 9:0 | Coefficient for Horizontal TAP 1 Phase 4 |
| coefh1_p5 | Payload Word 10 Bits 25:16 | Coefficient for Horizontal TAP 1 Phase 5 |
| coefh1_p6 | Payload Word 11 Bits 9:0 | Coefficient for Horizontal TAP 1 Phase 6 |
| coefh1_p7 | Payload Word 11 Bits 25:16 | Coefficient for Horizontal TAP 1 Phase 7 |
| coefh2_p0 | Payload Word 12 Bits 9:0 | Coefficient for Horizontal TAP 2 Phase 0 |
| coefh2_p1 | Payload Word 12 Bits 25:16 | Coefficient for Horizontal TAP 2 Phase 1 |
| coefh2_p2 | Payload Word 13 Bits 9:0 | Coefficient for Horizontal TAP 2 Phase 2 |
| coefh2_p3 | Payload Word 13 Bits 25:16 | Coefficient for Horizontal TAP 2 Phase 3 |
| coefh2_p4 | Payload Word 14 Bits 9:0 | Coefficient for Horizontal TAP 2 Phase 4 |
| coefh2_p5 | Payload Word 14 Bits 25:16 | Coefficient for Horizontal TAP 2 Phase 5 |
| coefh2_p6 | Payload Word 15 Bits 9:0 | Coefficient for Horizontal TAP 2 Phase 6 |
| coefh2_p7 | Payload Word 15 Bits 25:16 | Coefficient for Horizontal TAP 2 Phase 7 |
| coefh3_p0 | Payload Word 16 Bits 9:0 | Coefficient for Horizontal TAP 3 Phase 0 |
| coefh3_p1 | Payload Word 16 Bits 25:16 | Coefficient for Horizontal TAP 3 Phase 1 |
| coefh3_p2 | Payload Word 17 Bits 9:0 | Coefficient for Horizontal TAP 3 Phase 2 |
| coefh3_p3 | Payload Word 17 Bits 25:16 | Coefficient for Horizontal TAP 3 Phase 3 |
| coefh3_p4 | Payload Word 18 Bits 9:0 | Coefficient for Horizontal TAP 3 Phase 4 |
| coefh3_p5 | Payload Word 18 Bits 25:16 | Coefficient for Horizontal TAP 3 Phase 5 |
| coefh3_p6 | Payload Word 19 Bits 9:0 | Coefficient for Horizontal TAP 3 Phase 6 |
| coefh3_p7 | Payload Word 19 Bits 25:16 | Coefficient for Horizontal TAP 3 Phase 7 |
| coefh4_p0 | Payload Word 20 Bits 9:0 | Coefficient for Horizontal TAP 4 Phase 0 |
| coefh4_p1 | Payload Word 20 Bits 25:16 | Coefficient for Horizontal TAP 4 Phase 1 |
| coefh4_p2 | Payload Word 21 Bits 9:0 | Coefficient for Horizontal TAP 4 Phase 2 |
| coefh4_p3 | Payload Word 21 Bits 25:16 | Coefficient for Horizontal TAP 4 Phase 3 |
| coefh4_p4 | Payload Word 22 Bits 9:0 | Coefficient for Horizontal TAP 4 Phase 4 |
| coefh4_p5 | Payload Word 22 Bits 25:16 | Coefficient for Horizontal TAP 4 Phase 5 |
| coefh4_p6 | Payload Word 23 Bits 9:0 | Coefficient for Horizontal TAP 4 Phase 6 |
| coefh4_p7 | Payload Word 23 Bits 25:16 | Coefficient for Horizontal TAP 4 Phase 7 |
| countvalhorz | Payload Word 24 Bits 14:0 | Numerator value for fractional counter countval / 2048 |
| fineoffsethorz | Payload Word 24 Bits 30:16 | Count offset for start of line |
| coefv0_p0 | Payload Word 25 Bits 9:0 | Coefficient for Vertical TAP 0 Phase 0 |
| coefv0_p1 | Payload Word 25 Bits 25:16 | Coefficient for Vertical TAP 0 Phase 1 |
| coefv0_p2 | Payload Word 26 Bits 9:0 | Coefficient for Vertical TAP 0 Phase 2 |
| coefv0_p3 | Payload Word 26 Bits 25:16 | Coefficient for Vertical TAP 0 Phase 3 |
| coefv0_p4 | Payload Word 27 Bits 9:0 | Coefficient for Vertical TAP 0 Phase 4 |
| coefv0_p5 | Payload Word 27 Bits 25:16 | Coefficient for Vertical TAP 0 Phase 5 |
| coefv0_p6 | Payload Word 28 Bits 9:0 | Coefficient for Vertical TAP 0 Phase 6 |
| coefv0_p7 | Payload Word 28 Bits 25:16 | Coefficient for Vertical TAP 0 Phase 7 |
| coefv1_p0 | Payload Word 29 Bits 9:0 | Coefficient for Vertical TAP 1 Phase 0 |
| coefv1_p1 | Payload Word 29 Bits 25:16 | Coefficient for Vertical TAP 1 Phase 1 |
| coefv1_p2 | Payload Word 30 Bits 9:0 | Coefficient for Vertical TAP 1 Phase 2 |
| coefv1_p3 | Payload Word 30 Bits 25:16 | Coefficient for Vertical TAP 1 Phase 3 |
| coefv1_p4 | Payload Word 31 Bits 9:0 | Coefficient for Vertical TAP 1 Phase 4 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-112. Frame with Resizer Configuration Descriptor Format (continued)**

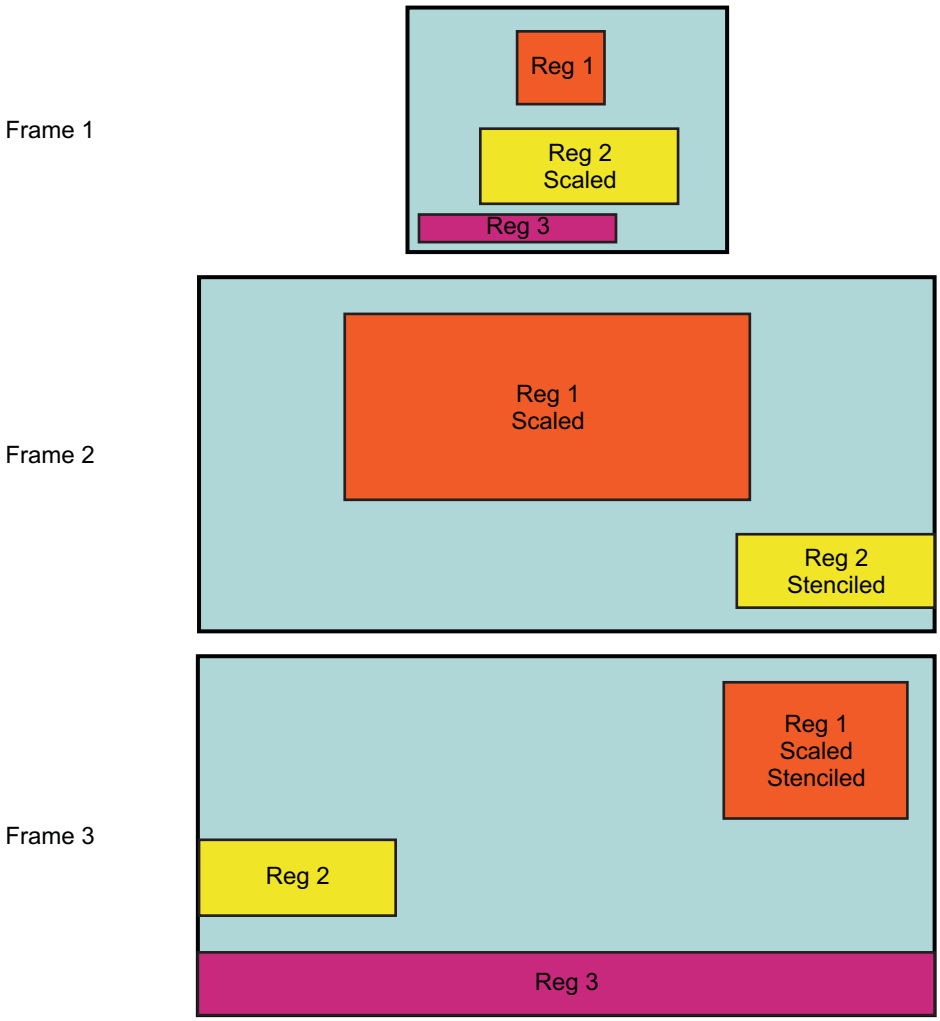
| Attribute | Location | Value |
|---|---|---|
| coefv1_p5 | Payload Word 31 Bits 25:16 | Coefficient for Vertical TAP 1 Phase 5 |
| coefv1_p6 | Payload Word 32 Bits 9:0 | Coefficient for Vertical TAP 1 Phase 6 |
| coefv1_p7 | Payload Word 32 Bits 25:16 | Coefficient for Vertical TAP 1 Phase 7 |
| coefv2_p0 | Payload Word 33 Bits 9:0 | Coefficient for Vertical TAP 2 Phase 0 |
| coefv2_p1 | Payload Word 33 Bits 25:16 | Coefficient for Vertical TAP 2 Phase 1 |
| coefv2_p2 | Payload Word 34 Bits 9:0 | Coefficient for Vertical TAP 2 Phase 2 |
| coefv2_p3 | Payload Word 34 Bits 25:16 | Coefficient for Vertical TAP 2 Phase 3 |
| coefv2_p4 | Payload Word 35 Bits 9:0 | Coefficient for Vertical TAP 2 Phase 4 |
| coefv2_p5 | Payload Word 35 Bits 25:16 | Coefficient for Vertical TAP 2 Phase 5 |
| coefv2_p6 | Payload Word 36 Bits 9:0 | Coefficient for Vertical TAP 2 Phase 6 |
| coefv2_p7 | Payload Word 36 Bits 25:16 | Coefficient for Vertical TAP 2 Phase 7 |
| coefv3_p0 | Payload Word 37 Bits 9:0 | Coefficient for Vertical TAP 3 Phase 0 |
| coefv3_p1 | Payload Word 37 Bits 25:16 | Coefficient for Vertical TAP 3 Phase 1 |
| coefv3_p2 | Payload Word 38 Bits 9:0 | Coefficient for Vertical TAP 3 Phase 2 |
| coefv3_p3 | Payload Word 38 Bits 25:16 | Coefficient for Vertical TAP 3 Phase 3 |
| coefv3_p4 | Payload Word 39 Bits 9:0 | Coefficient for Vertical TAP 3 Phase 4 |
| coefv3_p5 | Payload Word 39 Bits 25:16 | Coefficient for Vertical TAP 3 Phase 5 |
| coefv3_p6 | Payload Word 40 Bits 9:0 | Coefficient for Vertical TAP 3 Phase 6 |
| coefv3_p7 | Payload Word 40 Bits 25:16 | Coefficient for Vertical TAP 3 Phase 7 |
| countvalvert | Payload Word 41 Bits 14:0 | Numerator value for fractional counter countval / 2048 coefv0_p0 |
| fineoffsetvert | Payload Word 41 Bits 30:16 | Count offset for start of line |
| rav_coef | Payload Word 42 Bits 8:0 | Average Filter Coefficient |
| rav_factor | Payload Word 42 Bits 23:16 | Average Filter accumulator averaging factor |
| rav_rowacc | Payload Word 43 Bits 7:0 | Average Accumulator value weight |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.5.1.5.3 Example Graphics List

An example of how a graphics port can be configured will be shown using an example of three frames with the frame size changing. In this example, Figure 12-170 shows the output from the graphics module for the given list. Although in the example it is shown as one list it could just as easily be formatted as one list per frame with each list loaded when the previous list finishes.

**Figure 12-170. Example Graphics Output**






Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-113. Example Graphics List Description**

| Descriptor Type | Description | Size in 128 words |
|---|---|---|
| Configuration Descriptor | Sets up Frame 1 size and Scaler Coeficients must be indirect because of the size | 1 Payload size 12 |
| Data Transfer Descriptor | Sets up up region 1 | 2 |
| Data Transfer Descriptor | Sets up up region 2 size is the pre-scaled size | 2 |
| Data Transfer Descriptor | Sets up region 3 | 2 |
| Configuration Descriptor | Sets up Frame size for Frame 2. Assuming scaler setup stays the same so just a frame configuration is load. It can be direct or indirect. | 2 (Direct) 1 (Indirect) 1 (Payload Size) |
| Data transfer descriptor | Sets up the region 1. Size is the pre-scaled size | 2 |
| Data transfer descriptor | Sets up the stencil data for region 2. | 2 |
| Data transfer descriptor | Sets up the data for region 2 | 2 |
| Data transfer descriptor | Sets up the stencil data for region 1. Size is the pre-scaled size of region 1 | 2 |
| Data transfer descriptor | Sets up the data for region 1 | 2 |
| Data transfer descriptor | Sets up the data for region 2 | 2 |
| Data transfer descriptor | Sets up the data for region 3 | 2 |

After this list is created in memory then the List Address would be written with location of the list. This would be followed with a write to the List Attribute register where a list number and setting the list size to either 23 if the direct list was used for the frame only configuration. If the indirect frame only descriptor is used then the list size would be 22.

#### *12.2.13.5.1.6 List Configuration Example*

To exercise different paths in VPSS, lists are to be prepared in memory to be processed by VPDMA. Consider the Transcode path. The input data is to be read in through Client no:8 (Channel 0 for Luma and Channel 1 for Chroma) and output is to be stored back in memory through client no:3 (Channel 102 YUV interleaved output).

The order to keep descriptors is as follows:

1. Configuration descriptors (To configure the Scaler in the path)
2. Outbound Descriptors
3. Inbound Descriptors


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.2.13.6 Data Formats

Each of the channels has a type of data that it can support based upon the client that it services. Also for each channel, the data type will assume that data is packed in memory a certain way and will be prevented to the client in the same manner to the client no matter what the format of the data in memory.

#### *12.2.13.6.1 YUV Data Formats*

The YUV formatted channels expect video data that is in YUV color space. The YUV data can be type is for YUV data and it can support both interleaved data where Luma and Chroma are in the same data buffer or it can support co-planar data where the Luma and Chroma are in separate data buffers. The YUV channel also can be given a position to give a different start position for the client instead of the default upper left hand corner of the frame. The storage format for YUV data depends on the data type field. The data type must be set to a data type that the channel can support. All the clients that data are provided too will either accept Luma Only, Chroma Only or Luma and Chroma in a parallel data bus.

##### *12.2.13.6.1.1 Y 4:4:4 (Data Type 0)*

The Y 4:4:4 data type is used for a co-planar 4:4:4 data. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container as described in the TILER specification. This data block should have the width and height set to the desired frame size expected by the receiving client.

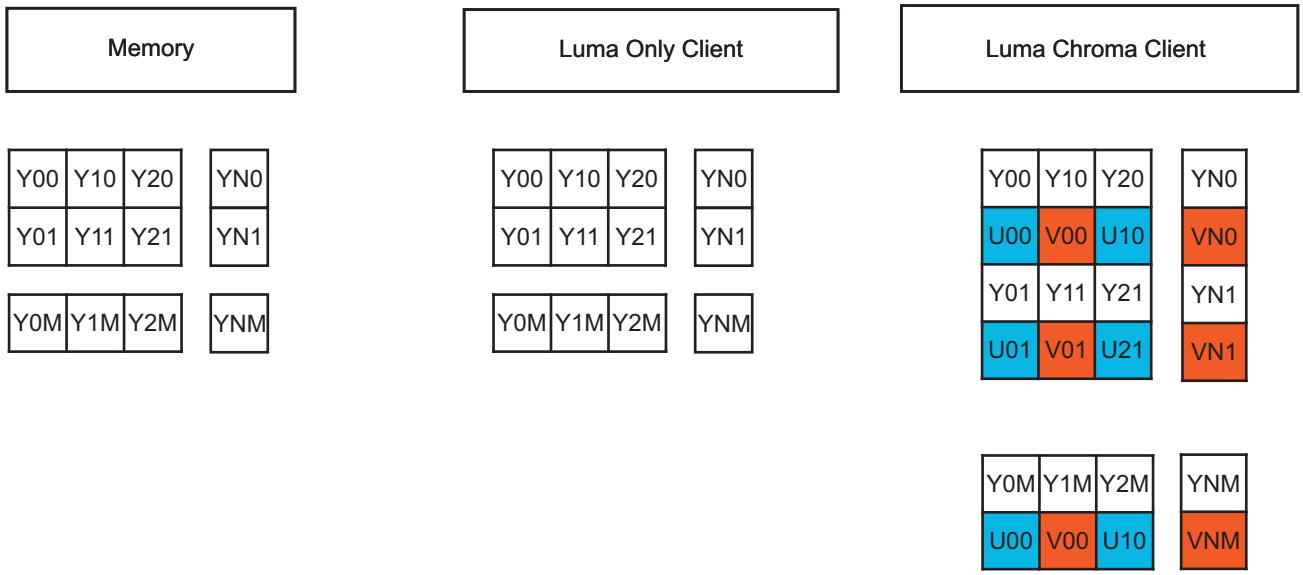
**Figure 12-171. Y 4:4:4 (Data Type 0)**

Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.1.2 Y 4:2:2 (Data Type 1)

The Y 4:2:2 data type is used for a co-planar 4:2:2 data. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container as described in the TILER specification. This data block should have the width and the height of the desired frame sent by the VPDMA to the receive client.

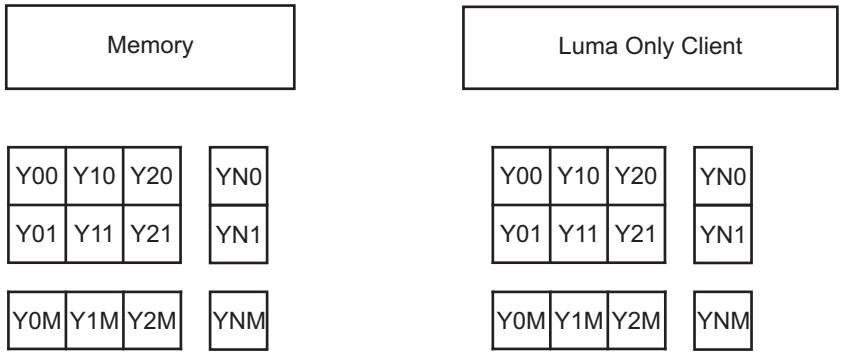
**Figure 12-172. Y 4:2:2 (Data Type 1)**




#### 12.2.13.6.1.3 Y 4:2:0 (Data Type 2)

The Y 4:2:0 data type is used for a co-planar 4:2:0 data type. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container as described in the TILER specification. This data block should have the width and the height of the expected frame for the client.

**Figure 12-173. Y 4:2:0 (Data Type 2)**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.1.4 C 4:4:4 (Data Type 4)

The C 4:4:4 data type is used for a co-planar 4:4:4 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container as described in the TILER specification. This data block should have the width and height of the expected client frame.

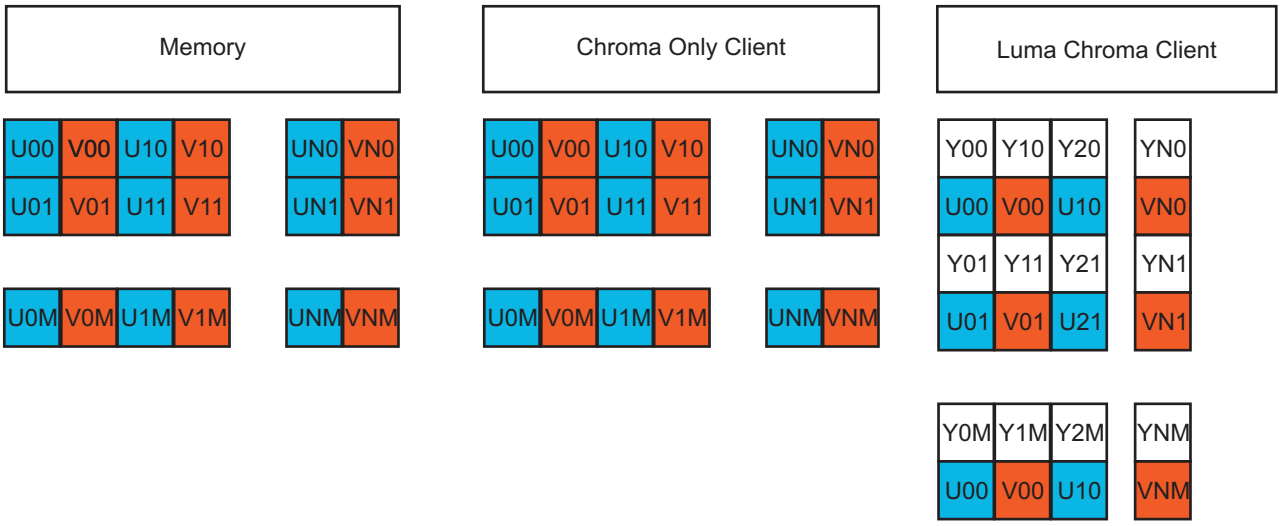
**Figure 12-174. C 4:4:4 (Data Type 4)**

Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.1.5 C 4:2:2 (Data Type 5)

The C 4:2:2 data type is used for co-planar 4:2:2 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container as described in the TILER specification. This data block should have the width and the height of the expected client frame.

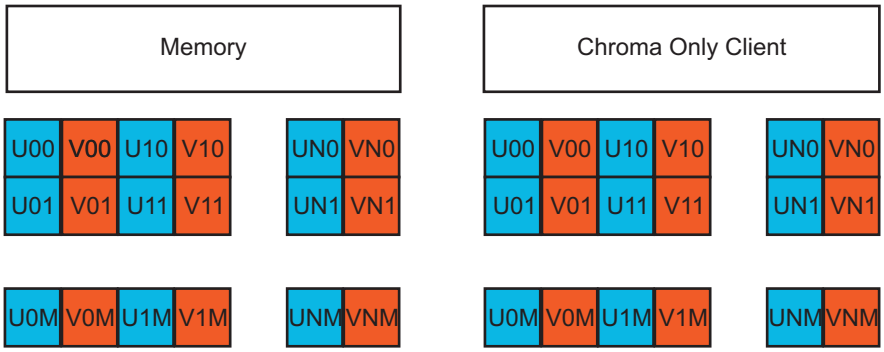
**Figure 12-175. C 4:2:2 (Data Type 5)**




#### 12.2.13.6.1.6 C 4:2:0 (Data Type 6)

The C 4:2:0 data type is used for a co-planar 4:2:0 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in a 16 bit container as described in the TILER specification. This data block should have the width and half the height of the expected clients frame.

**Figure 12-176. C 4:2:0 (Data Type 6)**



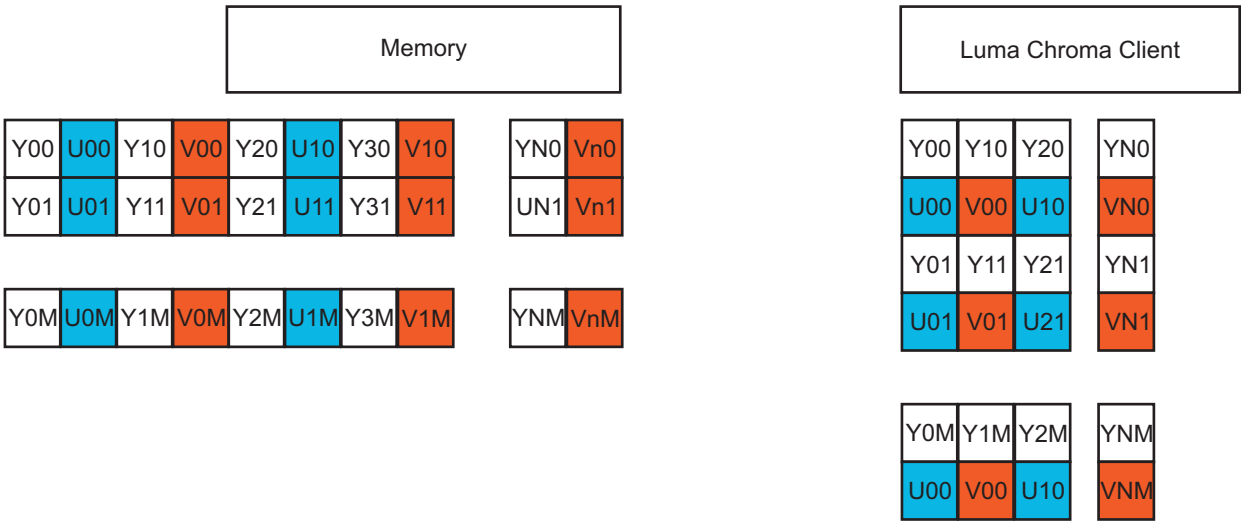


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.1.7 YC 4:2:2 (Data Type 7)

The YC 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Y0 in the lowest byte followed by Cb followed by Y1 finally Cr in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container as described in the TILER specification. The data block width and height should be set the same as the expected client.

**Figure 12-177. YC 4:2:2 (Data Type 7)**



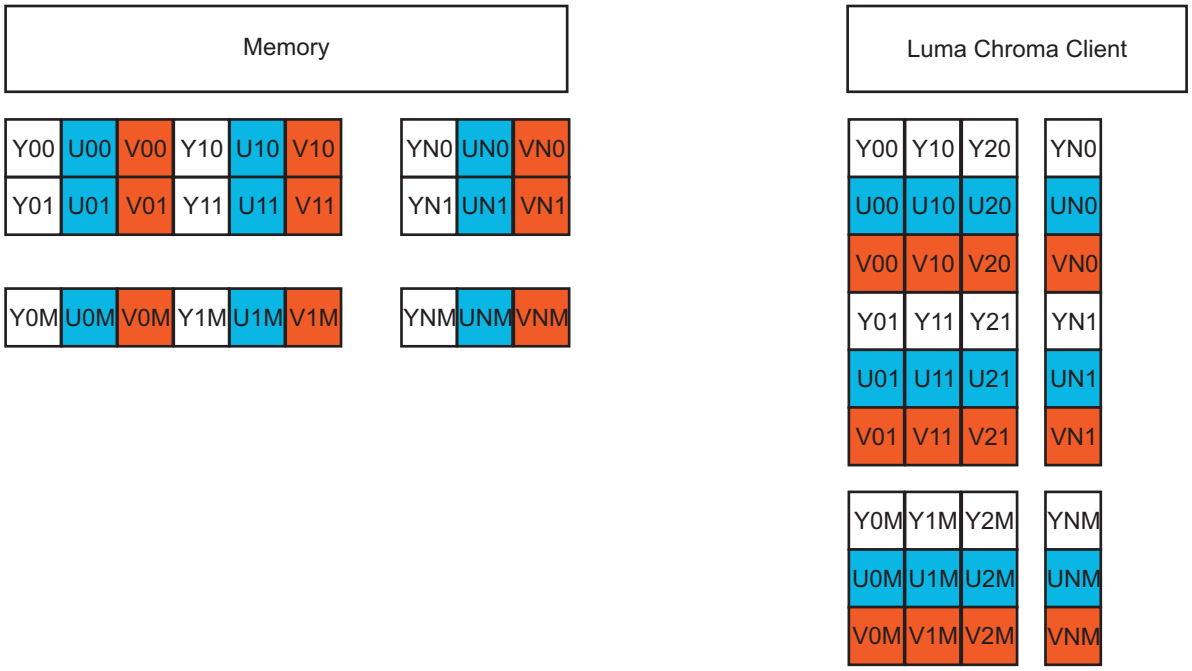


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.1.8 YC 4:4:4 (Data Type 8)

The YC 4:4:4 data type is used for interleaved 4:4:4 data. It is expected to be packed with Y0 in the lowest byte followed by Cr followed by Cb. The transfer counts each YCrCb triplet in a 24 bit word as a pixel. A 2D transfer of the data is NOT supported in the VPDMA. The data block width and height should be set to the number of pixels and lines that are expected by the client.

**Figure 12-178. YC 4:4:4 (Data Type 8)**



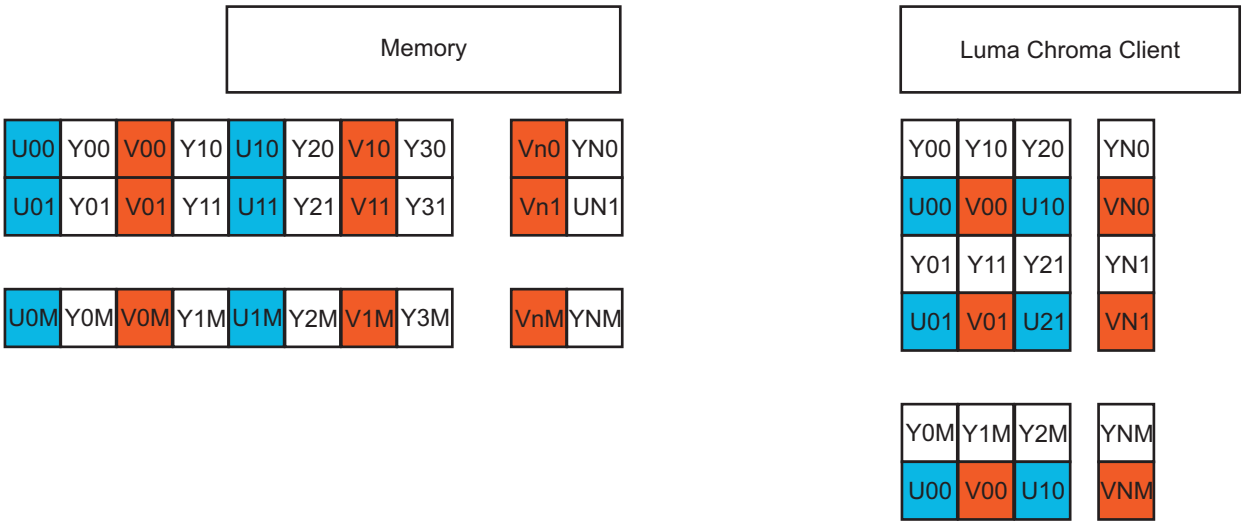


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.1.9 CY 4:2:2 (Data Type 23h)

The CY 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Cb in the lowest byte followed by Y0 followed by Cr finally Y1 in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container as described in the TILER specification. The data block width and height should be set the same as the expected client.

**Figure 12-179. CY 4:2:2 (Data Type 23h)**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.1.10 Cb 4:4:4 (Data Type 14h)

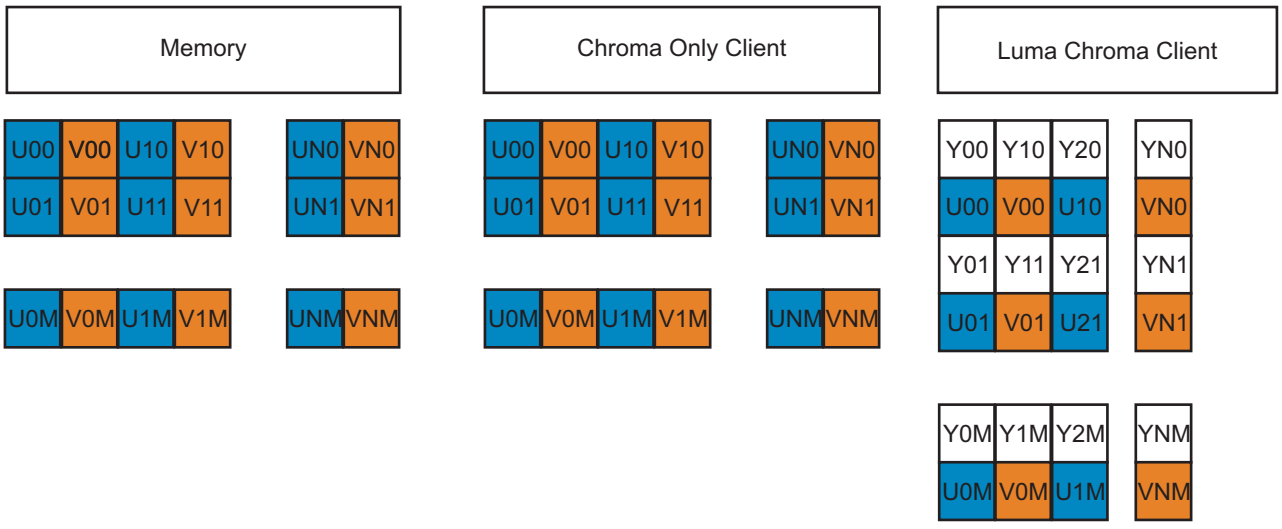
The Cb 4:4:4 data type is used for a co-planar 4:4:4 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container as described in the TILER specification. This data block should have the width and height of the expected client frame.

**Figure 12-180. Cb 4:4:4 (Data Type 14h)**

#### 12.2.13.6.1.11 Cb 4:2:2 (Data Type 15h)

The C 4:2:2 data type is used for co-planar 4:2:2 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container as described in the TILER specification. This data block should have the width and the height of the expected client frame.

**Figure 12-181. Cb 4:2:2 (Data Type 15h)**



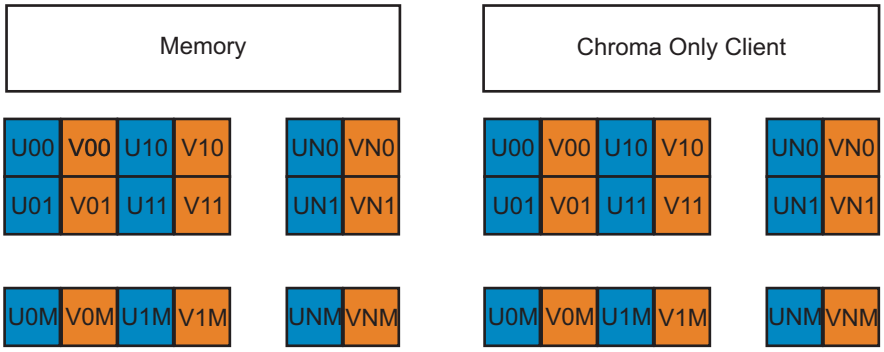


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.1.12 Cb 4:2:0 (Data Type 16h)

The Cb 4:2:0 data type is used for a co-planar 4:2:0 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in a 16 bit container as described in the TILER specification. This data block should have the width and half the height of the expected clients frame.

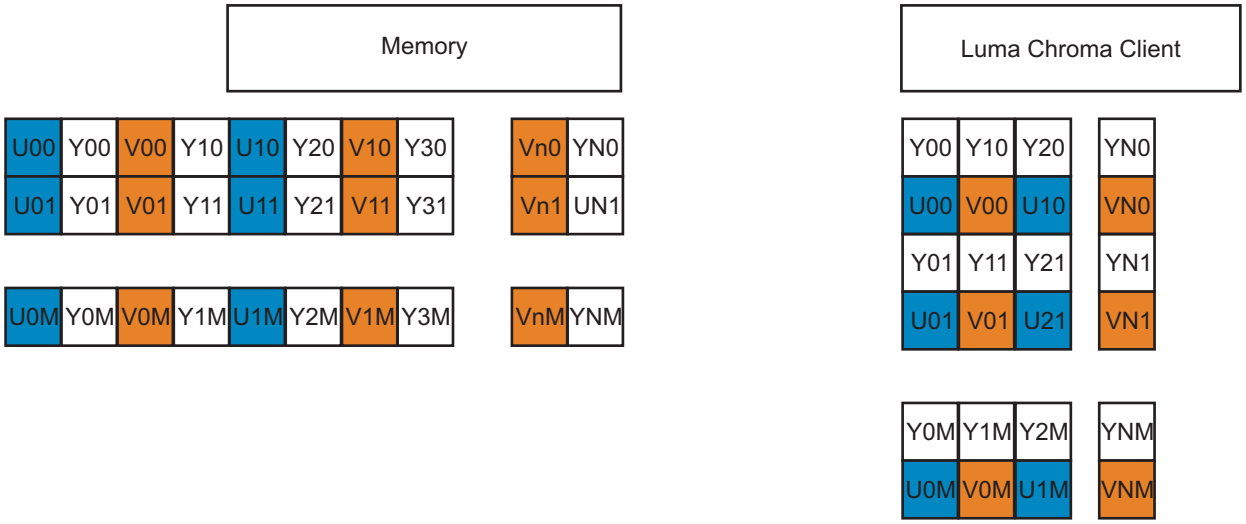
**Figure 12-182. Cb 4:2:0 (Data Type 16h)**




#### 12.2.13.6.1.13 CbY 4:2:2 (Data Type 17h)

The CY 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Cb in the lowest byte followed by Y0 followed by Cr finally Y1 in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container as described in the TILER specification. The data block width and height should be set the same as the expected client.

**Figure 12-183. CbY 4:2:2 (Data Type 17h)**



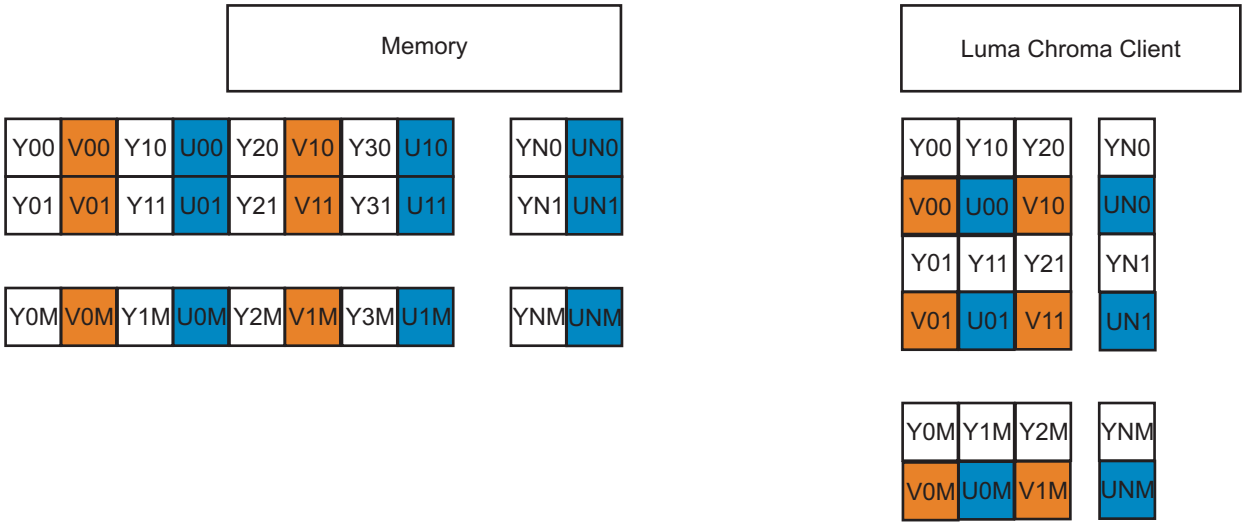


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.1.14 YC 4:2:2 (Data Type 27h)

The YC 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Y0 in the lowest byte followed by Cr followed by Y1 finally Cb in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container as described in the TILER specification. The data block width and height should be set the same as the expected client.

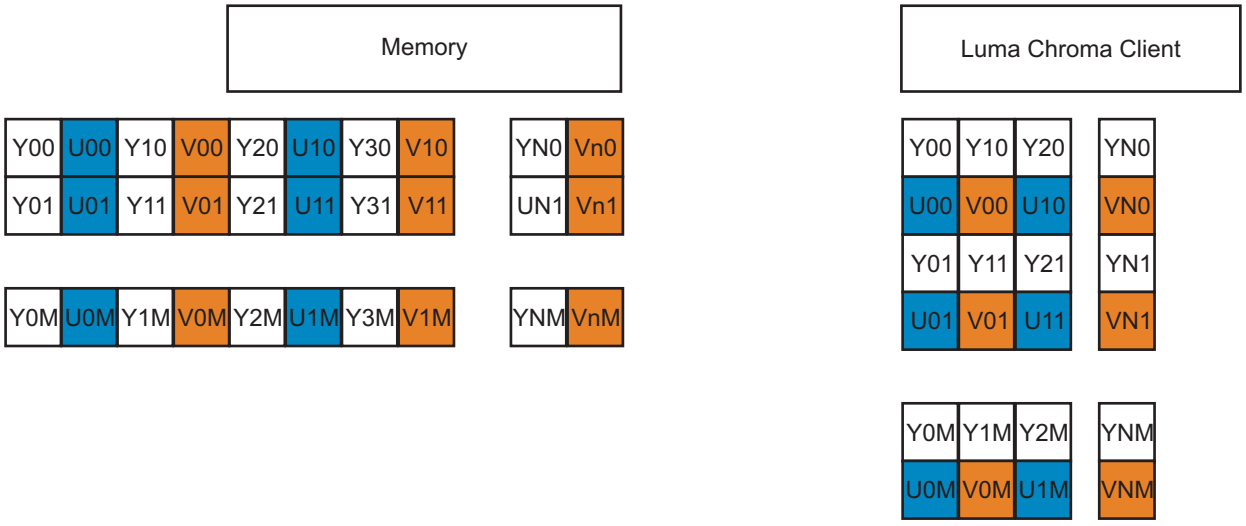
**Figure 12-184. YC 4:2:2 (Data Type 27h)**




#### 12.2.13.6.1.15 YCb 4:2:2 (Data Type 37h)

The YCb 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Y0 in the lowest byte followed by Cb followed by Y1 finally Cr in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container as described in the TILER specification. The data block width and height should be set the same as the expected client.

**Figure 12-185. YC 4:2:2 (Data Type 37h)**





Copyright © 2011–2016, Texas Instruments Incorporated

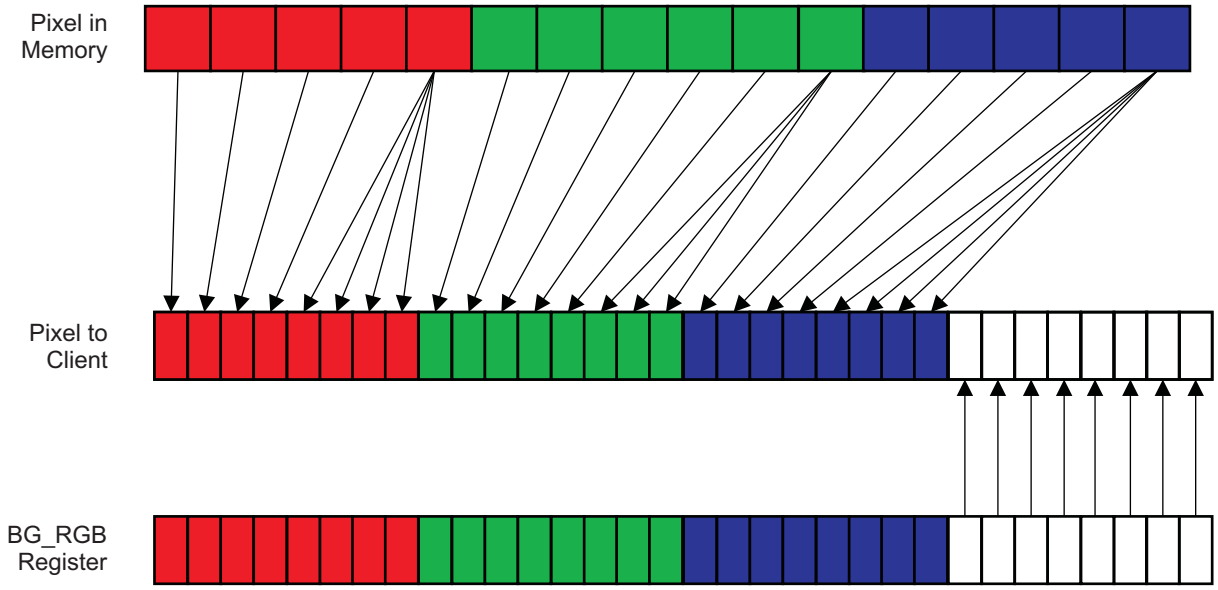
#### 12.2.13.6.2 RGB Data Formats

The RGB channel type is used to provide data for a client that expects to transmit or receive RGB data. If a channel supports it also bit map data can be used where the pixel value of the data can be used as an address for a Color Look-up Table (CLUT) to provide a full 32 bit RGBA data stream to the client. In all modes the client is provided RGBA 8888 data. The lower bits if not provided by the data stream are a replication of the lower bit of the data. For outbound data the input is assumed to be RGB 888 and the Alpha value is taken from the Background Color register to make a full ARGB 8888. The lower bits are dropped from this data if a data type specifies less then the full 8 bits per color.

##### 12.2.13.6.2.1 RGB16-565 (Data Type 0)

In RGB16-565 mode, each pixel is a single RGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the middle 6 bits for green data and the upper 5 bits for red data.

**Figure 12-186. RGB16-565 (Data Type 0)**



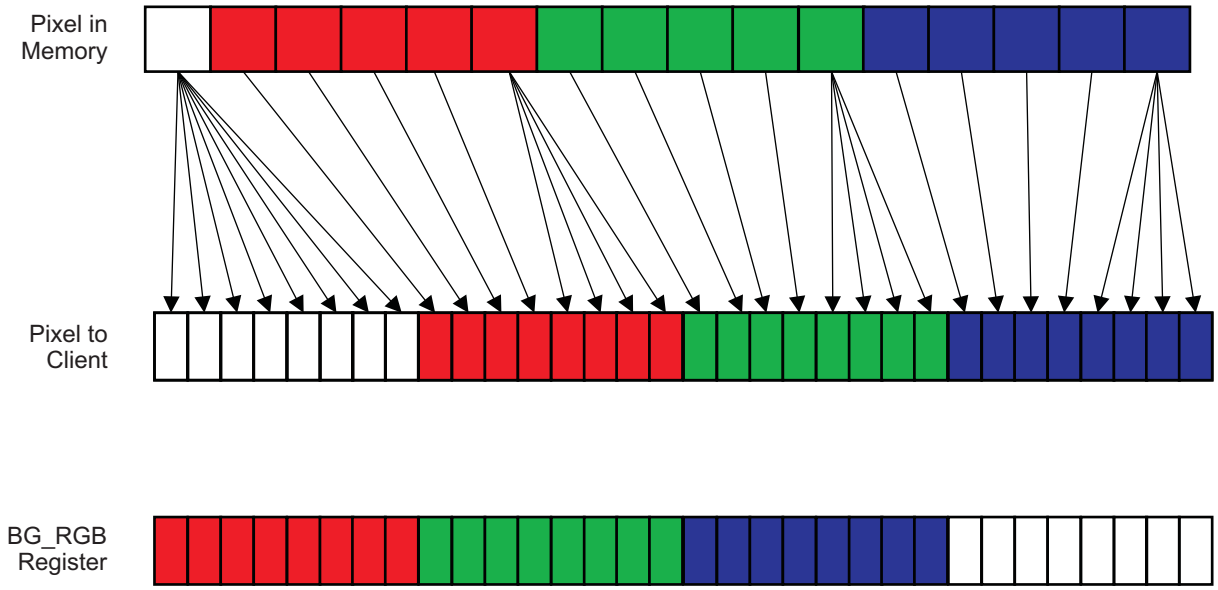


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.2 ARGB-1555 (Data Type 1)

In ARGB-1555 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data and the upper most bit for the blend value to use 0 or 0xff.

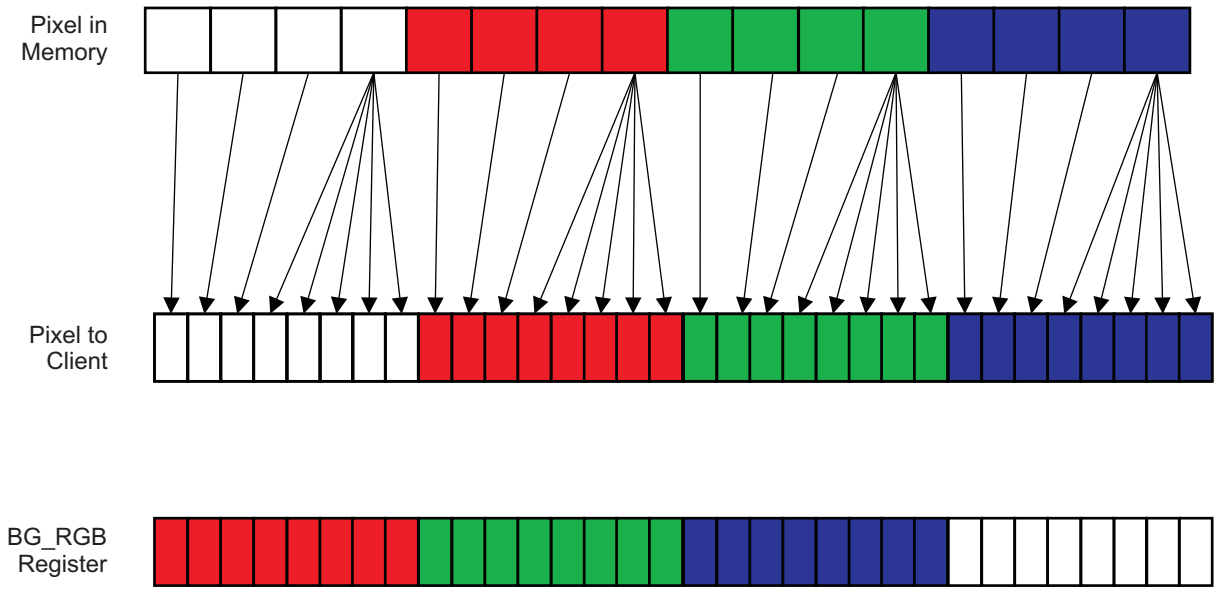
**Figure 12-187. ARGB-1555 (Data Type 1)**




#### 12.2.13.6.2.3 ARGB-4444 (Data Type 2)

In ARGB16-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for blue data, the next 4 bits for green data the next 4 bits for red data and the upper most 4 bits for the blend value.

**Figure 12-188. ARGB-4444 (Data Type 2)**



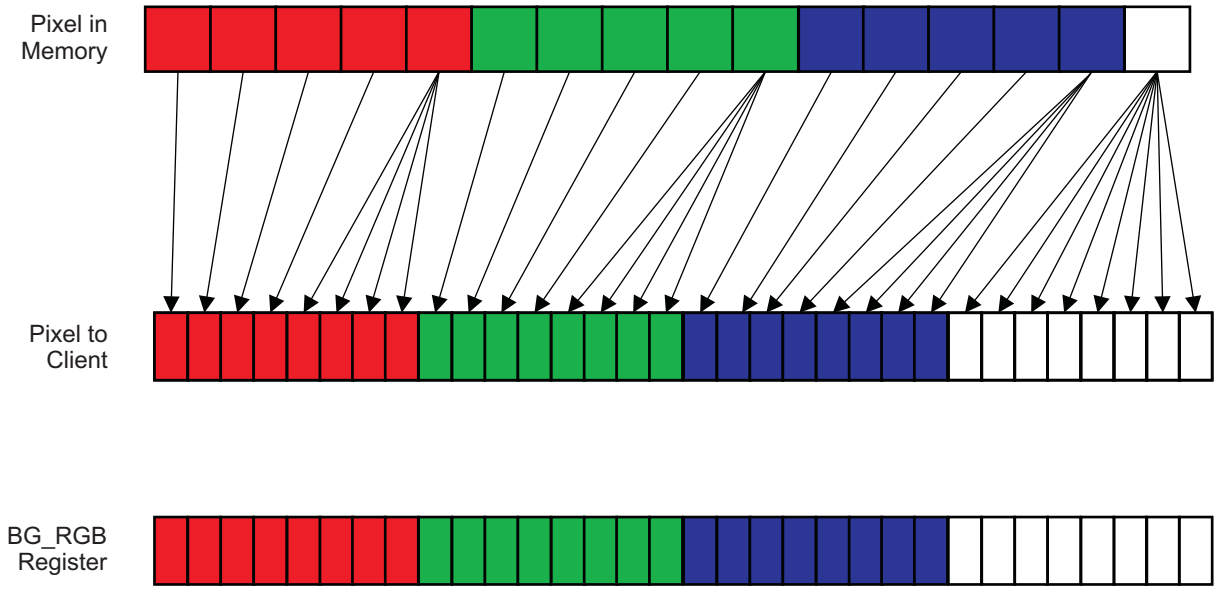


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.4 RGBA-5551 (Data Type 3)

In RGBA-5551 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest most bit for the blend value to use 0 or 0xff the next 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data.

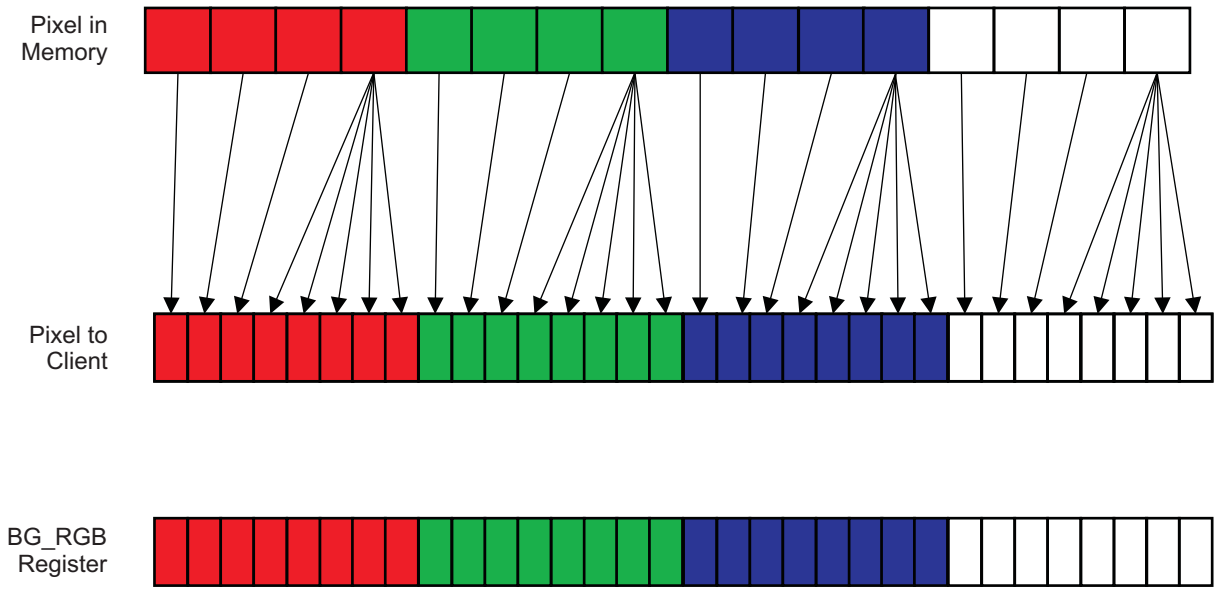
**Figure 12-189. RGBA-5551 (Data Type 3)**




#### 12.2.13.6.2.5 RGBA-4444 (Data Type 4)

In RGBA-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for the blend value, the next 4 bits for blue data, the next 4 bits for green data and the upper most 4 bits for red data.

**Figure 12-190. RGBA-4444 (Data Type 4)**



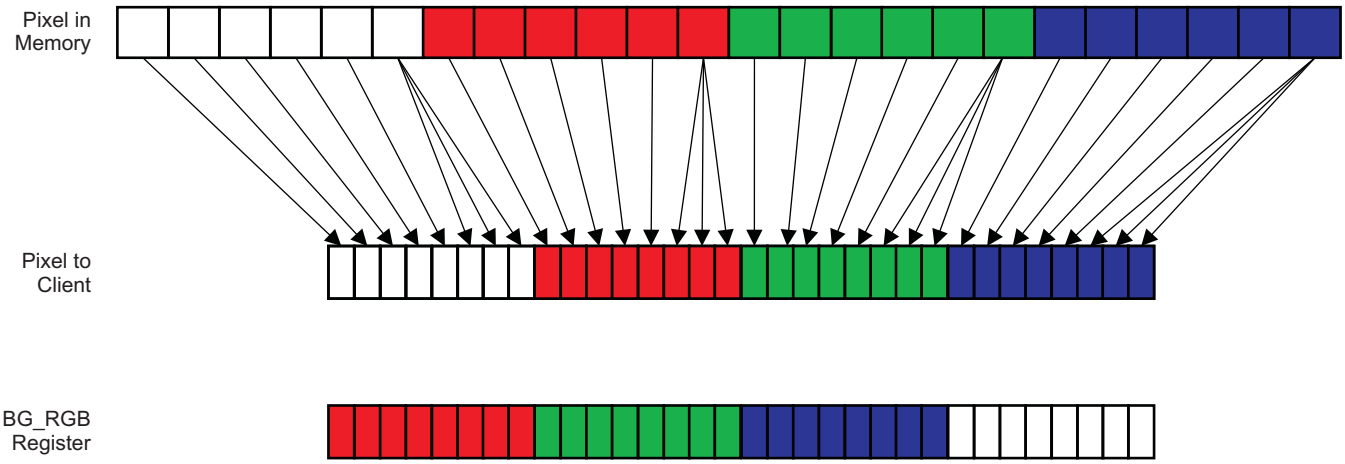


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.6 ARGB24-6666 (Data Type 5)

In ARGB24-6666 mode, each pixel is a single ARGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper most 6 bits for red data.

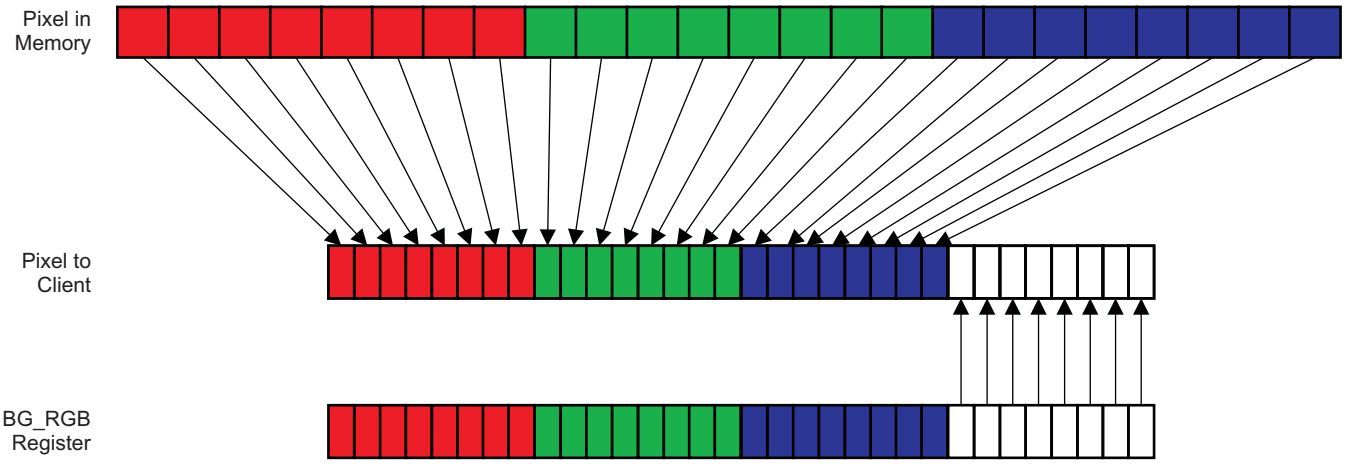
**Figure 12-191. ARGB24-6666 (Data Type 5)**




#### 12.2.13.6.2.7 RGB24-888 (Data Type 6)

In RGB24-888 mode, each pixel is a single RGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data it uses the BG_RGB Blend value for the Blend value.

**Figure 12-192. RGB24-888 (Data Type 6)**



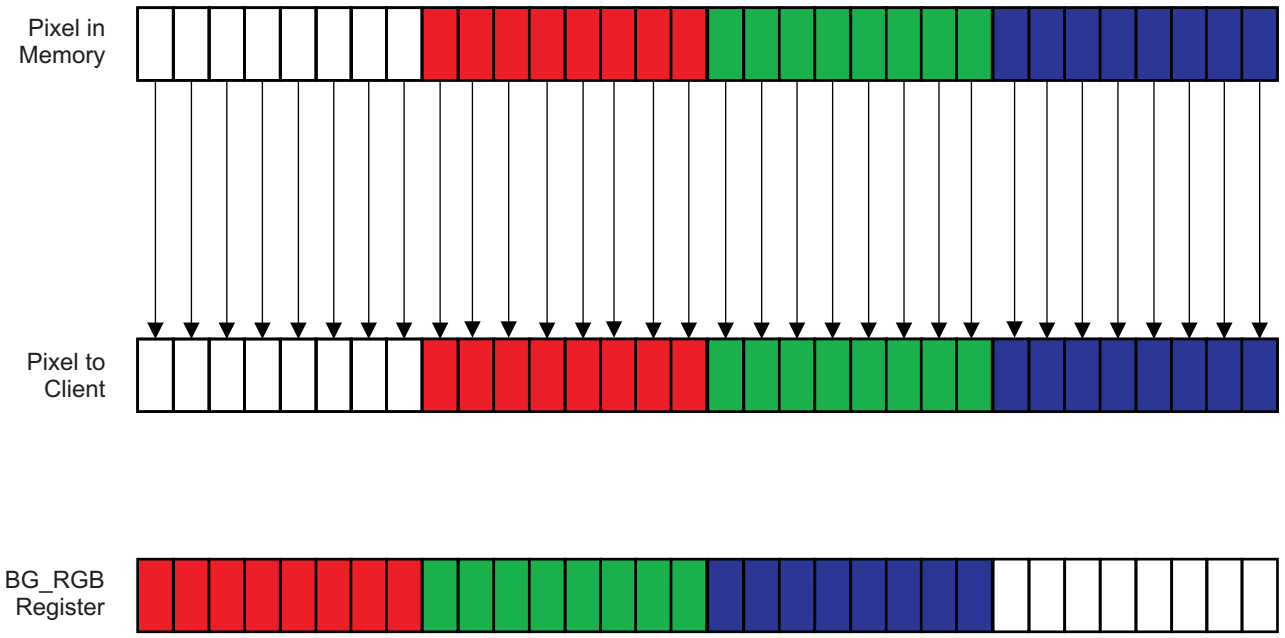


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.8 ARGB32-8888 (Data Type 7)

In ARGB32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bitsof data uses the lowest 8 bits for blue data, the next 8 bits for green data and the next 8 bits for red data it uses the upper most bits for the blend value.

**Figure 12-193. ARGB32-8888 (Data Type 7)**



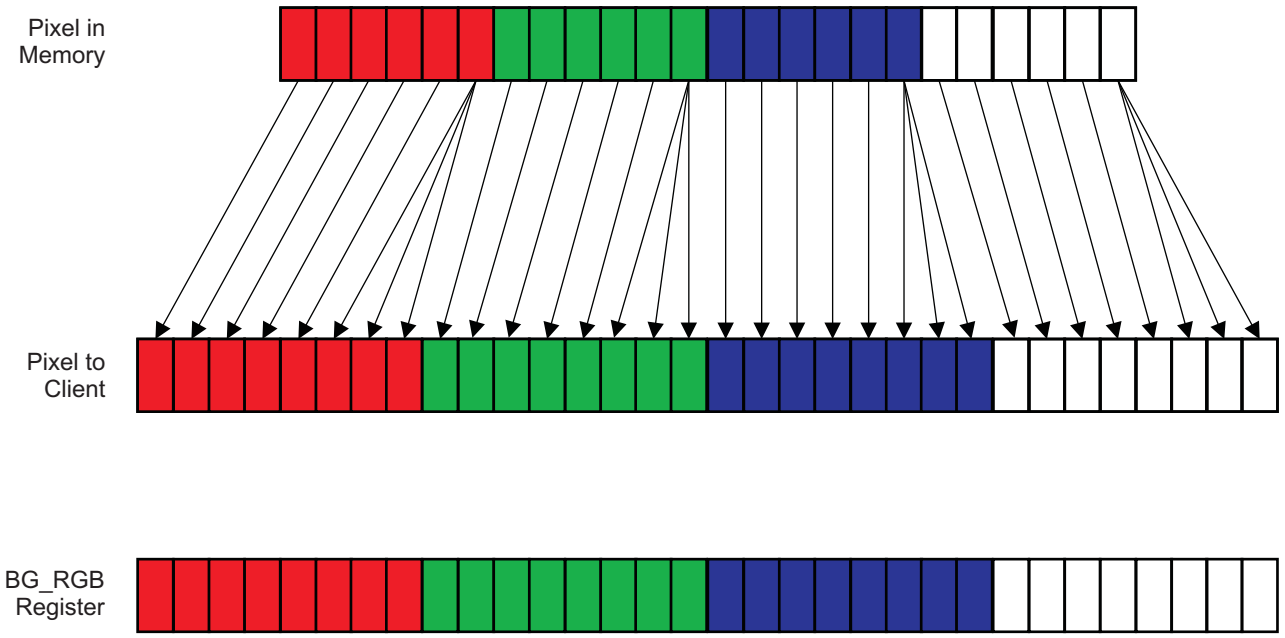


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.9 RGBA24-6666 (Data Type 8)

In RGBA24-6666 mode, each pixel is a single RGBA pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper 6 bits for red data.

**Figure 12-194. RGBA24-6666 (Data Type 8)**



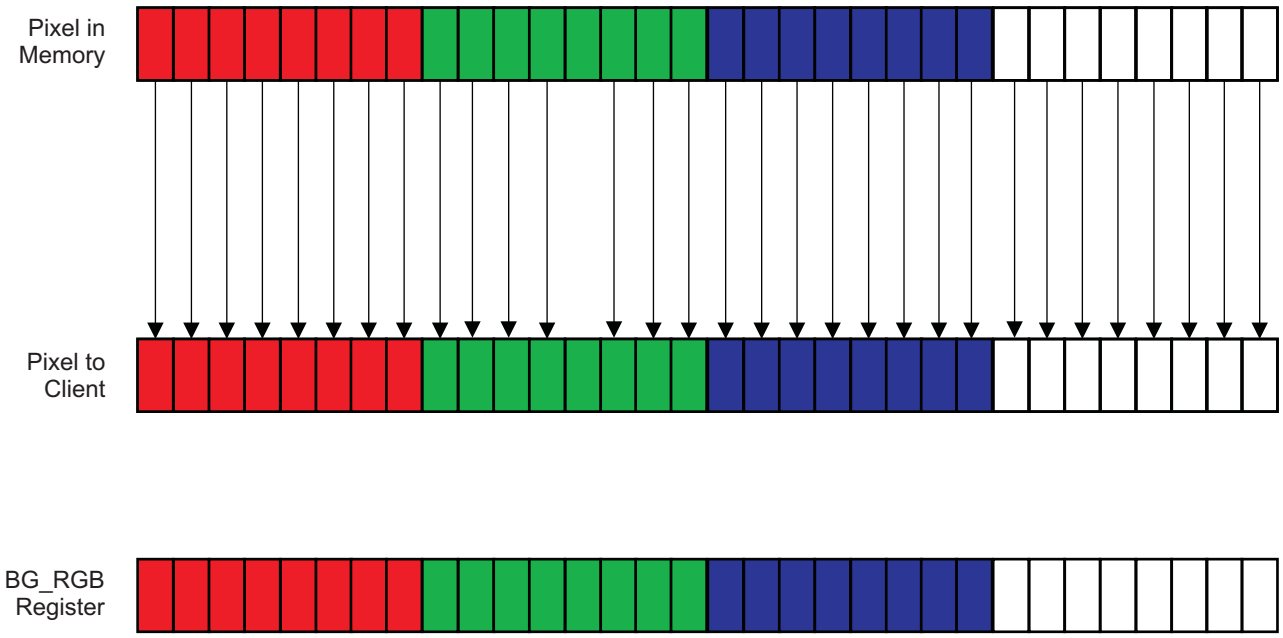


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.10 RGBA32-8888 (Data Type 9)

In RGBA32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bits of data uses the lowest 8 bits for the blend value the next 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data.

**Figure 12-195. RGBA32-8888 (Data Type 9)**



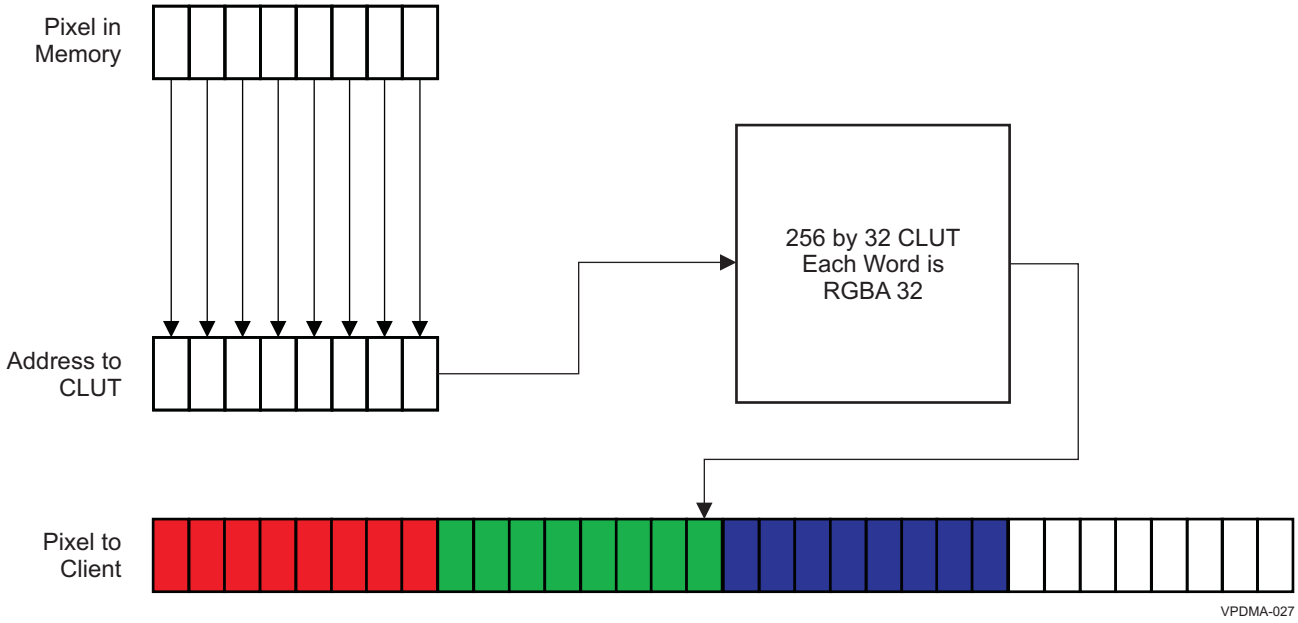



Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.11 Bitmap-8 (Data Type 20h)

In Bitmap-8 data, each pixel is an 8 bit value that is used as an address into a 256 × 32 memory that gives the final RGBA32-8888 data value.

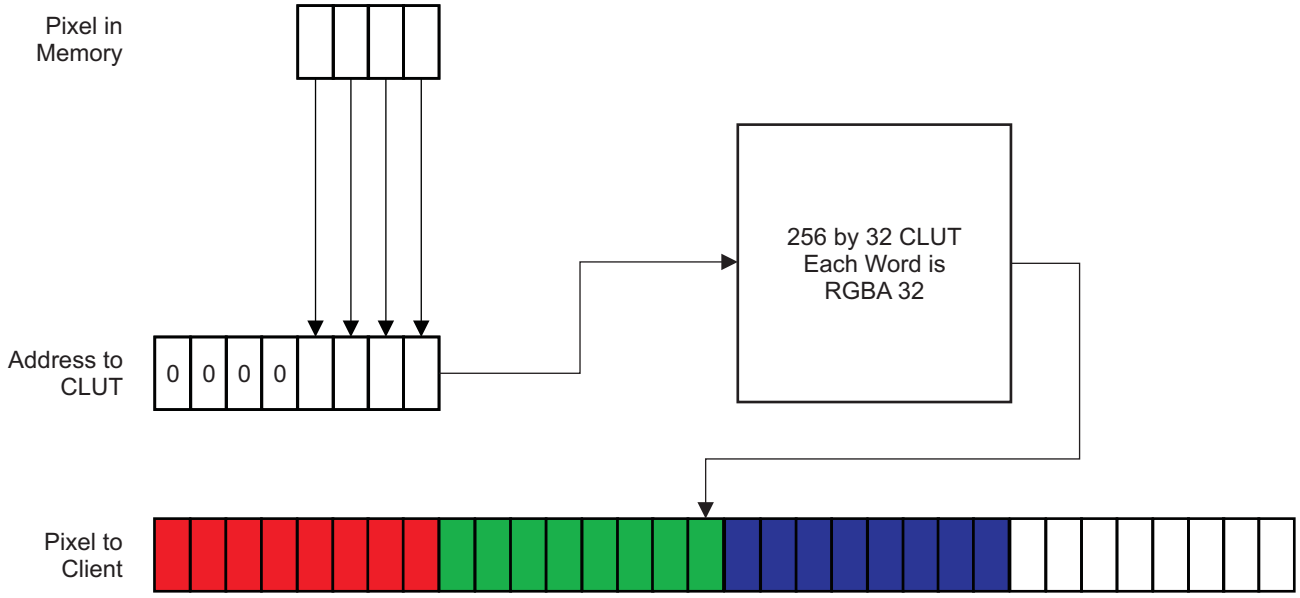
**Figure 12-196. Bitmap-8 (Data Type 20h)**




#### 12.2.13.6.2.12 Bitmap-4 Lower (Data Type 22h)

In Bitmap-4 Lower data, each pixel is an 4 bit value that is used as the lower 4 bits of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 4 bits in this case are tied to 0.

**Figure 12-197. Bitmap-4 Lower (Data Type 22h)**



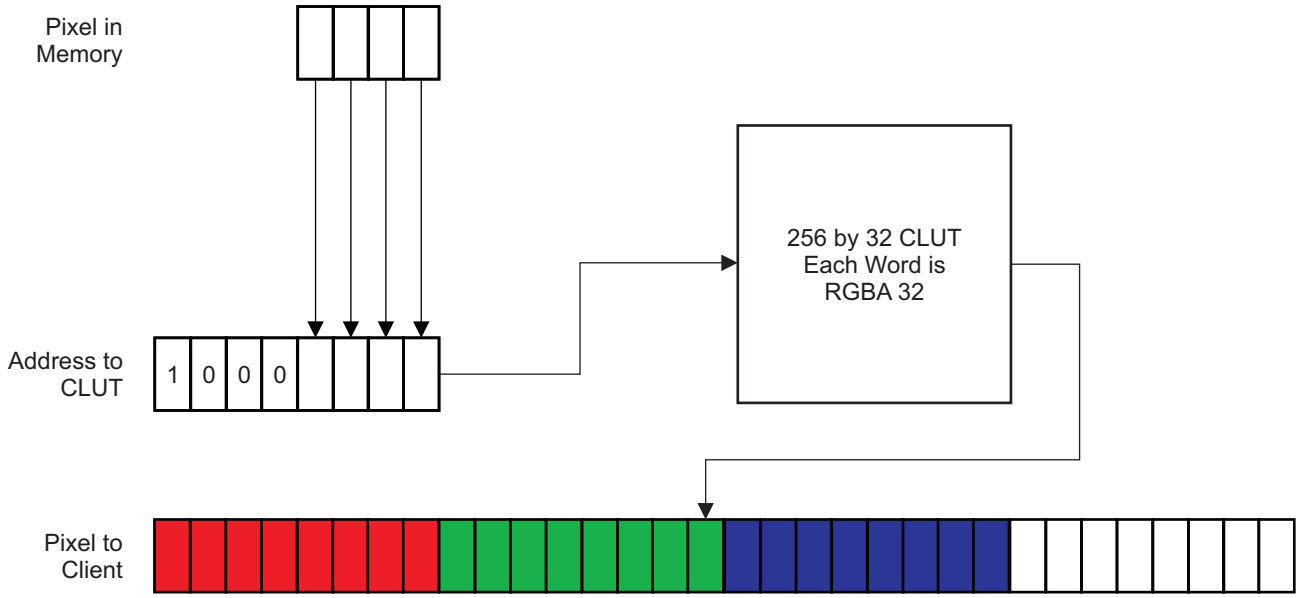


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.13 Bitmap-4 Upper (Data Type 23h)

In Bitmap-4 Upper data, each pixel is an 4 bit value that is used as the lower 4 bits of address into a 256× 32 memory that gives the final RGBA32-8888 data value. The upper 4 bits in this case are tied to 4'b1000.

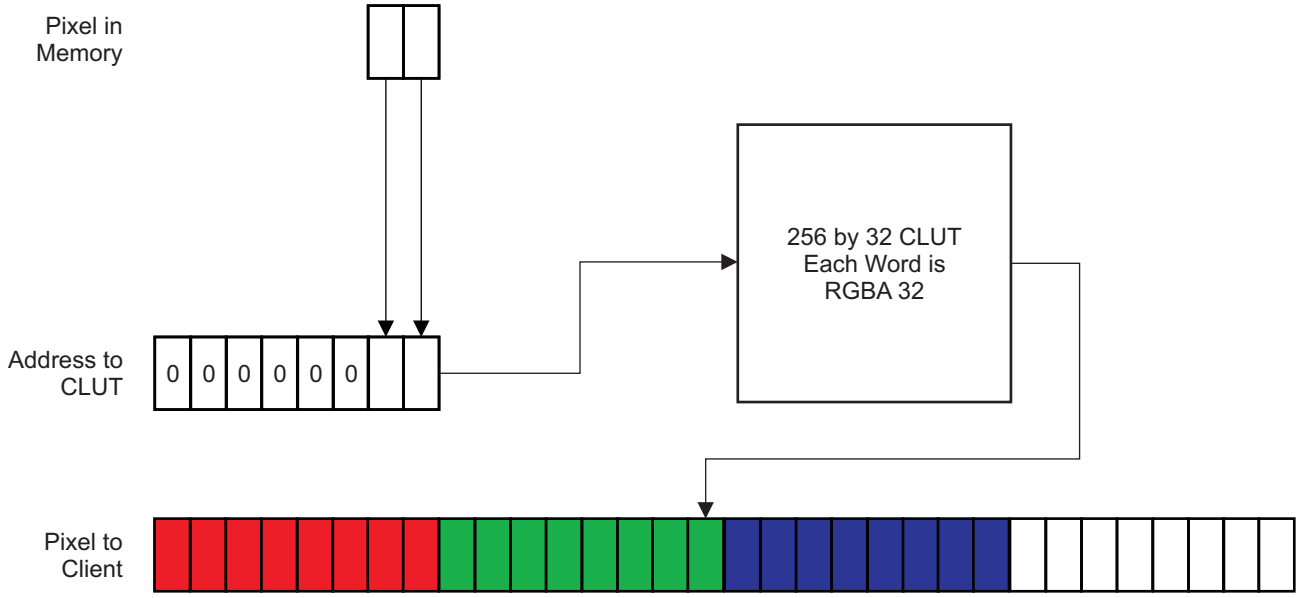
**Figure 12-198. Bitmap-4 Upper (Data Type 23h)**




#### 12.2.13.6.2.14 Bitmap-2 Offset0 (Data Type 24h)

In Bitmap-2 Offset0 data, each pixel is a 2 bit value that is used as the lower 2 bits of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 2 bits in this case are tied to 0. Bits 2-5 are tied to 0.

**Figure 12-199. Bitmap-2 Offset0 (Data Type 24h)**



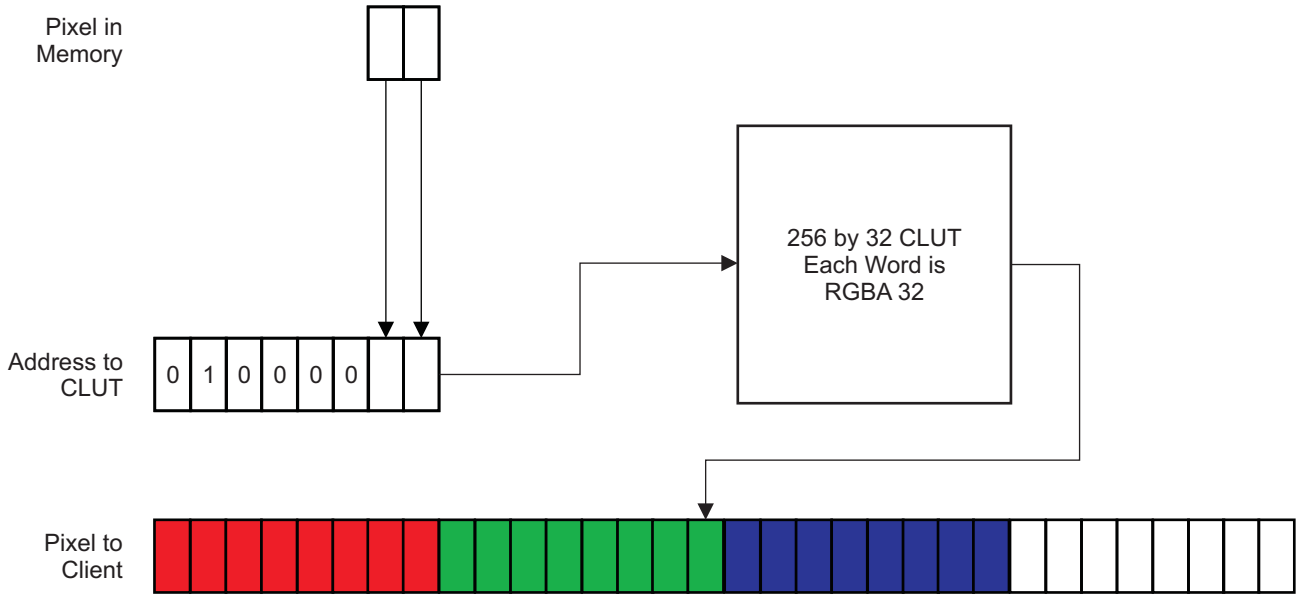


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.15 Bitmap-2 Offset1 (Data Type 25h)

In Bitmap-2 Offset1 data, each pixel is a 2 bit value that is used as the lower 2 bits of address into a 256× 32 memory that gives the final RGBA32-8888 data value. The upper 2 bits in this case are tied to 2'b01. Bits 2-5 are tied to 0.

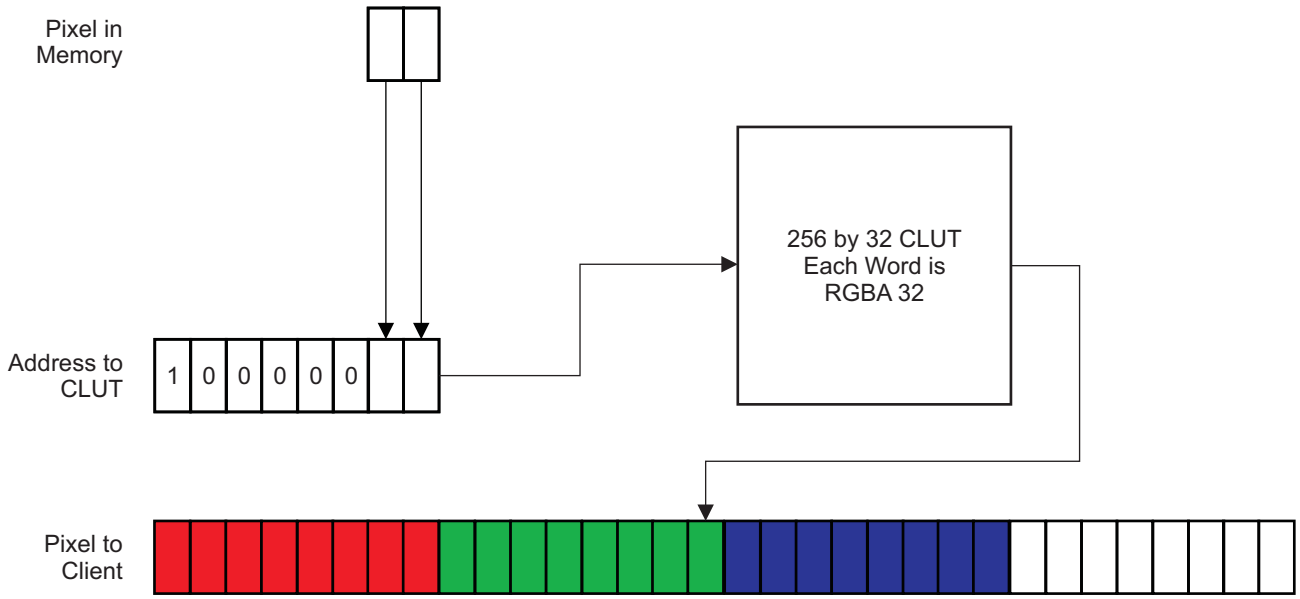
**Figure 12-200. Bitmap-2 Offset1 (Data Type 25h)**




#### 12.2.13.6.2.16 Bitmap-2 Offset2 (Data Type 26h)

In Bitmap-2 Offset2 data, each pixel is a 2 bit value that is used as the lower 2 bits of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 2 bits in this case are tied to 2'10. Bits 2-5 are tied to 0.

**Figure 12-201. Bitmap-2 Offset2 (Data Type 26h)**



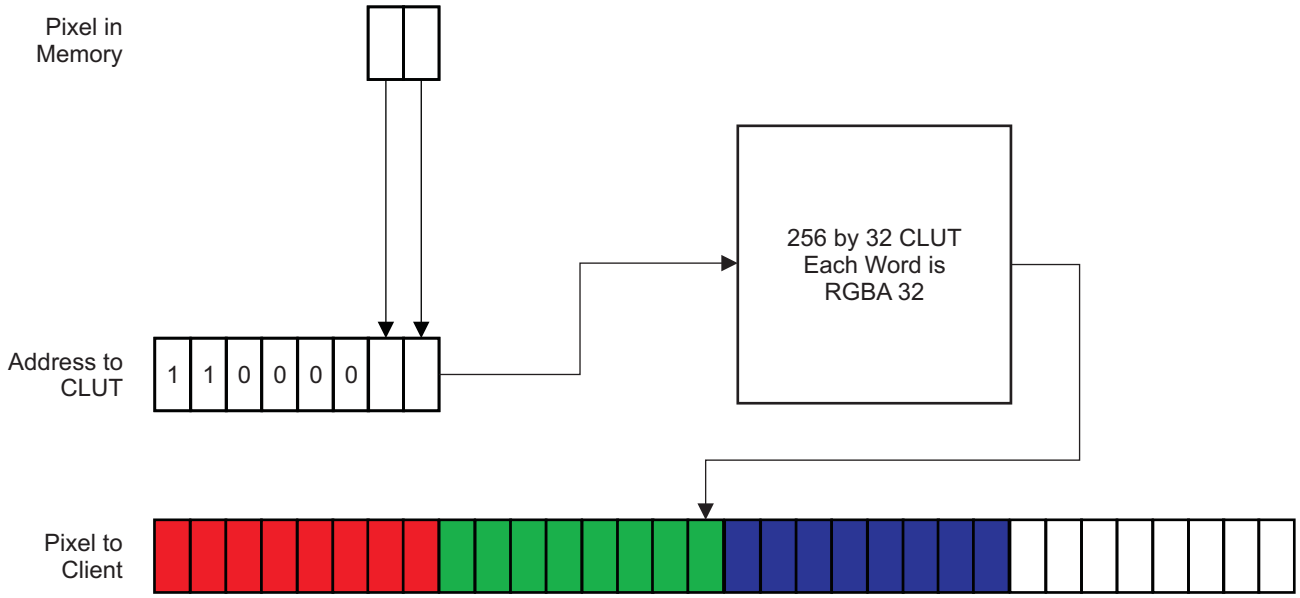


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.17 Bitmap-2 Offset3 (Data Type 27h)

In Bitmap-2 Offset3 data, each pixel is a 2 bit value that is used as the lower 2 bits of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 2 bits in this case are tied to 2'b11. Bits 2-5 are tied to 0.

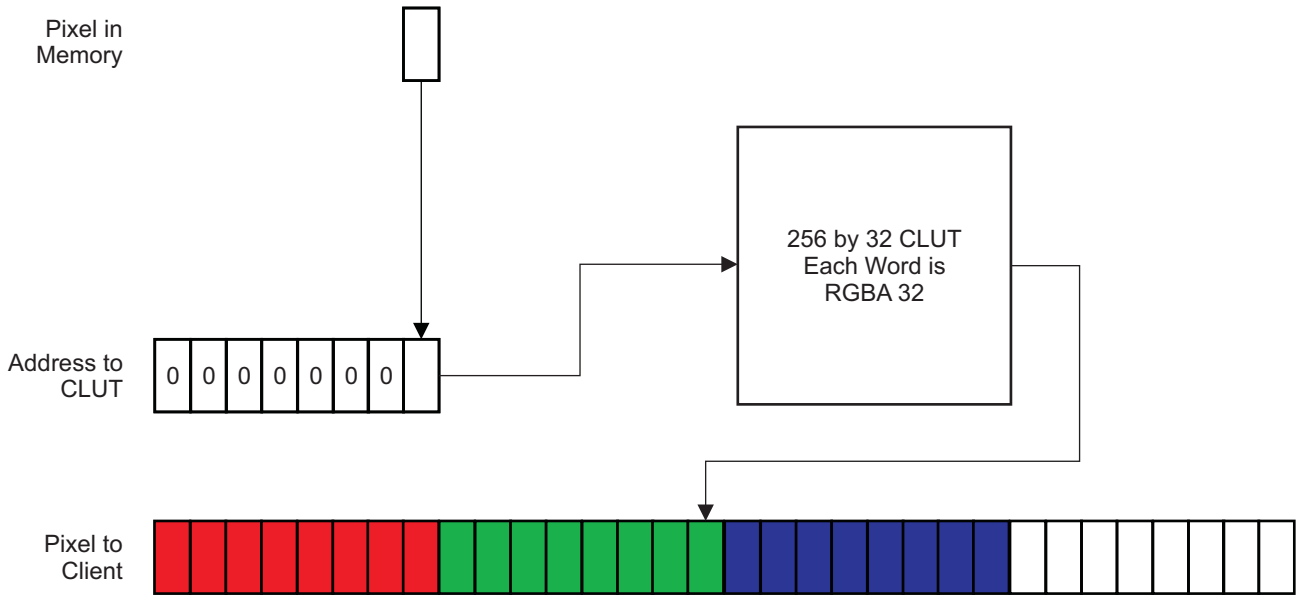
**Figure 12-202. Bitmap-2 Offset3 (Data Type 27h)**




#### 12.2.13.6.2.18 Bitmap-1 Offset0 (Data Type 28h)

In Bitmap-1 Offset0 data, each pixel is an 1 bit value that is used as the lower bit of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 3 bits in this case are tied to 3'b000. Bits 1-4 are tied to 0.

**Figure 12-203. Bitmap-1 Offset0 (Data Type 28h)**



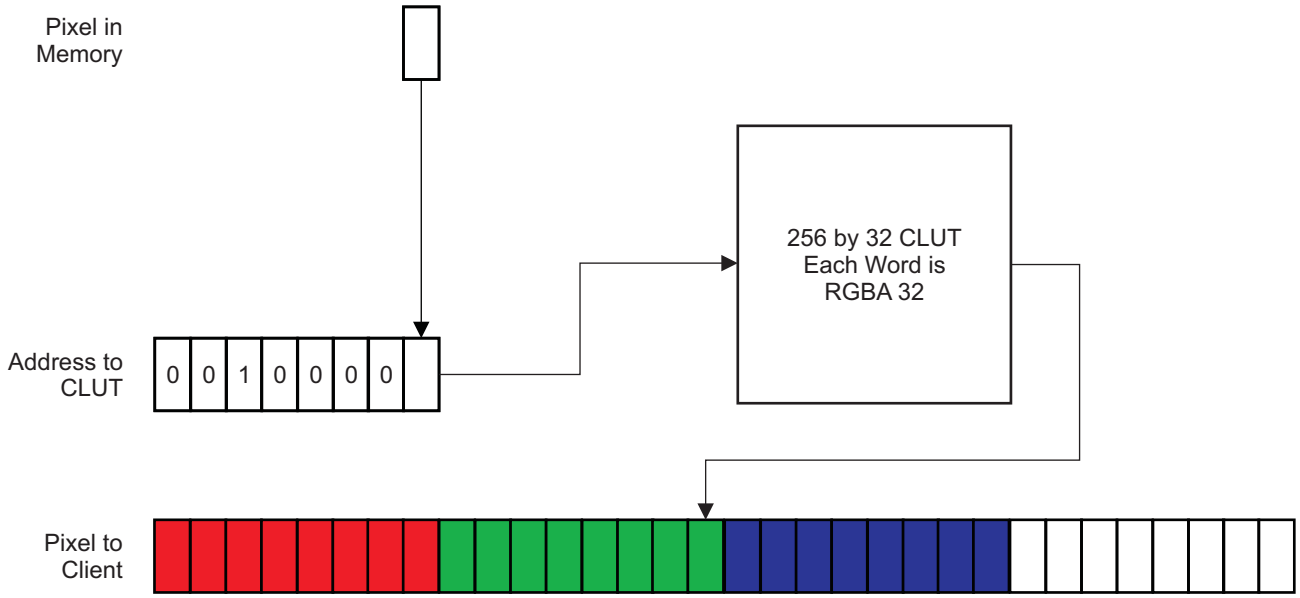


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.19 Bitmap-1 Offset1 (Data Type 29h)

In Bitmap-1 Offset1 data, each pixel is an 1 bit value that is used as the lower bit of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 3 bits in this case are tied to 3'b001. Bits 1-4 are tied to 0.

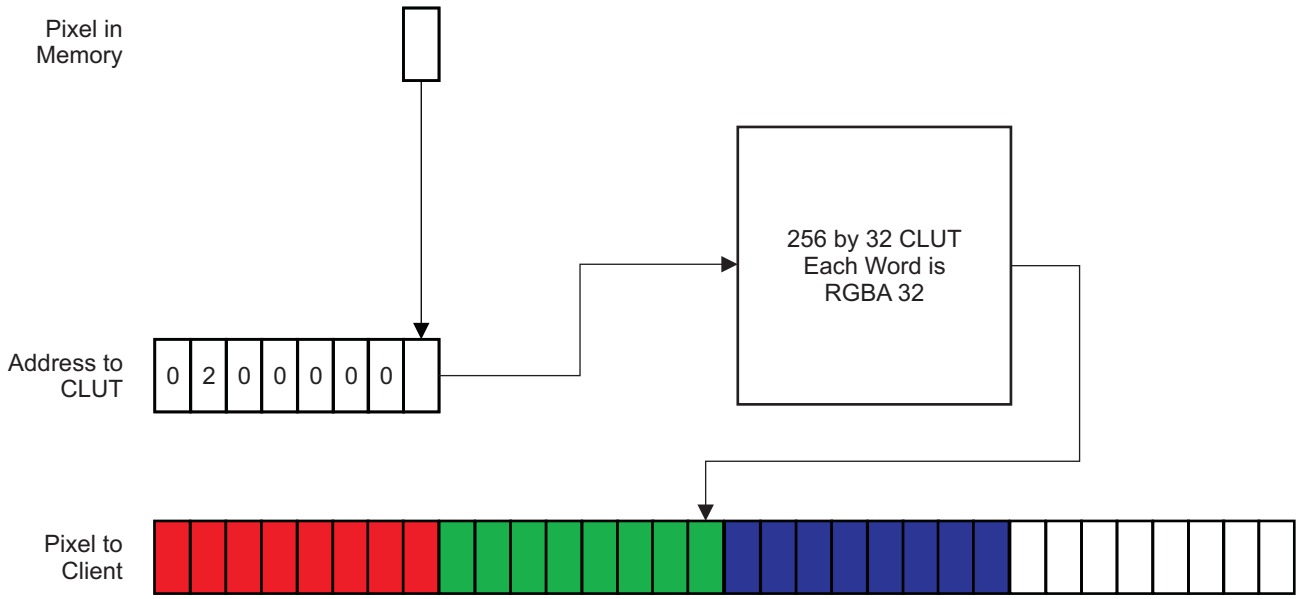
**Figure 12-204. Bitmap-1 Offset1 (Data Type 29h)**




#### 12.2.13.6.2.20 Bitmap-1 Offset2 (Data Type 2Ah)

In Bitmap-1 Offset2 data, each pixel is an 1 bit value that is used as the lower bit of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 3 bits in this case are tied to 3'b010. Bits 1-4 are tied to 0.

**Figure 12-205. Bitmap-1 Offset2 (Data Type 2Ah)**



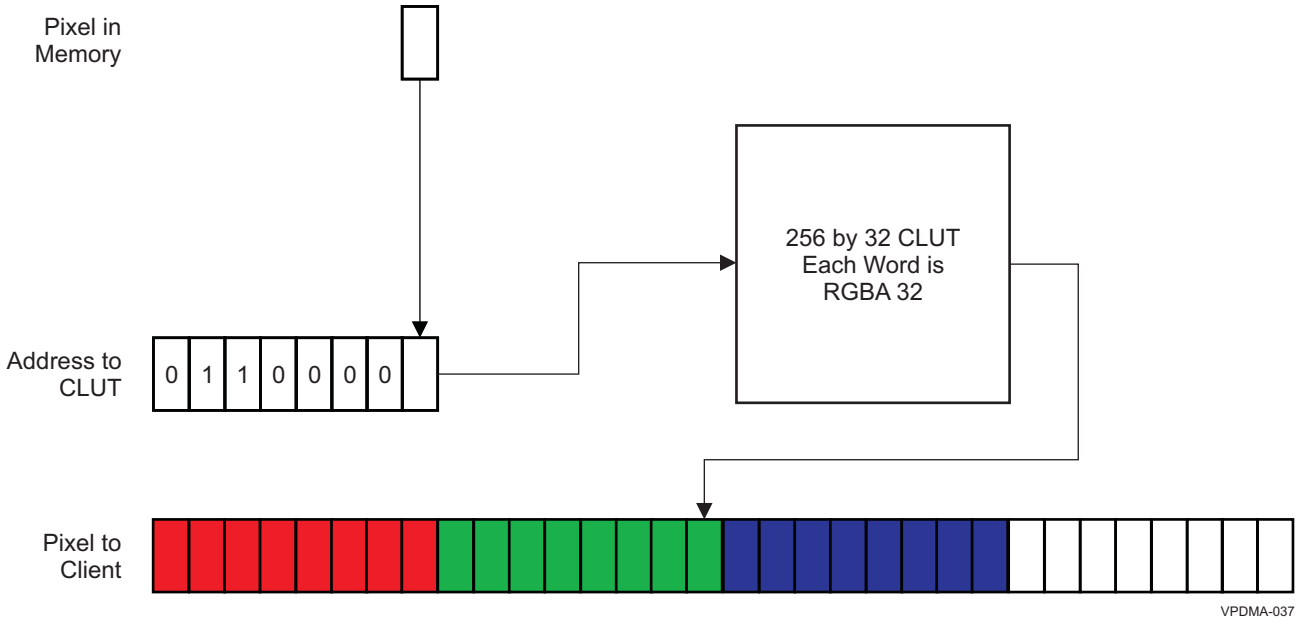


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.21 Bitmap-1 Offset3 (Data Type 2Bh)

In Bitmap-1 Offset3 data, each pixel is an 1 bit value that is used as the lower bit of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 3 bits in this case are tied to 3'b000. Bits 1-4 are tied to 0.

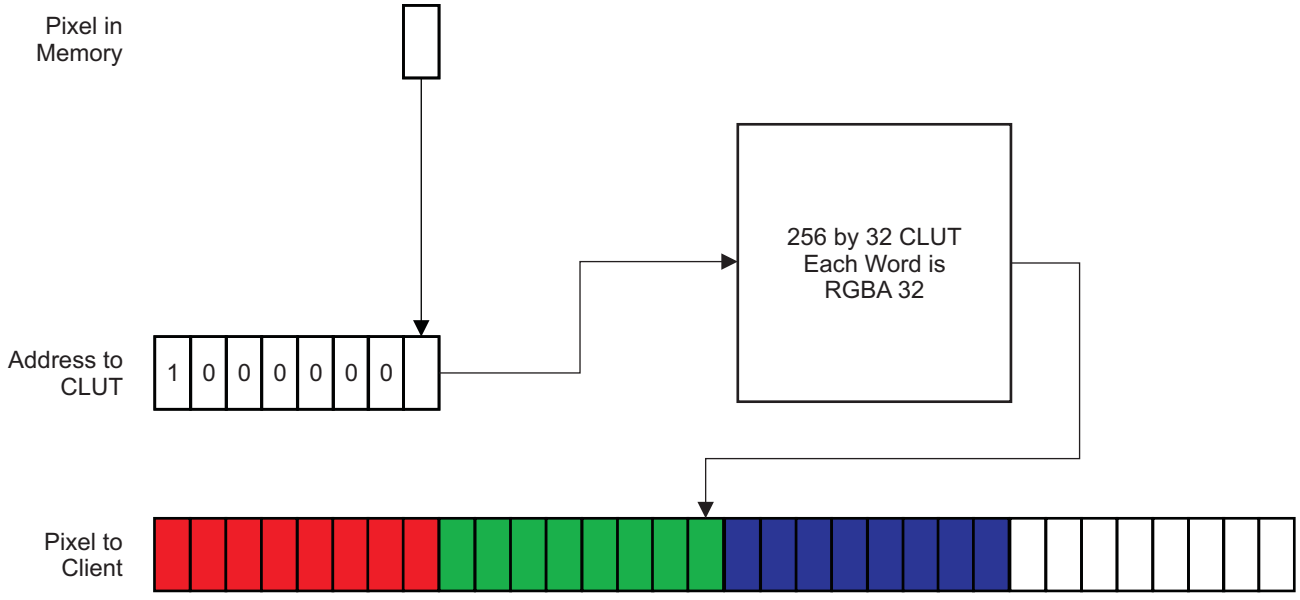
**Figure 12-206. Bitmap-1 Offset3 (Data Type 2Bh)**




#### 12.2.13.6.2.22 Bitmap-1 Offset4 (Data Type 2Ch)

In Bitmap-1 Offset4 data, each pixel is an 1 bit value that is used as the lower bit of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 3 bits in this case are tied to 3'b000. Bits 1-4 are tied to 0.

**Figure 12-207. Bitmap-1 Offset4 (Data Type 2Ch)**



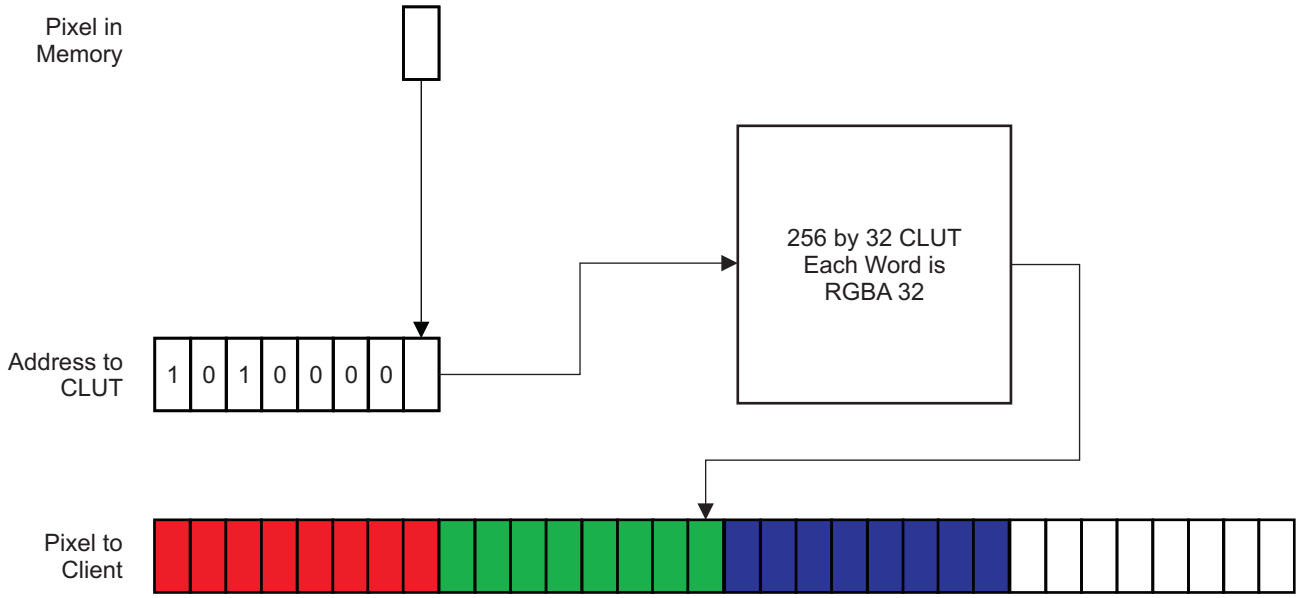


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.23 Bitmap-1 Offset5 (Data Type 2Dh)

In Bitmap-1 Offset5 data, each pixel is an 1 bit value that is used as the lower bit of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 3 bits in this case are tied to 3'b101. Bits 1-4 are tied to 0.

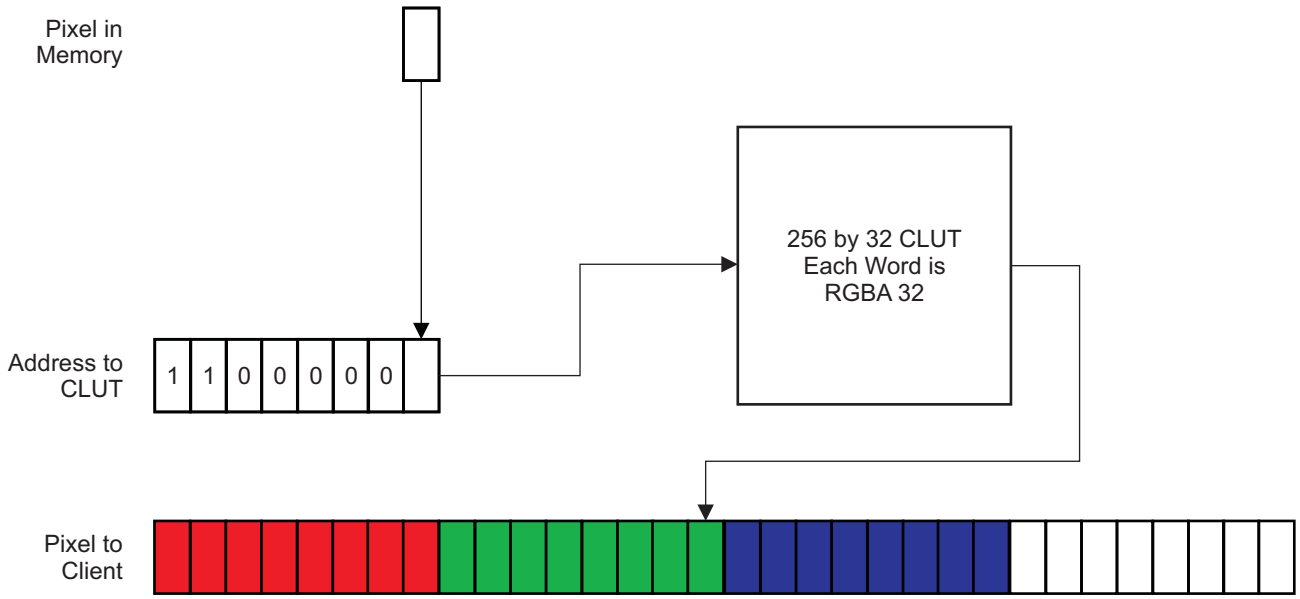
**Figure 12-208. Bitmap-1 Offset5 (Data Type 2Dh)**




#### 12.2.13.6.2.24 Bitmap-1 Offset6 (Data Type 2Eh)

In Bitmap-1 Offset6 data, each pixel is an 1 bit value that is used as the lower bit of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 3 bits in this case are tied to 3'b110. Bits 1-4 are tied to 0.

**Figure 12-209. Bitmap-1 Offset6 (Data Type 2Eh)**



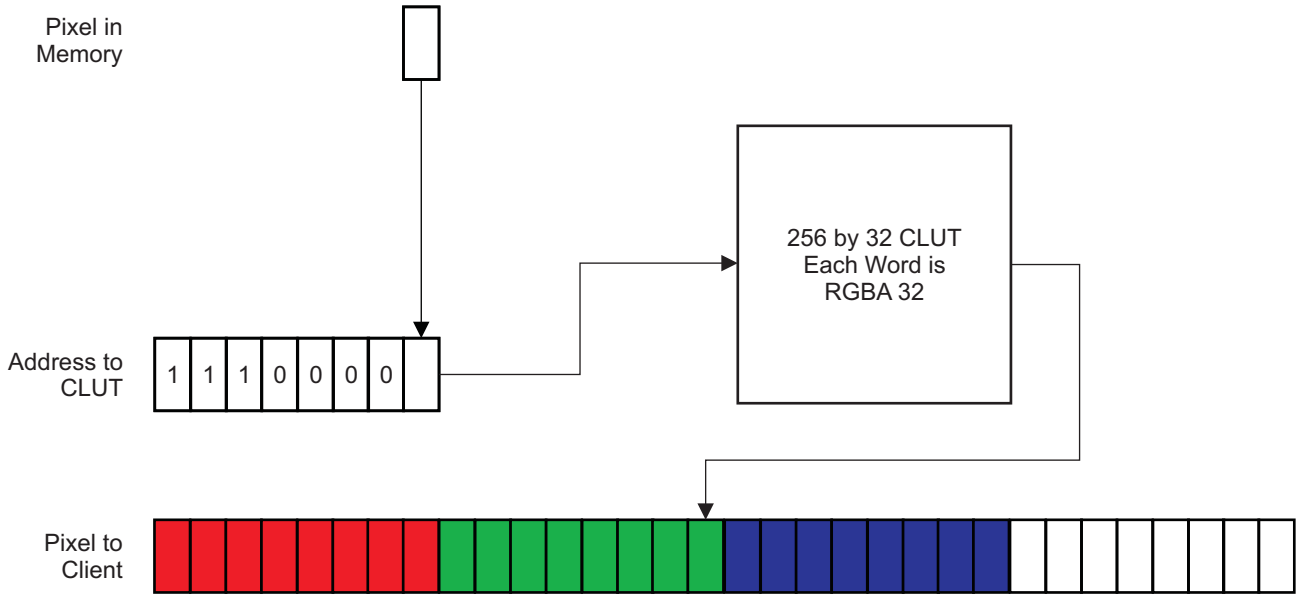


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.2.13.6.2.25 Bitmap-1 Offset7 (Data Type 2Fh)

In Bitmap-1 Offset7 data, each pixel is an 1 bit value that is used as the lower bit of address into a 256 × 32 memory that gives the final RGBA32-8888 data value. The upper 3 bits in this case are tied to 3'b111. Bits 1-4 are tied to 0.

**Figure 12-210. Bitmap-1 Offset7 (Data Type 2Fh)**




### 12.2.13.6.3 Miscellaneous Data Type

The Miscellaneous channel type is for any data type that is not a normal video type. The Miscellaneous channel type makes no assumptions on data type and just passes the data to the client and supports a single buffer for the client. The size of the miscellaneous data type is always determined by the Data Type field of the descriptor which specifies the number of bits in the descriptor.

### 12.2.13.6.4 Free Channel Data Type

The Free channel data type is always ignored as it uses the same data type of the descriptor that first calls the free channel.


Copyright © 2011–2016, Texas Instruments Incorporated

## 12.3 Registers

A register name is comprised of <MODULE_NAME>_<mmr_name> with the MODULE_NAME in upper-case and the mmr_name in that module in lower-case. For example, register name CSC_csc02: where CSC represents the module name (for which registers are specified with offsets in this section) and csc02 represents the MMR name in the module MMR list.

Each module may have multiple instances of it in the subsystem (see Table 12-114). For example, the CSC module has 6 instances with different names: CSC_HD0, CSC_HD1, CSC_SD, CSC_WB2, CSC_VIP0, and CSC_VIP1. See Figure 12-1 for the module instance names and its location in the subsystem. All the instances of a module share the same set of registers mentioned in this section.

Table 12-115 lists the memory map for the HDVPSS registers.

To calculate actual/final address of a specific MMR or register, use this example: register CSC_csc02, for the instance CSC_HD1.

| | | |
|---|---|---|
| Determine the address of CSC_csc02: | 0x4810 0000 | (Base Address of HDVPSS) |
| | + 0x0000 0C00 | (Instance Offset Address: CSC_HD1 from Table 12-115) |
| | + 0x0000 0008 | (Register Offset Address: CSC_csc02 from , corresponding instance register in this section or search for the register name in this section) |
| | = 0x4810 0C08 | (Actual address of CSC_csc02) |

**Table 12-114. HDVPSS Module Instances**

| Module Name | Number of Instances | Instance Names |
|---|---|---|
| CHR_US | 6 | CHR_US_P0, CHR_US_P1, CHR_US_P2, CSC_US_AUX, CHR_US_SEC0, CHR_US_SEC1 |
| CSC | 6 | CSC_HD0, CSC_HD1, CSC_SD, CSC_WB2, CSC_VIP0, CSC_VIP1 |
| COMP | 1 | COMP |
| CIG | 1 | CIG |
| GRPX[1] | 3 | GRPX0, GRPX1, GRPX2 |
| SD_VENC | 1 | SD_VENC |
| HD_VENC | 3 | HDMI/DVO1, DVO2 |
| VIP_PARSER | 2 | VIP0_PARSER, VIP1_PARSER |
| DEI | 1 | DEI |
| NF | 1 | NF |
| SC | 5 | SC_1, SC_2, SC_3, SC_4, SC_5 |
| VCOMP | 1 | VCOMP |
| VPDMA[2] | 1 | VPDMA |
| CHR_DS[3] | 5 | CHR_DS0_VIP0, CHR_DS1_VIP0, CHR_DS0_VIP1, CHR_DS1_VIP1, CHR_DS_NF |

(1) GRPX module does not contain any 32-bit MMRs. The GRPX module is controlled by configuring the attributes mentioned in Section 12.2.5.

(2) CPU and VPDMA (through configuration descriptor) can access all the registers mentioned in this section. The VPDMA can also access GRPX attribute configuration and coefficients of all the scaler instances; however, the CPU can not.

(3) CHR_DS module does not contain any MMRs. The CHR_DS module is enabled by selecting 420 data formatted output in the clkc_vip registers.


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** The block diagram and text sections of this chapter consistently use the instance numbering for the VIP and GRPX submodules as VIP0/VIP1 and GRPX0/GRPX1/GRPX2. However, the register sections use the numbering VIP1/VIP2 and GRPX1/GRPX2/GRPX3. The following list is the convention for submodule naming to register control:

- Submodule VIP0 is controlled by registers VIP1
- Submodule VIP1 is controlled by registers VIP2
- Submodule GRPX0 is controlled by registers GRPX1
- Submodule GRPX1 is controlled by registers GRPX2
- Submodule GRPX2 is controlled by registers GRPX3

**Table 12-115. HDVPSS Registers**

| HDVPSS Submodule | Submodule Base Address |
|---|---|
| INTC | 0000h |
| CLKC | 0100h |
| CHR_US_P0 | 0300h |
| CHR_US_P1 | 0400h |
| CHR_US_P2 | 0500h |
| DEI | 0600h |
| SC_1 | 0700h |
| CHR_US_AUX | 0A00h |
| SC_2 | 0B00h |
| CSC_HD1 | 0C00h |
| CSC_SD | 0D00h |
| VCOMP | 0E00h |
| CSC_HD0 | 0F00h |
| SC_5 | 5000h |
| CIG | 5100h |
| COMP | 5200h |
| CSC_WB2 | 5300h |
| CHR_US_SEC0 | 5400h |
| CHR_US_SEC1 | 5480h |
| VIP_PARSER0 | 5500h |
| CSC_VIP0 | 5700h |
| SC_3 | 5800h |
| VIP_PARSER1 | 5A00h |
| CSC_VIP1 | 5C00h |
| SC_4 | 5D00h |
| SD_VENC | 5E00h |
| HD_VENC_D_DVO1 | 6000h |
| HD_VENC_D_DVO2 | A000h |
| NF | C200h |
| VPDMA | D000h |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.1 CHR_US Registers

Table 12-116 lists the memory-mapped registers for the CHR_US. All register offset addresses not listed in Table 12-116 should be considered as reserved locations and the register contents should not be modified.

**Table 12-116. CHR_US REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 4h | CHR_US_reg0 | Upsampling Coeffs | Section 12.3.1.1 |
| 8h | CHR_US_reg1 | Upsampling Coeffs | Section 12.3.1.2 |
| Ch | CHR_US_reg2 | Upsampling Coeffs | Section 12.3.1.3 |
| 10h | CHR_US_reg3 | Upsampling Coeffs | Section 12.3.1.4 |
| 14h | CHR_US_reg4 | Upsampling Coeffs | Section 12.3.1.5 |
| 18h | CHR_US_reg5 | Upsampling Coeffs | Section 12.3.1.6 |
| 1Ch | CHR_US_reg6 | Upsampling Coeffs | Section 12.3.1.7 |
| 20h | CHR_US_reg7 | Upsampling Coeffs | Section 12.3.1.8 |

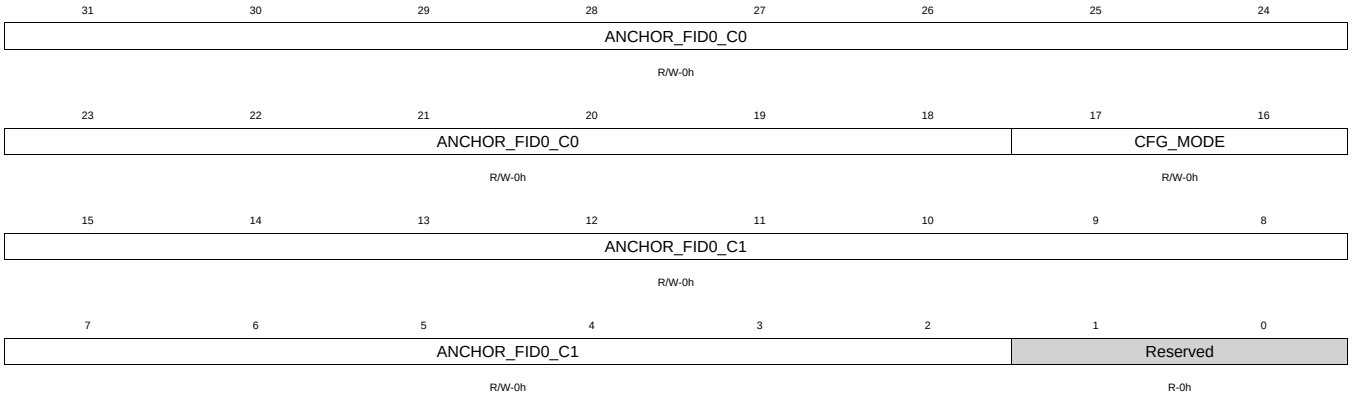

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.1.1 CHR_US_reg0 Register (offset = 4h) [reset = 0h]

CHR_US_reg0 is shown in Figure 12-211 and described in Table 12-117.

Upsampling Coeffs

**Figure 12-211. CHR_US_reg0 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-117. CHR_US_reg0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | ANCHOR_FID0_C0 | R/W | 0h | C0 coefficient for Anchor Pixel. Used when field_id = 0 |
| 17-16 | CFG_MODE | R/W | 0h | 00 = Mode A<br>01 = Mode B |
| 15-2 | ANCHOR_FID0_C1 | R/W | 0h | C1 coefficient for Anchor Pixel. Used when field_id = 0 |
| 1-0 | Reserved | R | 0h | Reserved |



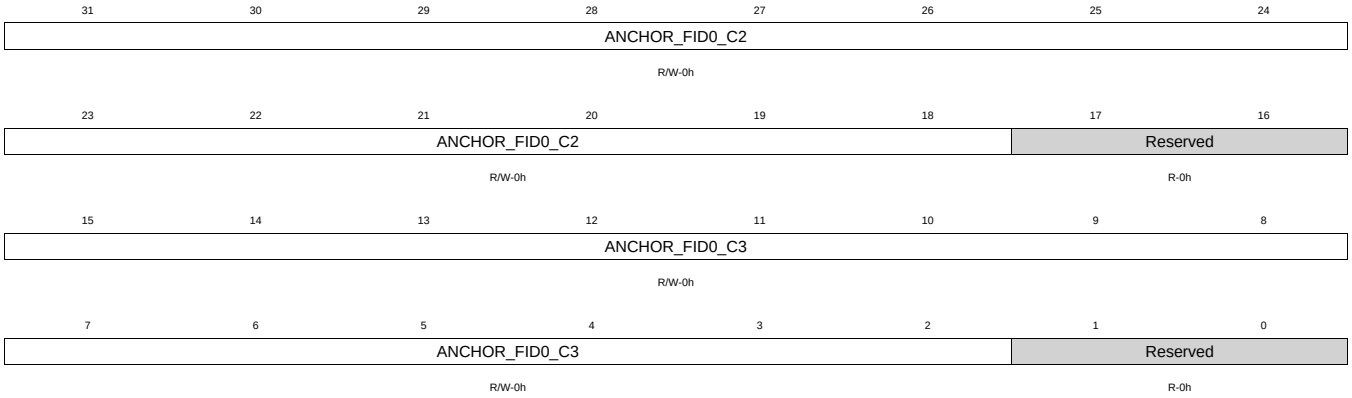
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.1.2 CHR_US_reg1 Register (offset = 8h) [reset = 0h]

CHR_US_reg1 is shown in Figure 12-212 and described in Table 12-118.

Upsampling Coeffs

**Figure 12-212. CHR_US_reg1 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-118. CHR_US_reg1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | ANCHOR_FID0_C2 | R/W | 0h | C2 coefficient for Anchor Pixel. Used when field_id = 0 |
| 17-16 | Reserved | R | 0h | Reserved |
| 15-2 | ANCHOR_FID0_C3 | R/W | 0h | C3 coefficient for Anchor Pixel. Used when field_id = 0 |
| 1-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.1.3 CHR_US_reg2 Register (offset = Ch) [reset = 0h]

CHR_US_reg2 is shown in Figure 12-213 and described in Table 12-119.

Upsampling Coeffs

**Figure 12-213. CHR_US_reg2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INTERP_FID0_C0 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INTERP_FID0_C0 | | | | | | Reserved | |
| R/W-0h | | | | | | R-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INTERP_FID0_C1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTERP_FID0_C1 | | | | | | Reserved | |
| R/W-0h | | | | | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-119. CHR_US_reg2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | INTERP_FID0_C0 | R/W | 0h | C0 coefficient for Interpolated Pixel. Used when field_id = 0 |
| 17-16 | Reserved | R | 0h | Reserved |
| 15-2 | INTERP_FID0_C1 | R/W | 0h | C1 coefficient for Interpolated Pixel. Used when field_id = 0 |
| 1-0 | Reserved | R | 0h | Reserved |



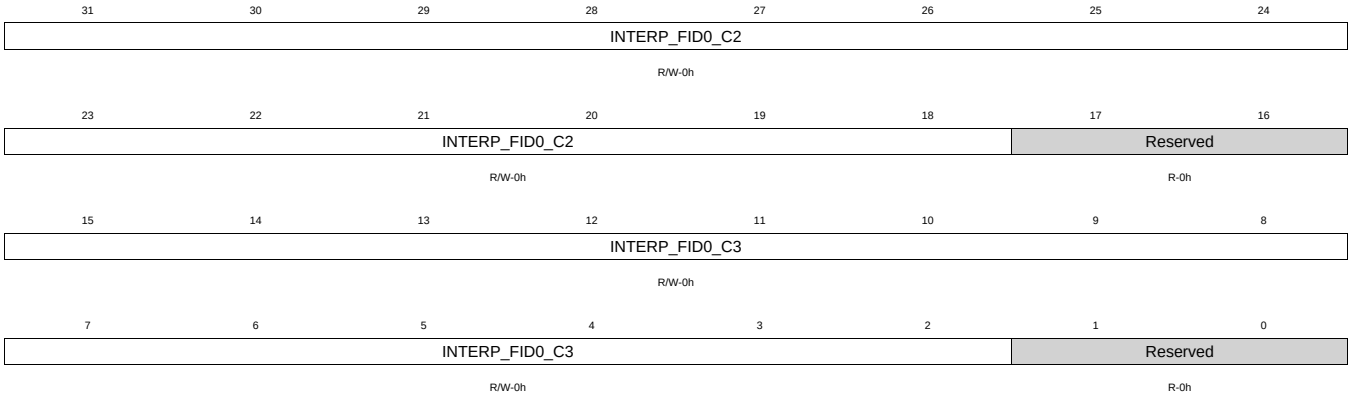
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.1.4 CHR_US_reg3 Register (offset = 10h) [reset = 0h]

CHR_US_reg3 is shown in Figure 12-214 and described in Table 12-120.

Upsampling Coeffs

**Figure 12-214. CHR_US_reg3 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-120. CHR_US_reg3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | INTERP_FID0_C2 | R/W | 0h | C2 coefficient for Interpolated Pixel. Used when field_id = 0 |
| 17-16 | Reserved | R | 0h | Reserved |
| 15-2 | INTERP_FID0_C3 | R/W | 0h | C3 coefficient for Interpolated Pixel. Used when field_id = 0 |
| 1-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.1.5 CHR_US_reg4 Register (offset = 14h) [reset = 0h]

CHR_US_reg4 is shown in Figure 12-215 and described in Table 12-121.

Upsampling Coeffs

**Figure 12-215. CHR_US_reg4 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ANCHOR_FID1_C0 | | | | | | | |
| R/W-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ANCHOR_FID1_C0 | | | | | | Reserved | |
| R/W-0h | | | | | | R-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| ANCHOR_FID1_C1 | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ANCHOR_FID1_C1 | | | | | | Reserved | |
| R/W-0h | | | | | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-121. CHR_US_reg4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | ANCHOR_FID1_C0 | R/W | 0h | C0 coefficient for Anchor Pixel. Used when field_id = 1 |
| 17-16 | Reserved | R | 0h | Reserved |
| 15-2 | ANCHOR_FID1_C1 | R/W | 0h | C1 coefficient for Anchor Pixel. Used when field_id = 1 |
| 1-0 | Reserved | R | 0h | Reserved |



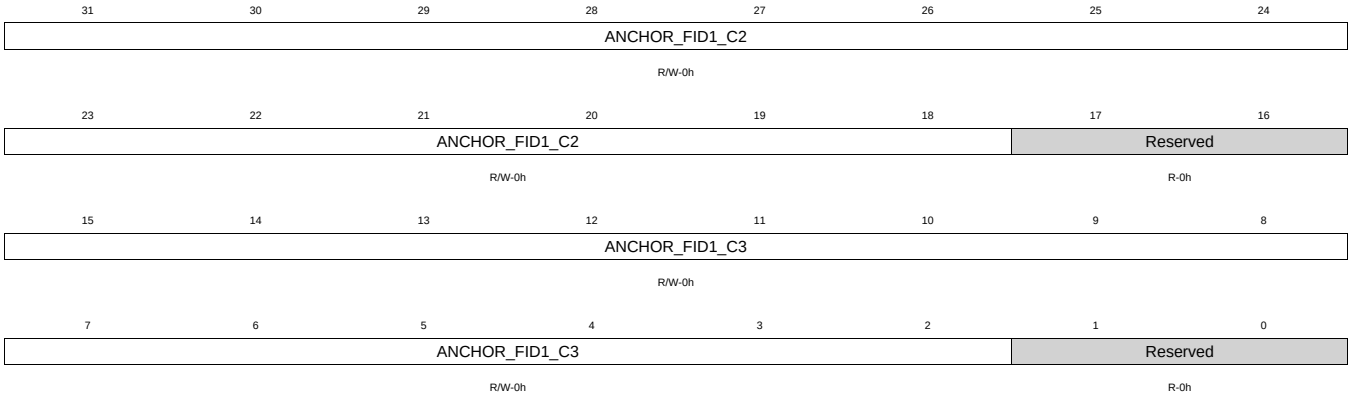
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.1.6 CHR_US_reg5 Register (offset = 18h) [reset = 0h]

CHR_US_reg5 is shown in Figure 12-216 and described in Table 12-122.

Upsampling Coeffs

**Figure 12-216. CHR_US_reg5 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-122. CHR_US_reg5 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | ANCHOR_FID1_C2 | R/W | 0h | C2 coefficient for Anchor Pixel. Used when field_id = 1 |
| 17-16 | Reserved | R | 0h | Reserved |
| 15-2 | ANCHOR_FID1_C3 | R/W | 0h | C3 coefficient for Anchor Pixel. Used when field_id = 1 |
| 1-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.1.7 CHR_US_reg6 Register (offset = 1Ch) [reset = 0h]

CHR_US_reg6 is shown in Figure 12-217 and described in Table 12-123.

Upsampling Coeffs

**Figure 12-217. CHR_US_reg6 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INTERP_FID1_C0 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INTERP_FID1_C0 | | | | | | Reserved | |
| R/W-0h | | | | | | R-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INTERP_FID1_C1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTERP_FID1_C1 | | | | | | Reserved | |
| R/W-0h | | | | | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-123. CHR_US_reg6 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | INTERP_FID1_C0 | R/W | 0h | C0 coefficient for Interpolated Pixel. Used when field_id = 1 |
| 17-16 | Reserved | R | 0h | Reserved |
| 15-2 | INTERP_FID1_C1 | R/W | 0h | C1 coefficient for Interpolated Pixel. Used when field_id = 1 |
| 1-0 | Reserved | R | 0h | Reserved |



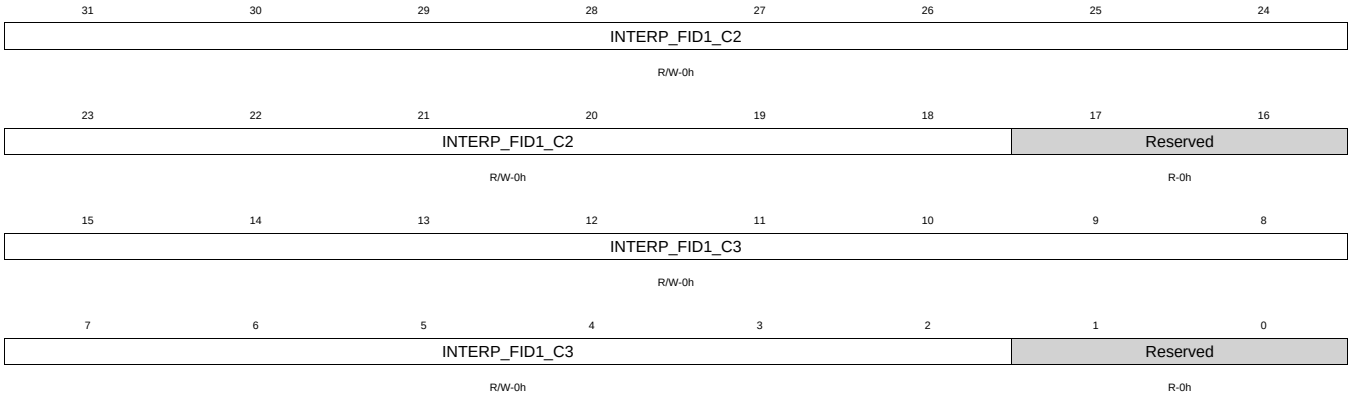
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.1.8 CHR_US_reg7 Register (offset = 20h) [reset = 0h]

CHR_US_reg7 is shown in Figure 12-218 and described in Table 12-124.

Upsampling Coeffs

**Figure 12-218. CHR_US_reg7 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-124. CHR_US_reg7 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | INTERP_FID1_C2 | R/W | 0h | C2 coefficient for Interpolated Pixel. Used when field_id = 1 |
| 17-16 | Reserved | R | 0h | Reserved |
| 15-2 | INTERP_FID1_C3 | R/W | 0h | C3 coefficient for Interpolated Pixel. Used when field_id = 1 |
| 1-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2 CIG Registers

Table 12-125 lists the memory-mapped registers for the CIG. All register offset addresses not listed in Table 12-125 should be considered as reserved locations and the register contents should not be modified.

**Table 12-125. CIG REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | CIG_reg0 | CIG Mode Reg | Section 12.3.2.1 |
| 4h | CIG_reg1 | CIG Display Config Reg | Section 12.3.2.2 |
| 8h | CIG_reg2 | CIG HDMI Transparency Config Reg | Section 12.3.2.3 |
| Ch | CIG_reg3 | CIG HDMI Transparent Color | Section 12.3.2.4 |
| 10h | CIG_reg4 | CIG HDCOMP Transparency Config Reg | Section 12.3.2.5 |
| 14h | CIG_reg5 | CIG HDCOMP Transparent Color | Section 12.3.2.6 |
| 18h | CIG_reg6 | CIG PIP Display Config Reg | Section 12.3.2.7 |
| 1Ch | CIG_reg7 | CIG PIP Position Config Reg | Section 12.3.2.8 |
| 20h | CIG_reg8 | CIG PIP Size Config Reg | Section 12.3.2.9 |
| 24h | CIG_reg9 | CIG PIP Transparency Config Reg | Section 12.3.2.10 |
| 28h | CIG_reg10 | CIG PIP Transparent Color | Section 12.3.2.11 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.1 CIG_reg0 Register (offset = 0h) [reset = 0h]

CIG_reg0 is shown in and described in Table 12-126.

CIG Mode Reg

**Table 12-126. CIG_reg0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31- | Reserved | R | 0h | Reserved |
| 7 | PIP_P2I_EN | R/W | 0h | Enable Output Interlacing for PIP output<br>0: Disabled<br>1: Enabled |
| 6 | PIP_FULLSIZE | R/W | 0h | CIG PIP window size<br>0: Sub-window<br>1: full-size |
| 5 | PIP_EN | R/W | 0h | CIG PIP path enable<br>0: Disabled<br>1: Enabled |
| 4 | CI_FIELD_RPT_EN | R/W | 0h | CIG filter field drop/repeat option enable for vertical filtering<br>0 : disable (use the Vertical filter)<br>1: enable (use field repeat) |
| 3 | CI_VDEC_EN | R/W | 0h | CIG filter vertical decimation enable<br>0: Disabled (horizontal decimation only)<br>1: Enabled (both vertical and horizontal enabled) |
| 2 | CI_P2I_EN | R/W | 0h | Enable Output Interlacing for Constrained output<br>0: Disabled<br>1: Enabled |
| 1 | P2I_EN | R/W | 0h | Enable Output Interlacing for Non-constrained output<br>0: Disabled<br>1: Enabled |
| 0 | CIG_EN | R/W | 0h | CIG module enable<br>0: Disabled (Bypass Mode)<br>1: CIG filter Enabled |

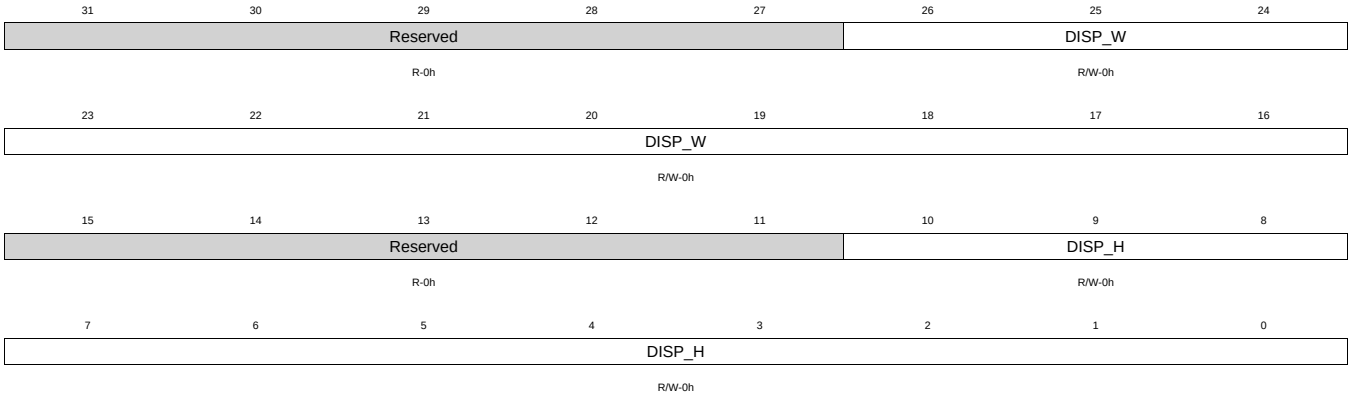

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.2 CIG_reg1 Register (offset = 4h) [reset = 0h]

CIG_reg1 is shown in Figure 12-219 and described in Table 12-127.

CIG Display Config Reg

**Figure 12-219. CIG_reg1 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-127. CIG_reg1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | Reserved |
| 26-16 | DISP_W | R/W | 0h | Output display Width (max = 1920) |
| 15-11 | Reserved | R | 0h | Reserved |
| 10-0 | DISP_H | R/W | 0h | Output display Height (max = 0x7FF) |

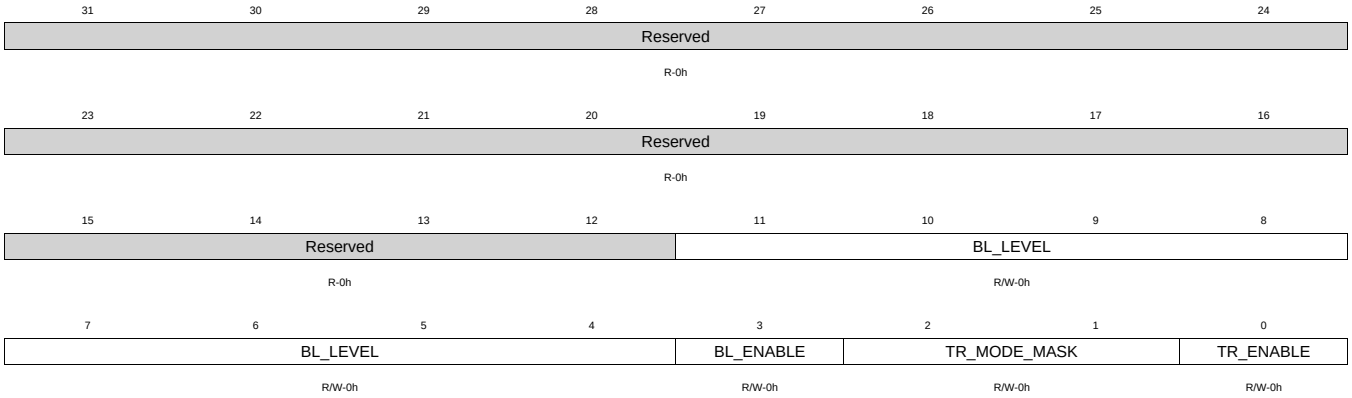

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.3 CIG_reg2 Register (offset = 8h) [reset = 0h]

CIG_reg2 is shown in Figure 12-220 and described in Table 12-128.

CIG HDMI Transparency Config Reg

**Figure 12-220. CIG_reg2 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-128. CIG_reg2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | Reserved | R | 0h | Reserved |
| 11-4 | BL_LEVEL | R/W | 0h | Blending Value Assigned to the video pixel's alpha if blending is enabled. |
| 3 | BL_ENABLE | R/W | 0h | Blending Enable<br>0: Disable Blending<br>1: Apply Blending |
| 2-1 | TR_MODE_MASK | R/W | 0h | Transparency Color Mask Bit (Number of LSB bits to mask when checking for pixel transparency)<br>0: No masking<br>1: Mask 1 LSB bit<br>2: Mask 2 LSB bits<br>3: Mask 3 LSB bits |
| 0 | TR_ENABLE | R/W | 0h | Transparency Enable<br>0: Disable Transparency<br>1: Enable Transparency |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.4 CIG_reg3 Register (offset = Ch) [reset = 0h]

CIG_reg3 is shown in Figure 12-221 and described in Table 12-129.

CIG HDMI Transparent Color

**Figure 12-221. CIG_reg3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TR_COLOR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TR_COLOR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TR_COLOR | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-129. CIG_reg3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-0 | TR_COLOR | R/W | 0h | Transparency Color 23:16 R 15:8 G 7:0 B (If the video pixel matches this color while transparency is enabled, the alpha value is forced to 0.) |

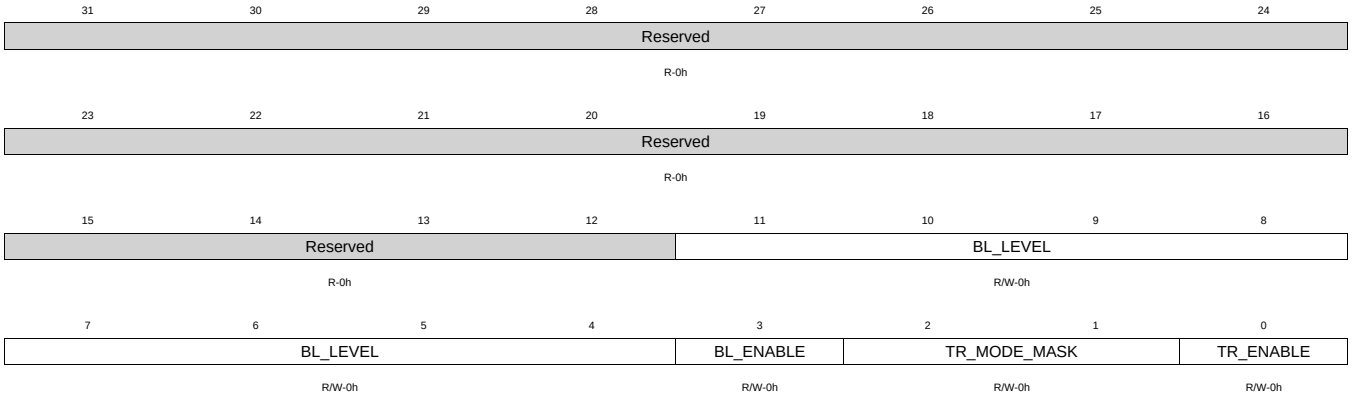

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.5 CIG_reg4 Register (offset = 10h) [reset = 0h]

CIG_reg4 is shown in Figure 12-222 and described in Table 12-130.

CIG HDCOMP Transparency Config Reg

**Figure 12-222. CIG_reg4 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-130. CIG_reg4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | Reserved | R | 0h | Reserved |
| 11-4 | BL_LEVEL | R/W | 0h | Blending Value Assigned to the video pixel's alpha if blending is enabled. |
| 3 | BL_ENABLE | R/W | 0h | Blending Enable<br>0: Disable Blending<br>1: Apply Blending |
| 2-1 | TR_MODE_MASK | R/W | 0h | Transparency Color Mask Bit (Number of LSB bits to mask when checking for pixel transparency)<br>0: No masking<br>1: Mask 1 LSB bit<br>2: Mask 2 LSB bits<br>3: Mask 3 LSB bits |
| 0 | TR_ENABLE | R/W | 0h | Transparency Enable<br>0: Disable Transparency<br>1: Enable Transparency |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.6 CIG_reg5 Register (offset = 14h) [reset = 0h]

CIG_reg5 is shown in Figure 12-223 and described in Table 12-131.

CIG HDCOMP Transparent Color

**Figure 12-223. CIG_reg5 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| TR_COLOR | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| TR_COLOR | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TR_COLOR | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-131. CIG_reg5 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-0 | TR_COLOR | R/W | 0h | Transparency Color 23:16 R 15:8 G 7:0 B (If the video pixel matches this color while transparency is enabled, the alpha value is forced to 0.) |

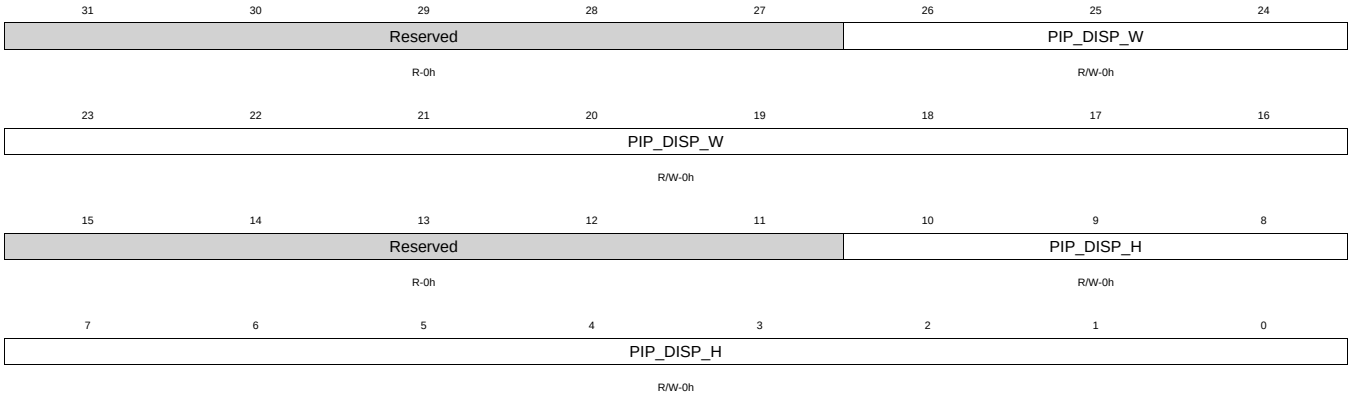

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.7 CIG_reg6 Register (offset = 18h) [reset = 0h]

CIG_reg6 is shown in Figure 12-224 and described in Table 12-132.

CIG PIP Display Config Reg

**Figure 12-224. CIG_reg6 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-132. CIG_reg6 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | Reserved |
| 26-16 | PIP_DISP_W | R/W | 0h | PIP Output display Width (max = 1920) |
| 15-11 | Reserved | R | 0h | Reserved |
| 10-0 | PIP_DISP_H | R/W | 0h | PIP Output display Height (max = 0x7FF) |

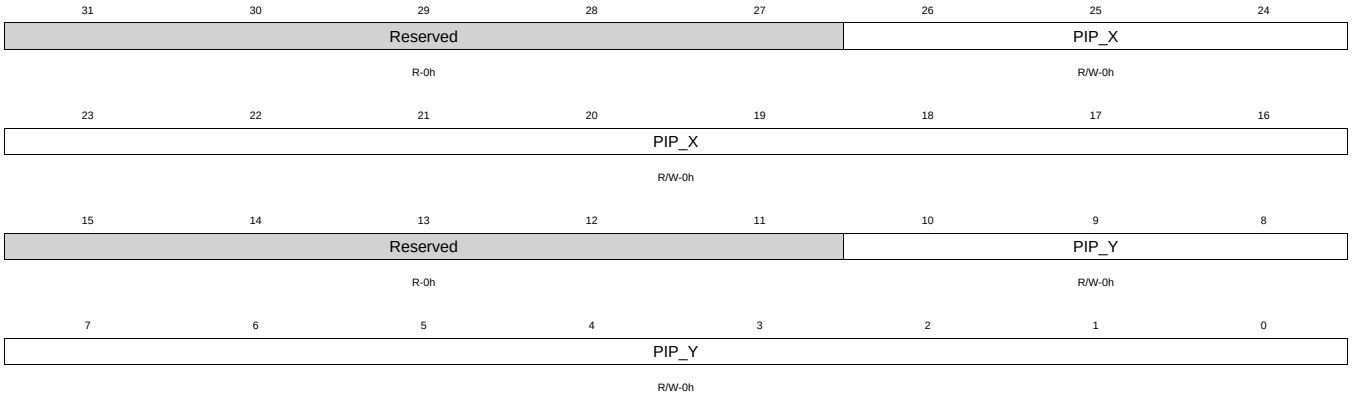

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.8 CIG_reg7 Register (offset = 1Ch) [reset = 0h]

CIG_reg7 is shown in Figure 12-225 and described in Table 12-133.

CIG PIP Position Config Reg

**Figure 12-225. CIG_reg7 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-133. CIG_reg7 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | Reserved |
| 26-16 | PIP_X | R/W | 0h | PIP window X position |
| 15-11 | Reserved | R | 0h | Reserved |
| 10-0 | PIP_Y | R/W | 0h | PIP window Y position |

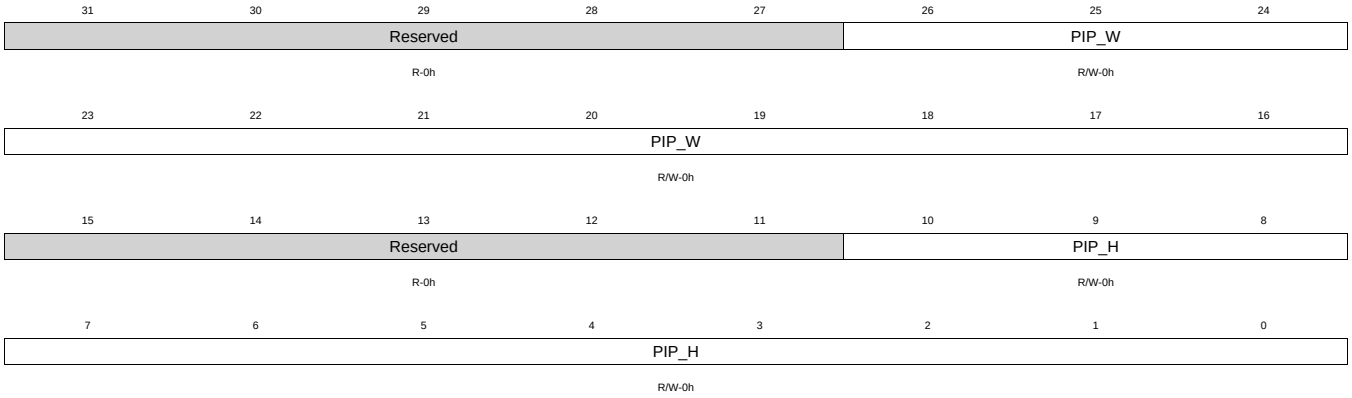

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.9 CIG_reg8 Register (offset = 20h) [reset = 0h]

CIG_reg8 is shown in Figure 12-226 and described in Table 12-134.

CIG PIP Size Config Reg

**Figure 12-226. CIG_reg8 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-134. CIG_reg8 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | Reserved |
| 26-16 | PIP_W | R/W | 0h | PIP window Width ( pip_x + pip_w must be less than or equal to disp_w) |
| 15-11 | Reserved | R | 0h | Reserved |
| 10-0 | PIP_H | R/W | 0h | PIP window Height ( pip_y + pip_h must be less than or equal to disp_h) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.10 CIG_reg9 Register (offset = 24h) [reset = 0h]

CIG_reg9 is shown in Figure 12-227 and described in Table 12-135.

CIG PIP Transparency Config Reg

**Figure 12-227. CIG_reg9 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | BL_LEVEL | | | |
| R-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BL_LEVEL | | | | BL_ENABLE | TR_MODE_MASK | | TR_ENABLE |
| R/W-0h | | | | R/W-0h | R/W-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-135. CIG_reg9 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | Reserved | R | 0h | Reserved |
| 11-4 | BL_LEVEL | R/W | 0h | Blending Value Assigned to the video pixel's alpha if blending is enabled. |
| 3 | BL_ENABLE | R/W | 0h | Blending Enable<br>0: Disable Blending<br>1: Apply Blending |
| 2-1 | TR_MODE_MASK | R/W | 0h | Transparency Color Mask Bit (Number of LSB bits to mask when checking for pixel transparency)<br>0: No masking<br>1: Mask 1 LSB bit<br>2: Mask 2 LSB bits<br>3: Mask 3 LSB bits |
| 0 | TR_ENABLE | R/W | 0h | Transparency Enable<br>0: Disable Transparency<br>1: Enable Transparency |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.2.11 CIG_reg10 Register (offset = 28h) [reset = 0h]

CIG_reg10 is shown in Figure 12-228 and described in Table 12-136.

CIG PIP Transparent Color

**Figure 12-228. CIG_reg10 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TR_COLOR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TR_COLOR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TR_COLOR | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-136. CIG_reg10 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-0 | TR_COLOR | R/W | 0h | Transparency Color 23:16 R 15:8 G 7:0 B (If the video pixel matches this color while transparency is enabled, the alpha value is forced to 0.) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.3 COMP Registers

Table 12-137 lists the memory-mapped registers for the COMP. All register offset addresses not listed in Table 12-137 should be considered as reserved locations and the register contents should not be modified.

**Table 12-137. COMP REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | COMP_status | Compositor Status | Section 12.3.3.1 |
| 4h | COMP_hdmi_settings | COMP HDMI/DVO1 Settings | Section 12.3.3.2 |
| 8h | COMP_dvo2_settings | COMP DVO2 Settings | Section 12.3.3.3 |
| 10h | COMP_sd_settings | COMP SD Settings | Section 12.3.3.4 |
| 14h | COMP_back_color_settings | COMP Background Color Settings | Section 12.3.3.5 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 12.3.3.1 COMP_status Register (offset = 0h) [reset = 0h]

COMP_status is shown in and described in .

Compositor Status


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.3.2 COMP_hdmi_settings Register (offset = 4h) [reset = E400h]

COMP_hdmi_settings is shown in Figure 12-229 and described in Table 12-138.

COMP HDMI/DVO1 Settings

**Figure 12-229. COMP_hdmi_settings Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | FB_SEL |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| G2_ORDER | | G1_ORDER | | G0_ORDER | | VID_ORDER | |
| R/W-3h | | R/W-2h | | R/W-1h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| G_ORDER | VID_BLD_ORDER | VID_EN | Reserved | PIP_EN | GRPX0_EN | GRPX1_EN | GRPX2_EN |
| R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-138. COMP_hdmi_settings Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | Reserved | R | 0h | Reserved |
| 16 | FB_SEL | R/W | 0h | Feedback data selection<br>0: select data from video alpha blending<br>1: select data from final alpha blending |
| 15-14 | G2_ORDER | R/W | 3h | Graphic2 layer display order when g_order = 1. From low to high: 00, 01, 10, and 11. |
| 13-12 | G1_ORDER | R/W | 2h | Graphic1 layer display order when g_order = 1. From low to high: 00, 01, 10, and 11. |
| 11-10 | G0_ORDER | R/W | 1h | Graphic0 layer display order when g_order = 1. From low to high: 00, 01, 10, and 11. |
| 9-8 | VID_ORDER | R/W | 0h | Video layer display order. From low to high: 00, 01, 10, and 11. |
| 7 | G_ORDER | R/W | 0h | Global reorder. 1: global reorder is on. The graphic layer priority is based on g0_reorder, g1_reorder, g2_order |
| 6 | VID_BLD_ORDER | R/W | 0h | Video Build Order<br>0: Use HD_VID as base (lower) layer for video stream<br>1: Use HD_PIP as base (lower) layer for video stream |
| 5 | VID_EN | R/W | 0h | HD VID channel Enable 0 : Disabled 1 : Enabled |
| 4 | Reserved | R | 0h | Reserved |
| 3 | PIP_EN | R/W | 0h | HD PIP channel enable 0 : Disabled 1 : Enabled |
| 2 | GRPX0_EN | R/W | 0h | GRPX2 channel enable 0 : Disabled 1 : Enabled |
| 1 | GRPX1_EN | R/W | 0h | GRPX1 channel enable 0 : Disabled 1 : Enabled |
| 0 | GRPX2_EN | R/W | 0h | GRPX2 channel enable 0 : Disabled 1 : Enabled |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.3.3 COMP_dvo2_settings Register (offset = 8h) [reset = E400h]

COMP_dvo2_settings is shown in Figure 12-230 and described in Table 12-139.

**Figure 12-230. COMP_dvo2_settings Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | FB_SEL |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| G2_ORDER | | G1_ORDER | | G0_ORDER | | VID_ORDER | |
| R/W-3h | | R/W-2h | | R/W-1h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| G_ORDER | VID_BLD_ORDER | VID_EN | Reserved | PIP_EN | GRPX0_EN | GRPX1_EN | GRPX2_EN |
| R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-139. COMP_dvo2_settings Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | Reserved | R | 0h | Reserved |
| 16 | FB_SEL | R/W | 0h | Feedback data selection: 0: select data from video alpha blending 1: select data from final alpha blending |
| 15-14 | G2_ORDER | R/W | 3h | Graphic2 layer display order when g_order = 1. From low to high: 00, 01, 10, and 11. |
| 13-12 | G1_ORDER | R/W | 2h | Graphic1 layer display order when g_order = 1. From low to high: 00, 01, 10, and 11. |
| 11-10 | G0_ORDER | R/W | 1h | Graphic0 layer display order when g_order = 1. From low to high: 00, 01, 10, and 11. |
| 9-8 | VID_ORDER | R/W | 0h | Video layer display order. From low to high: 00, 01, 10, and 11. |
| 7 | G_ORDER | R/W | 0h | Global reorder. 1: global reorder is on. The graphic layer priority is based on g0_reorder, g1_reorder, g2_order settings 0: no global reorder. The graphic layer priority is based priority bits [35:32] in the data bus. |
| 6 | VID_BLD_ORDER | R/W | 0h | Video Build Order 1: Use HD_PIP as base (lower) layer for video stream 0: Use HD_vid as base (lower) layer for video stream |
| 5 | VID_EN | R/W | 0h | HD VID channel Enable 0 : Disabled 1 : Enabled |
| 4 | Reserved | R | 0h | Reserved |
| 3 | PIP_EN | R/W | 0h | HD PIP channel enable 0 : Disabled 1 : Enabled |
| 2 | GRPX0_EN | R/W | 0h | GRPX0 channel enable 0 : Disabled 1 : Enabled |
| 1 | GRPX1_EN | R/W | 0h | GRPX1 channel enable 0 : Disabled 1 : Enabled |
| 0 | GRPX2_EN | R/W | 0h | GRPX2 channel enable 0 : Disabled 1 : Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.3.4 COMP_sd_settings Register (offset = 10h) [reset = E400h]

COMP_sd_settings is shown in Figure 12-231 and described in Table 12-140.

**Figure 12-231. COMP_sd_settings Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | FB_SEL |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| G2_ORDER | | G1_ORDER | | G0_ORDER | | VID_ORDER | |
| R/W-3h | | R/W-2h | | R/W-1h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| G_ORDER | Reserved | VID_EN | Reserved | | GRPX0_EN | GRPX1_EN | GRPX2_EN |
| R/W-0h | R-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-140. COMP_sd_settings Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | Reserved | R | 0h | Reserved |
| 16 | FB_SEL | R/W | 0h | Feedback data selection: 0: select data from video alpha blending 1: select data from final alpha blending |
| 15-14 | G2_ORDER | R/W | 3h | Graphic2 layer display order when g_order = 1. From low to high: 00, 01, 10, and 11. |
| 13-12 | G1_ORDER | R/W | 2h | Graphic1 layer display order when g_order = 1. From low to high: 00, 01, 10, and 11. |
| 11-10 | G0_ORDER | R/W | 1h | Graphic0 layer display order when g_order = 1. From low to high: 00, 01, 10, and 11. |
| 9-8 | VID_ORDER | R/W | 0h | Video layer display order. From low to high: 00, 01, 10, and 11. |
| 7 | G_ORDER | R/W | 0h | Global reorder. 1: global reorder is on. The graphic layer priority is based on g0_reorder, g1_reorder, g2_order |
| 6 | Reserved | R | 0h | Reserved |
| 5 | VID_EN | R/W | 0h | SD VID channel Enable 0 : Disabled 1 : Enabled |
| 4-3 | Reserved | R | 0h | Reserved |
| 2 | GRPX0_EN | R/W | 0h | GRPX0 channel enable 0 : Disabled 1 : Enabled |
| 1 | GRPX1_EN | R/W | 0h | GRPX1 channel enable 0 : Disabled 1 : Enabled |
| 0 | GRPX2_EN | R/W | 0h | GRPX2 channel enable 0 : Disabled 1 : Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.3.5 COMP_back_color_settings Register (offset = 14h) [reset = 0h]

COMP_back_color_settings is shown in Figure 12-232 and described in Table 12-141.

**Figure 12-232. COMP_back_color_settings Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | BACK_CLR | | | | | |
| R-0h | | R/W-0h | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| BACK_CLR | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| BACK_CLR | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BACK_CLR | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-141. COMP_back_color_settings Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | Reserved |
| 29-0 | BACK_CLR | R/W | 0h | Background color in RGB format. This color will replace any pixel with alpha value of 0. It is also output to VENCs to be used if channel is not enabled or an underflow occurs |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.4 CSC Registers

Table 12-142 lists the memory-mapped registers for the CSC. All register offset addresses not listed in Table 12-142 should be considered as reserved locations and the register contents should not be modified.

**Table 12-142. CSC REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | CSC_csc00 | Color Space Converter Reg00 | Section 12.3.4.1 |
| 4h | CSC_csc01 | Color Space Converter Reg01 | Section 12.3.4.2 |
| 8h | CSC_csc02 | Color Space Converter Reg02 | Section 12.3.4.3 |
| Ch | CSC_csc03 | Color Space Converter Reg03 | Section 12.3.4.4 |
| 10h | CSC_csc04 | Color Space Converter Reg04 | Section 12.3.4.5 |
| 14h | CSC_csc05 | Color Space Converter Reg05 | Section 12.3.4.6 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.4.1 CSC_csc00 Register (offset = 0h) [reset = 0h]

CSC_csc00 is shown in Figure 12-233 and described in Table 12-143.

Color Space Converter Reg00

**Figure 12-233. CSC_csc00 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | B0 | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| B0 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | A0 | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| A0 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-143. CSC_csc00 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-16 | B0 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in CSC_csc00) |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-0 | A0 | R/W | 0h | Its is represented as Q3.10 number. So the value ranges from -4 to +4. To convert a decimal number, multiply the number by 1024 and write it in the register in hex format. For example, to program 0.673, 0x2B1 should be written in the register.<br>(round)(0.673 X 1024) = (round)689.152 = 689 = 0x2B1.<br>If the real number is negative, then multiply it by 1024, and convert it to 2's compliment format in 12-bit. For example, if a coefficient is -1.893, 0x186E needs to be written in the register.<br>(round)(-1.893*1024)= -1938 = 0x186E (2'S compliment format of -1938 in 13-bit width) |

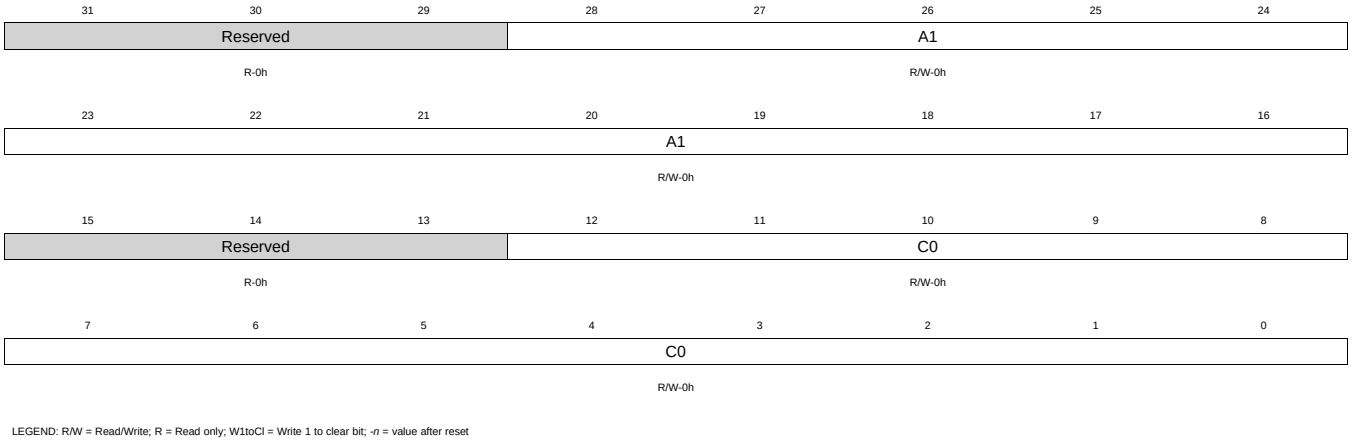

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.4.2 CSC_csc01 Register (offset = 4h) [reset = 0h]

CSC_csc01 is shown in Figure 12-234 and described in Table 12-144.

Color Space Converter Reg01

**Figure 12-234. CSC_csc01 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-144. CSC_csc01 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-16 | A1 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in CSC_csc00) |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-0 | C0 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in CSC_csc00) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.4.3 CSC_csc02 Register (offset = 8h) [reset = 0h]

CSC_csc02 is shown in Figure 12-235 and described in Table 12-145.

Color Space Converter Reg02

**Figure 12-235. CSC_csc02 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | C1 | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| C1 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | B1 | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| B1 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-145. CSC_csc02 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-16 | C1 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in CSC_csc00) |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-0 | B1 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in CSC_csc00) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.4.4 CSC_csc03 Register (offset = Ch) [reset = 0h]

CSC_csc03 is shown in Figure 12-236 and described in Table 12-146.

Color Space Converter Reg03

**Figure 12-236. CSC_csc03 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | B2 | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| B2 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | A2 | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| A2 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-146. CSC_csc03 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-16 | B2 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in CSC_csc00) |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-0 | A2 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in CSC_csc00) |

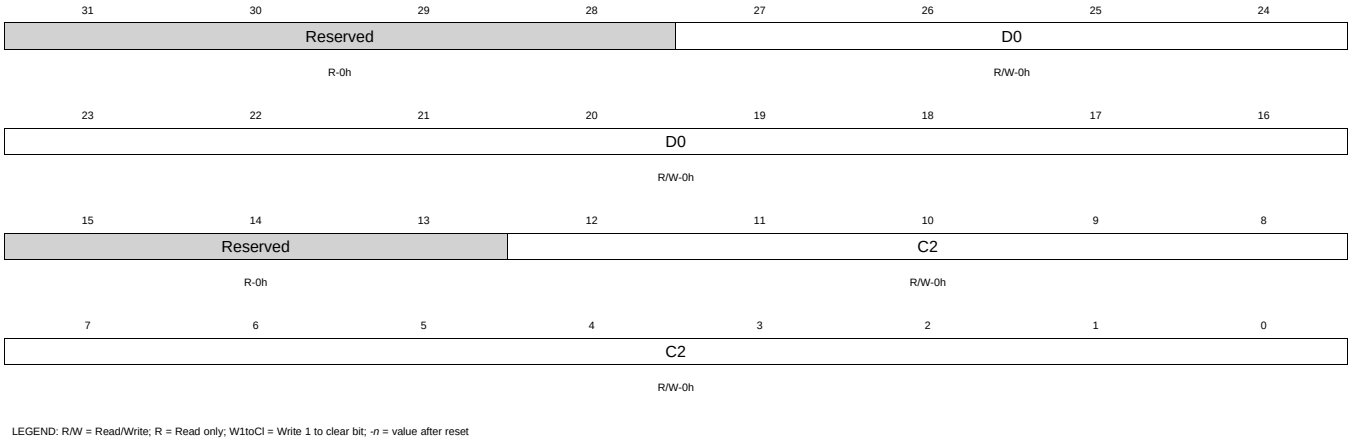

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.4.5 CSC_csc04 Register (offset = 10h) [reset = 0h]

CSC_csc04 is shown in Figure 12-237 and described in Table 12-147.

Color Space Converter Reg04

**Figure 12-237. CSC_csc04 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-147. CSC_csc04 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | Reserved |
| 27-16 | D0 | R/W | 0h | Coefficients of color space converter. This coefficient is an integer number in the range of -2048 to 2047. It is in 12-bit wide 2's compliment format. The MSB is sign bit. For example, if this coefficient is 749, then 0x2ED (hex format) should be assigned to this register. Another example, if this coefficient is -1021, then 0xC03 should be assigned to this register. |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-0 | C2 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in CSC_csc00) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.4.6 CSC_csc05 Register (offset = 14h) [reset = 0h]

CSC_csc05 is shown in Figure 12-238 and described in Table 12-148.

Color Space Converter Reg05

**Figure 12-238. CSC_csc05 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | BYPASS | D2 | | | |
| R-0h | | | R/W-0h | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| D2 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | D1 | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| D1 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-148. CSC_csc05 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28 | BYPASS | R/W | 0h | Full CSC bypass mode |
| 27-16 | D2 | R/W | 0h | Coefficients of color space converter. This coefficient is an integer number in the range of -2048 to 2047. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0 in CSC_csc04) |
| 15-12 | Reserved | R | 0h | Reserved |
| 11-0 | D1 | R/W | 0h | Coefficients of color space converter. This coefficient is an integer number in the range of -2048 to 2047. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0 in CSC_csc04) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5 DEI Registers

Table 12-149 lists the memory-mapped registers for the DEI. All register offset addresses not listed in Table 12-149 should be considered as reserved locations and the register contents should not be modified.

**Table 12-149. DEI REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | dei_reg0 | Frame Size Register | Section 12.3.5.1 |
| 4h | dei_reg1 | MDT Filter Bypass Register | Section 12.3.5.2 |
| 8h | dei_reg2 | MDT Spatial Frequency Threshold Register | Section 12.3.5.3 |
| Ch | dei_reg3 | EDI Configuration Control | Section 12.3.5.4 |
| 10h | dei_reg4 | EDI Lookup Table Register 0 | Section 12.3.5.5 |
| 14h | dei_reg5 | EDI Lookup Table Register 1 | Section 12.3.5.6 |
| 18h | dei_reg6 | EDI Lookup Table Register 2 | Section 12.3.5.7 |
| 1Ch | dei_reg7 | EDI Lookup Table Register 3 | Section 12.3.5.8 |
| 20h | dei_reg8 | FMD Window Register 0 | Section 12.3.5.9 |
| 24h | dei_reg9 | FMD Window Register 1 | Section 12.3.5.10 |
| 28h | dei_reg10 | FMD Control Register 0 | Section 12.3.5.11 |
| 2Ch | dei_reg11 | FMD Control Register 0 | Section 12.3.5.12 |
| 30h | dei_reg12 | FMD Status Register 0 | Section 12.3.5.13 |
| 34h | dei_reg13 | FMD Status Register 1 | Section 12.3.5.14 |
| 38h | dei_reg14 | FMD Status Register 2 | Section 12.3.5.15 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.1 dei_reg0 Register (offset = 0h) [reset = 01E002D0h]

dei_reg0 is shown in Figure 12-239 and described in Table 12-150.

Frame Size Register

**Figure 12-239. dei_reg0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | FIELD_FLUSH | BYPASS | Reserved | | | HEIGHT | |
| R-0h | R/W-0h | R/W-0h | R-0h | | | R/W-1E0h | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| HEIGHT | | | | | | | |
| R/W-1E0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | WIDTH | | |
| R-0h | | | | | R/W-2D0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WIDTH | | | | | | | |
| R/W-2D0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-150. dei_reg0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Reserved | R | 0h | Reserved |
| 30 | FIELD_FLUSH | R/W | 0h | Field Flush Mode 0: Normal Operation 1: Flush Internal Pipe for Current output Frame |
| 29 | BYPASS | R/W | 0h | Bypass Mode 0: Normal Deinterlace Mode 1: Input Source bypassed directly to output |
| 28-27 | Reserved | R | 0h | Reserved |
| 26-16 | HEIGHT | R/W | 1E0h | Frame height |
| 15-11 | Reserved | R | 0h | Reserved |
| 10-0 | WIDTH | R/W | 2D0h | Frame Width |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.2 dei_reg1 Register (offset = 4h) [reset = 0h]

dei_reg1 is shown in Figure 12-240 and described in Table 12-151.

MDT Filter Bypass Register

**Figure 12-240. dei_reg1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | MDT_SPATMAX_BYPASS | MDT_TEMPMAX_BYPASS |
| R-0h | | | | | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-151. dei_reg1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0h | Reserved |
| 1 | MDT_SPATMAX_BYPASS | R/W | 0h | Spatial Maximum Filtering Bypass for motion values used in EDI 0: Enable 1: Bypass |
| 0 | MDT_TEMPMAX_BYPASS | R/W | 0h | Spatio-temporal Maximum Filtering Bypass for motion valued used in EDI 0: Enable 1: Bypass |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.3 dei_reg2 Register (offset = 8h) [reset = 0C0C0804h]

dei_reg2 is shown in Figure 12-241 and described in Table 12-152.

MDT Spatial Frequency Threshold Register

**Figure 12-241. dei_reg2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MDT_MVSTMAX_COR_THR | | | | MDT_MV_COR_THR | | | |
| R/W-0h | | | | R/W-Ch | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MDT_SF_SC_THR3 | | | | | | | |
| R/W-Ch | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MDT_SF_SC_THR2 | | | | | | | |
| R/W-8h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MDT_SF_SC_THR1 | | | | | | | |
| R/W-4h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-152. dei_reg2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | MDT_MVSTMAX_COR_THR | R/W | 0h | This is used for increasing noise robustness. Increasing this threshold leads to more robustness to noise, but with the potential of introducing ghosting effect. Note that this threshold is used for motion values for EDI only, and it is in addition mdt_mv_cor_thr. |
| 27-24 | MDT_MV_COR_THR | R/W | Ch | This threshold is for the coring for motion value, mv. MDT will become more noise robust if this value increases. But the picture may be washed out if this value is set to high. This threshold can be interpreted as the noise threshold for calculating motion values for all blocks. |
| 23-16 | MDT_SF_SC_THR3 | R/W | Ch | Spatial frequency threshold 3 |
| 15-8 | MDT_SF_SC_THR2 | R/W | 8h | Spatial frequency threshold 2 |
| 7-0 | MDT_SF_SC_THR1 | R/W | 4h | Spatial frequency threshold It is used for adaptive scaling of motion values according to how busy the texture is. If the texture is flat, motion values need to be scaled up to reflect the sensitivity of motion values with respect to the detection error. Increasing the thresholds will make the motion value scaling more sensitive to the frequency of the texture.<br>Note: 0 <= mdt_sf_sc_thr1 <= mdt_sf_sc_thr2 <= mdt_sf_sc_thr3 |



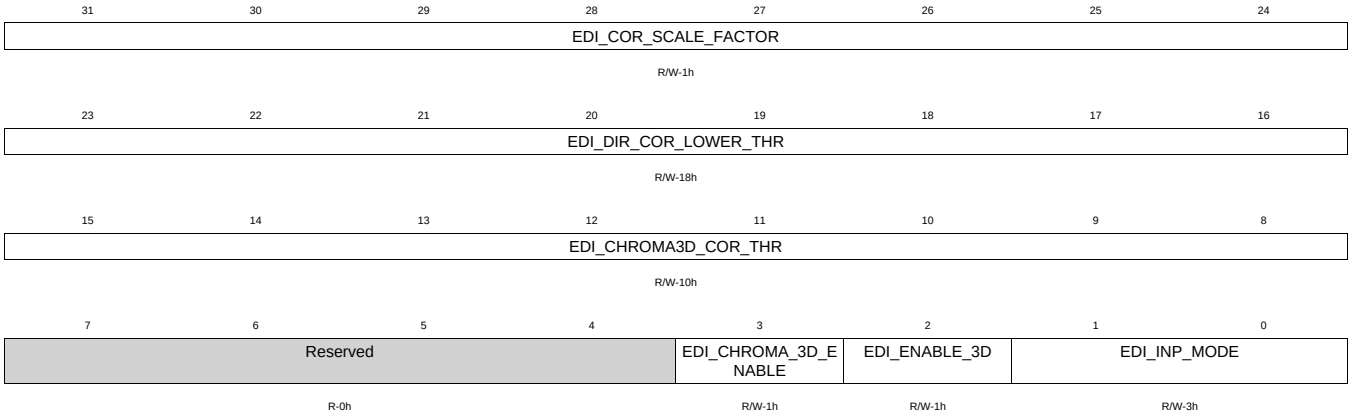
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.4 dei_reg3 Register (offset = Ch) [reset = 0118100Fh]

dei_reg3 is shown in Figure 12-242 and described in Table 12-153.

EDI Configuration Control

**Figure 12-242. dei_reg3 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-153. dei_reg3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | EDI_COR_SCALE_FACTOR | R/W | 1h | Scaling factor for correlation along detected edge |
| 23-16 | EDI_DIR_COR_LOWER_THR | R/W | 18h | Lower threshold used for correlation along detected edge |
| 15-8 | EDI_CHROMA3D_COR_THR | R/W | 10h | Correlation threshold used in 3D processing for chroma. Because the motion values used for chroma 3D processing are based on luma only. Extra protection is needed. Temporal interpolation is only performed for chroma, when there is strong spatial or temporal correlation for the chroma pixel being processed. When the pixel difference is less than this threshold, it is assumed that there exists strong correlation between these two pixels. Thus, increasing this value leads to more chroma pixels being processed in 3D. |
| 7-4 | Reserved | R | 0h | Reserved |
| 3 | EDI_CHROMA_3D_ENABLE | R/W | 1h | 3D Chroma Enable<br>0: Disable 3D processing for chroma<br>1: Enable 3D processing (temporal interpolation) |
| 2 | EDI_ENABLE_3D | R/W | 1h | 3D Enable<br>0: Disable 3D processing<br>1: Enable 3D processing (temporal interpolation) |
| 1-0 | EDI_INP_MODE | R/W | 3h | Interpolation mode<br>00: line average<br>01: field average<br>10: edge-directed interpolation for luma only<br>11: edge-directed interpolation for both luma and chroma<br>Note that mode 00 and 01 are used for debug purpose |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.5 dei_reg4 Register (offset = 10h) [reset = 08040200h]

dei_reg4 is shown in Figure 12-243 and described in Table 12-154.

EDI Lookup Table Register 0

**Figure 12-243. dei_reg4 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | EDI_LUT3 | | | | |
| R-0h | | | R/W-8h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | EDI_LUT2 | | | | |
| R-0h | | | R/W-4h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | EDI_LUT1 | | | | |
| R-0h | | | R/W-2h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | EDI_LUT0 | | | | |
| R-0h | | | R/W-0h | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-154. dei_reg4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-24 | EDI_LUT3 | R/W | 8h | EDI Lookup Table 3 |
| 23-21 | Reserved | R | 0h | Reserved |
| 20-16 | EDI_LUT2 | R/W | 4h | EDI Lookup Table 2 |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-8 | EDI_LUT1 | R/W | 2h | EDI Lookup Table 1 |
| 7-5 | Reserved | R | 0h | Reserved |
| 4-0 | EDI_LUT0 | R/W | 0h | EDI Lookup Table 0 |

**Note:** 0<=EDI_LUT0 <=EDI_LUT1 <=EDI_LUT2 <=EDI_LUT3.....<=EDI_LUT15



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.6 dei_reg5 Register (offset = 14h) [reset = 1010100Ch]

dei_reg5 is shown in Figure 12-244 and described in Table 12-155.

EDI Lookup Table Register 1

**Figure 12-244. dei_reg5 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | EDI_LUT7 | | | | |
| R-0h | | | R/W-10h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | EDI_LUT6 | | | | |
| R-0h | | | R/W-10h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | EDI_LUT5 | | | | |
| R-0h | | | R/W-10h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | EDI_LUT4 | | | | |
| R-0h | | | R/W-Ch | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-155. dei_reg5 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-24 | EDI_LUT7 | R/W | 10h | EDI Lookup Table 7 |
| 23-21 | Reserved | R | 0h | Reserved |
| 20-16 | EDI_LUT6 | R/W | 10h | EDI Lookup Table 6 |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-8 | EDI_LUT5 | R/W | 10h | EDI Lookup Table 5 |
| 7-5 | Reserved | R | 0h | Reserved |
| 4-0 | EDI_LUT4 | R/W | Ch | EDI Lookup Table 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.7 dei_reg6 Register (offset = 18h) [reset = 10101010h]

dei_reg6 is shown in Figure 12-245 and described in Table 12-156.

EDI Lookup Table Register 2

**Figure 12-245. dei_reg6 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | EDI_LUT11 | | | | |
| R-0h | | | R/W-10h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | EDI_LUT10 | | | | |
| R-0h | | | R/W-10h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | EDI_LUT9 | | | | |
| R-0h | | | R/W-10h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | EDI_LUT8 | | | | |
| R-0h | | | R/W-10h | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-156. dei_reg6 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-24 | EDI_LUT11 | R/W | 10h | EDI Lookup Table 11 |
| 23-21 | Reserved | R | 0h | Reserved |
| 20-16 | EDI_LUT10 | R/W | 10h | EDI Lookup Table 10 |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-8 | EDI_LUT9 | R/W | 10h | EDI Lookup Table 9 |
| 7-5 | Reserved | R | 0h | Reserved |
| 4-0 | EDI_LUT8 | R/W | 10h | EDI Lookup Table 8 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.8 dei_reg7 Register (offset = 1Ch) [reset = 10101010h]

dei_reg7 is shown in Figure 12-246 and described in Table 12-157.

EDI Lookup Table Register 3

**Figure 12-246. dei_reg7 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | EDI_LUT15 | | | | |
| R-0h | | | R/W-10h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | EDI_LUT14 | | | | |
| R-0h | | | R/W-10h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | EDI_LUT13 | | | | |
| R-0h | | | R/W-10h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | EDI_LUT12 | | | | |
| R-0h | | | R/W-10h | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-157. dei_reg7 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-24 | EDI_LUT15 | R/W | 10h | EDI Lookup Table 15 |
| 23-21 | Reserved | R | 0h | Reserved |
| 20-16 | EDI_LUT14 | R/W | 10h | EDI Lookup Table 14 |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-8 | EDI_LUT13 | R/W | 10h | EDI Lookup Table 13 |
| 7-5 | Reserved | R | 0h | Reserved |
| 4-0 | EDI_LUT12 | R/W | 10h | EDI Lookup Table 12 |

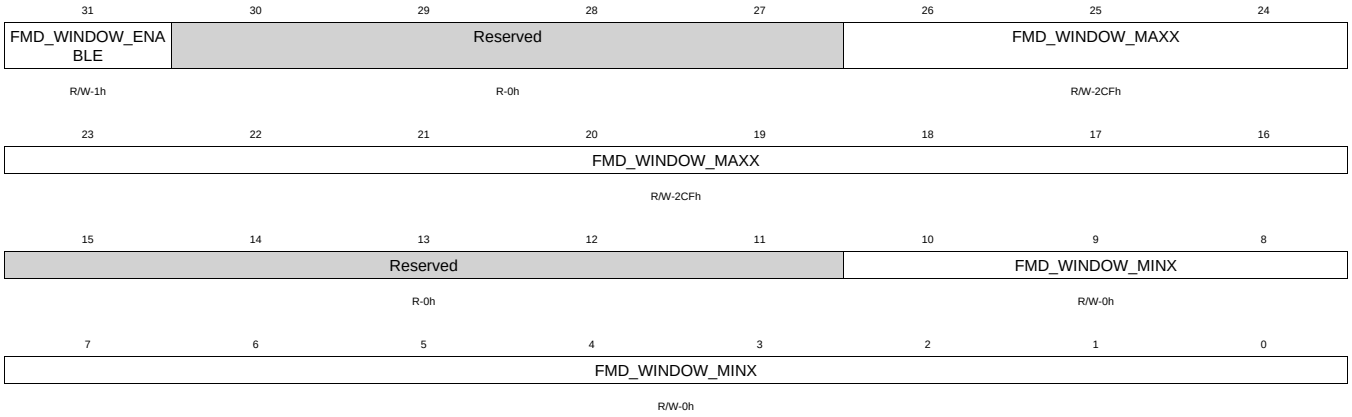

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.9 dei_reg8 Register (offset = 20h) [reset = 82CF0000h]

dei_reg8 is shown in Figure 12-247 and described in Table 12-158.

FMD Window Register 0

**Figure 12-247. dei_reg8 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-158. dei_reg8 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | FMD_WINDOW_ENABLE | R/W | 1h | Enable FMD operation window |
| 30-27 | Reserved | R | 0h | Reserved |
| 26-16 | FMD_WINDOW_MAXX | R/W | 2CFh | Right boundary of FMD operation window Must be less than width |
| 15-11 | Reserved | R | 0h | Reserved |
| 10-0 | FMD_WINDOW_MINX | R/W | 0h | Left boundary of FMD operation window |



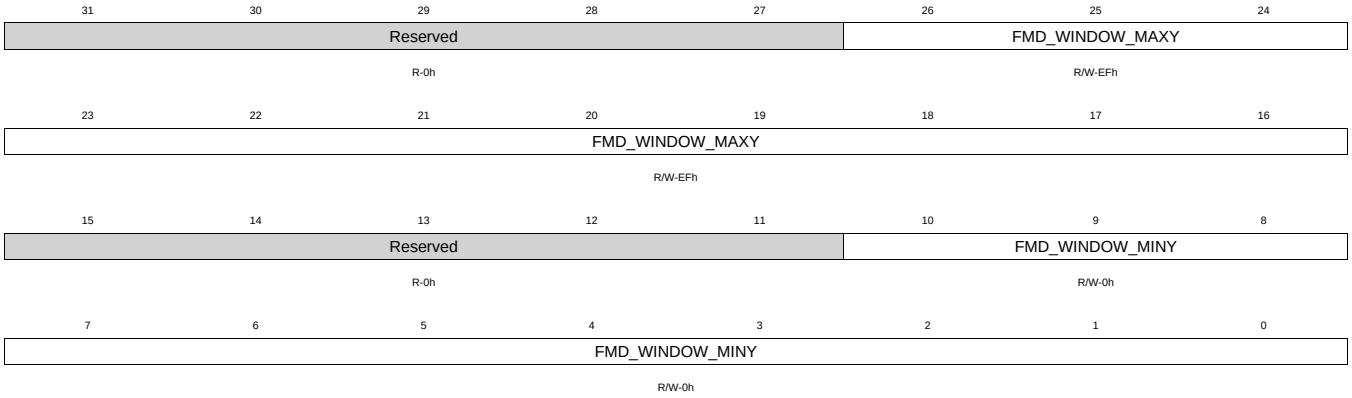
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.10 dei_reg9 Register (offset = 24h) [reset = 00EF0000h]

dei_reg9 is shown in Figure 12-248 and described in Table 12-159.

FMD Window Register 1

**Figure 12-248. dei_reg9 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-159. dei_reg9 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | Reserved |
| 26-16 | FMD_WINDOW_MAXY | R/W | EFh | Bottom boundary of FMD operation window Must be less than height/2 |
| 15-11 | Reserved | R | 0h | Reserved |
| 10-0 | FMD_WINDOW_MINY | R/W | 0h | Top boundary of FMD operation window |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.11 dei_reg10 Register (offset = 28h) [reset = 080A0000h]

dei_reg10 is shown in Figure 12-249 and described in Table 12-160.

FMD Control Register 0

**Figure 12-249. dei_reg10 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| FMD_CAF_LINE_THR | | | | | | | |
| R/W-8h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FMD_CAF_FIELD_THR | | | | | | | |
| R/W-Ah | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | FMD_BED_ENABLE | FMD_JAM_DIR | FMD_LOCK | FMD_ENABLE |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-160. dei_reg10 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | FMD_CAF_LINE_THR | R/W | 8h | CAF (Combing Artifacts) threshold used for the pixels from two lines in one field This is the threshold used for combing artifacts detection. The difference of two consecutive lines from the same field (so there is one line in between if two fields are merged into one progressive frame) is compared with this threshold. Decreasing this threshold leads to be more conservative in detecting CAF. Both fmd_caf_field_thr and fmd_caf_line_thr are close the values that two pixels differed by this value is observable. |
| 23-16 | FMD_CAF_FIELD_THR | R/W | Ah | CAF (Combing Artifacts) threshold used for the pixels from two fields This is the threshold used for combing artifacts detection. The difference of two consecutive lines (when merging two fields into one progressive frame) is used to compare with this threshold. Increasing this threshold leads to be more conservative in detecting CAF. |
| 15-4 | Reserved | R | 0h | Reserved |
| 3 | FMD_BED_ENABLE | R/W | 0h | Film Mode Bad Edit Detection<br>0: Disable<br>1: Enable |
| 2 | FMD_JAM_DIR | R/W | 0h | Film Mode Field Jamming Direction<br>0: Current field jammed with previous field<br>1: Current field jammed with next field |
| 1 | FMD_LOCK | R/W | 0h | Lock Deinterlacer to film mode<br>0: Standard Deinterlacer Processing<br>1: Film Mode Deinterlacer Processing |
| 0 | FMD_ENABLE | R/W | 0h | Enable film mode processing<br>0: Disable<br>1: Enable |



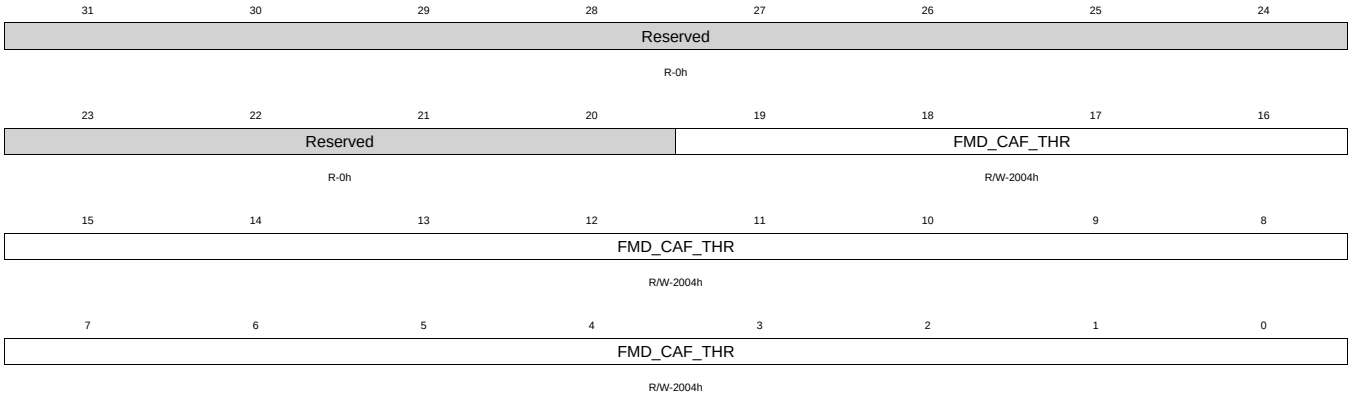
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.12 dei_reg11 Register (offset = 2Ch) [reset = 00002004h]

dei_reg11 is shown in Figure 12-250 and described in Table 12-161.

FMD Control Register 0

**Figure 12-250. dei_reg11 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-161. dei_reg11 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | Reserved |
| 19-0 | FMD_CAF_THR | R/W | 2004h | CAF threshold used for leaving film mode as part of bad edit detection (only used if fmd_bed_enable is active): If the combing artifacts is greater than this threshold, CAF is detected and thus the state machine will be forced to leave the film mode. If the user prefers to be more conservative in using film mode, decrease this threshold. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.13 dei_reg12 Register (offset = 30h) [reset = 0h]

dei_reg12 is shown in Figure 12-251 and described in Table 12-162.

FMD Status Register 0

**Figure 12-251. dei_reg12 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | FMD_RESET |
| R-0h | | | | | | | R-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | FMD_CAF | | | | |
| R-0h | | | R-0h | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FMD_CAF | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FMD_CAF | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-162. dei_reg12 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | Reserved | R | 0h | Reserved |
| 24 | FMD_RESET | R | 0h | When 1 , the film mode detection module needs to be reset by the software. This bit needs to be checked at each occurrence of the film mode detection interrupt. It will only be active in bad edit detection mode (fmd_bed_enable = 1) and the design is currently locked into film mode. |
| 23-21 | Reserved | R | 0h | Reserved |
| 20-0 | FMD_CAF | R | 0h | Detected combing artifacts |

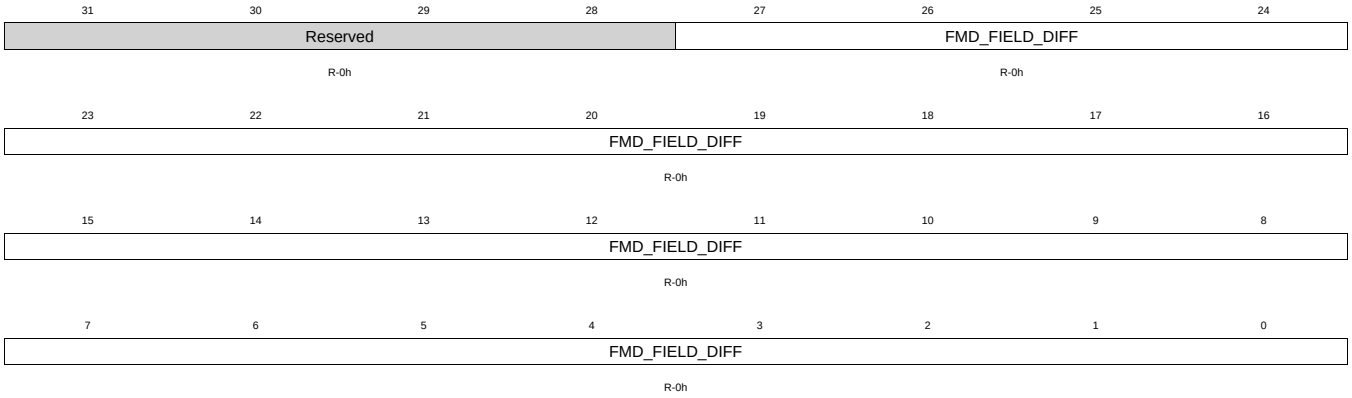

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.14 dei_reg13 Register (offset = 34h) [reset = 0h]

dei_reg13 is shown in Figure 12-252 and described in Table 12-163.

FMD Status Register 1

**Figure 12-252. dei_reg13 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-163. dei_reg13 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | Reserved |
| 27-0 | FMD_FIELD_DIFF | R | 0h | Field difference (difference between two neighboring fields, one top and one bottom) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.5.15 dei_reg14 Register (offset = 38h) [reset = 0h]

dei_reg14 is shown in Figure 12-253 and described in Table 12-164.

FMD Status Register 2

**Figure 12-253. dei_reg14 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | FMD_FRAME_DIFF | | | |
| R-0h | | | | R-0h | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FMD_FRAME_DIFF | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FMD_FRAME_DIFF | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-164. dei_reg14 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | Reserved |
| 19-0 | FMD_FRAME_DIFF | R | 0h | Frame difference (difference between two top or two bottom fields) |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.6 GRPX Registers

The graphics module (GRPX) is configured using VPDMA frame and region descriptors. Please refer to Section 12.2.5 and Section 12.2.13.5.1.5 for instructions on how to configure the graphics module.


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7 INTC_CLKC_CONTROL Registers

Table 12-165 lists the memory-mapped registers for the INTC_CLKC_CONTROL. All register offset addresses not listed in Table 12-165 should be considered as reserved locations and the register contents should not be modified.

**Table 12-165. INTC_CLKC_CONTROL REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | pid | PID Register | Section 12.3.7.1 |
| 10h | sysconfig | SYS Config Register | Section 12.3.7.2 |
| 20h | intc_intr0_status_raw0 | Interrupt0 Raw Register 0 | Section 12.3.7.3 |
| 24h | intc_intr0_status_raw1 | Interrupt0 Raw Register 1 | Section 12.3.7.4 |
| 28h | intc_intr0_status_ena0 | Interrupt0 Enabled Register 0 | Section 12.3.7.5 |
| 2Ch | intc_intr0_status_ena1 | Interrupt0 Enabled Register 1 | Section 12.3.7.6 |
| 30h | intc_intr0_ena_set0 | Interrupt0 Enable/Set Register 0 | Section 12.3.7.7 |
| 34h | intc_intr0_ena_set1 | Interrupt0 Enable/Set Register 1 | Section 12.3.7.8 |
| 38h | intc_intr0_ena_clr0 | Interrupt0 Enable/Clear Register 0 | Section 12.3.7.9 |
| 3Ch | intc_intr0_ena_clr1 | Interrupt0 Enable/Clear Register 1 | Section 12.3.7.10 |
| 40h | intc_intr1_status_raw0 | Interrupt1 Raw Register 0 | Section 12.3.7.11 |
| 44h | intc_intr1_status_raw1 | Interrupt1 Raw Register 1 | Section 12.3.7.12 |
| 48h | intc_intr1_status_ena0 | Interrupt1 Enabled Register 0 | Section 12.3.7.13 |
| 4Ch | intc_intr1_status_ena1 | Interrupt1 Enabled Register 1 | Section 12.3.7.14 |
| 50h | intc_intr1_ena_set0 | Interrupt1 Enable/Set Register 0 | Section 12.3.7.15 |
| 54h | intc_intr1_ena_set1 | Interrupt1 Enable/Set Register 1 | Section 12.3.7.16 |
| 58h | intc_intr1_ena_clr0 | Interrupt1 Enable/Clear Register 0 | Section 12.3.7.17 |
| 5Ch | intc_intr1_ena_clr1 | Interrupt1 Enable/Clear Register 1 | Section 12.3.7.18 |
| 60h | intc_intr2_status_raw0 | Interrupt2 Raw Register 0 | Section 12.3.7.19 |
| 64h | intc_intr2_status_raw1 | Interrupt2 Raw Register 1 | Section 12.3.7.20 |
| 68h | intc_intr2_status_ena0 | Interrupt2 Enabled Register 0 | Section 12.3.7.21 |
| 6Ch | intc_intr2_status_ena1 | Interrupt2 Enabled Register 1 | Section 12.3.7.22 |
| 70h | intc_intr2_ena_set0 | Interrupt2 Enable/Set Register 0 | Section 12.3.7.23 |
| 74h | intc_intr2_ena_set1 | Interrupt2 Enable/Set Register 1 | Section 12.3.7.24 |
| 78h | intc_intr2_ena_clr0 | Interrupt2 Enable/Clear Register 0 | Section 12.3.7.25 |
| 7Ch | intc_intr2_ena_clr1 | Interrupt2 Enable/Clear Register 1 | Section 12.3.7.26 |
| 80h | intc_intr3_status_raw0 | Interrupt3 Raw Register 0 | Section 12.3.7.27 |
| 84h | intc_intr3_status_raw1 | Interrupt3 Raw Register 0 | Section 12.3.7.28 |
| 88h | intc_intr3_status_ena0 | Interrupt3 Enabled Register 0 | Section 12.3.7.29 |
| 8Ch | intc_intr3_status_ena1 | Interrupt3 Enabled Register 1 | Section 12.3.7.30 |
| 90h | intc_intr3_ena_set0 | Interrupt3 Enable/Set Register 0 | Section 12.3.7.31 |
| 94h | intc_intr3_ena_set1 | Interrupt3 Enable/Set Register 1 | Section 12.3.7.32 |
| 98h | intc_intr3_ena_clr0 | Interrupt3 Enable/Clear Register 0 | Section 12.3.7.33 |
| 9Ch | intc_intr3_ena_clr1 | Interrupt3 Enable/Clear Register 1 | Section 12.3.7.34 |
| A0h | intc_eoi | End of Interrupt Register | Section 12.3.7.35 |
| 100h | clkc_clken | Clock Enable Register | Section 12.3.7.36 |
| 104h | clkc_rst | Soft Reset Register | |
| 108h | clkc_dps | Main Datapath Select Register | Section 12.3.7.37 |
| 10Ch | clkc_vip1dps | VIP1 Datapath Select Register | Section 12.3.7.38 |
| 110h | clkc_vip2dps | VIP2 Datapath Select Register | Section 12.3.7.39 |
| 114h | clkc_venc_clksel | VENC Clock Select Register | |
| 118h | clkc_venc_ena | VENC Enable Register | Section 12.3.7.40 |
| 11Ch | clkc_range_map | VC1 Range Map Register | Section 12.3.7.41 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-165. INTC_CLKC_CONTROL REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 120h | clkc_underflow | VENC Underflow Status Register | Section 12.3.7.42 |

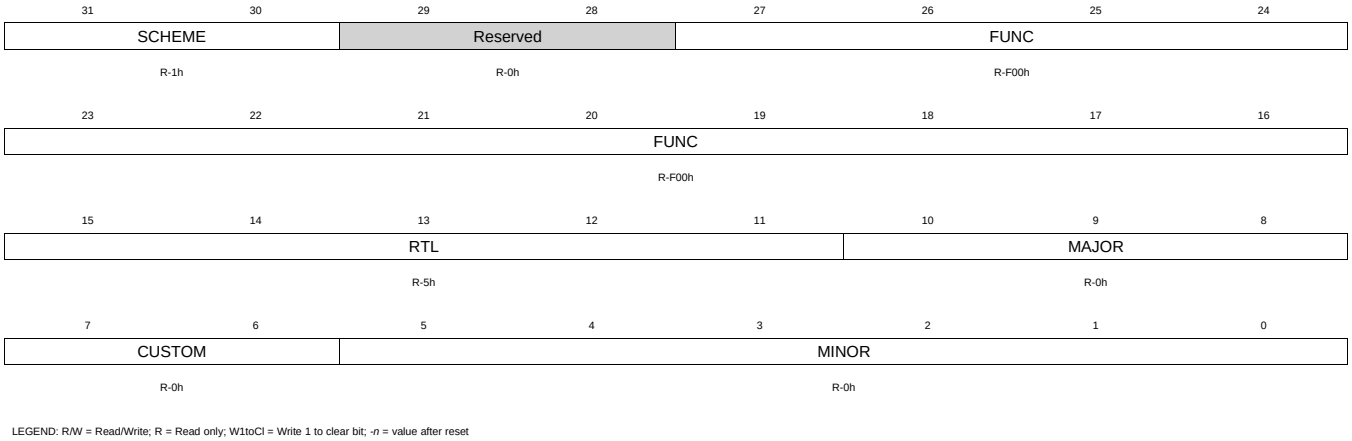

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.1 pid Register (offset = 0h) [reset = 4F002800h]

pid is shown in Figure 12-254 and described in Table 12-166.

PID Register

**Figure 12-254. pid Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-166. pid Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | SCHEME | R | 1h | The scheme of the register used. This indicates the PDR3.5 Method |
| 29-28 | Reserved | R | 0h | |
| 27-16 | FUNC | R | F00h | The function of the module being used |
| 15-11 | RTL | R | 5h | RTL Release Version The PDR release number of this IP |
| 10-8 | MAJOR | R | 0h | Major Release Number |
| 7-6 | CUSTOM | R | 0h | Custom IP |
| 5-0 | MINOR | R | 0h | Minor Release Number |



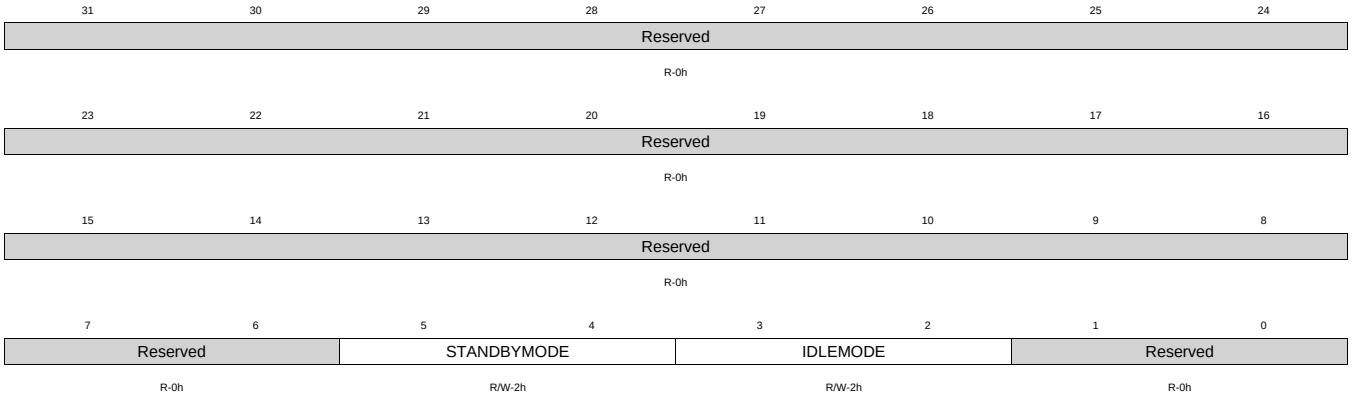
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.2 sysconfig Register (offset = 10h) [reset = 28h]

sysconfig is shown in Figure 12-255 and described in Table 12-167.

SYS Config Register

**Figure 12-255. sysconfig Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-167. sysconfig Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-4 | STANDBYMODE | R/W | 2h | Standymode setting for PWRSTNDBY IPGeneric |
| 3-2 | IDLEMODE | R/W | 2h | Idlemode setting for PWRIDLE IPGenerc |
| 1-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.3 intc_intr0_status_raw0 Register (offset = 20h) [reset = 0h]

intc_intr0_status_raw0 is shown in Figure 12-256 and described in Table 12-168.

Interrupt0 Raw Register 0

**Figure 12-256. intc_intr0_status_raw0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_RAW | DVO2_INT2_RAW | DVO2_INT1_RAW | DVO2_INT0_RAW | Reserved | | | DVO1_INT2_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_RAW | DVO1_INT0_RAW | VIP2_PARSER_INT_RAW | VIP1_PARSER_INT_RAW | Reserved | DEI_FMD_INT_RAW | Reserved | VPDMA_INT0_DESCRIPTOR_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT0_LIST7_NOTIFY_RAW | VPDMA_INT0_LIST7_COMPLETE_RAW | VPDMA_INT0_LIST6_NOTIFY_RAW | VPDMA_INT0_LIST6_COMPLETE_RAW | VPDMA_INT0_LIST5_NOTIFY_RAW | VPDMA_INT0_LIST5_COMPLETE_RAW | VPDMA_INT0_LIST4_NOTIFY_RAW | VPDMA_INT0_LIST4_COMPLETE_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT0_LIST3_NOTIFY_RAW | VPDMA_INT0_LIST3_COMPLETE_RAW | VPDMA_INT0_LIST2_NOTIFY_RAW | VPDMA_INT0_LIST2_COMPLETE_RAW | VPDMA_INT0_LIST1_NOTIFY_RAW | VPDMA_INT0_LIST1_COMPLETE_RAW | VPDMA_INT0_LIST0_NOTIFY_RAW | VPDMA_INT0_LIST0_COMPLETE_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-168. intc_intr0_status_raw0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_RAW | R/W | 0h | SD VENC Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 30 | DVO2_INT2_RAW | R/W | 0h | DVO2 Interrupt2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 29 | DVO2_INT1_RAW | R/W | 0h | DVO2 Interrupt1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 28 | DVO2_INT0_RAW | R/W | 0h | DVO2 Interrupt0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_RAW | R/W | 0h | DVO1 Interrupt2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 23 | DVO1_INT1_RAW | R/W | 0h | DVO1 Interrupt1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 22 | DVO1_INT0_RAW | R/W | 0h | DVO1 Interrupt0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_RAW | R/W | 0h | VIP2 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_RAW | R/W | 0h | VIP1 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_RAW | R/W | 0h | DEI Film Mode Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 17 | Reserved | R | 0h | |
| 16 | VPDMA_INT0_DESCRIPTOR_RAW | R/W | 0h | VPDMA INT0 Descriptor Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 15 | VPDMA_INT0_LIST7_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 14 | VPDMA_INT0_LIST7_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-168. intc_intr0_status_raw0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | VPDMA_INT0_LIST6_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 12 | VPDMA_INT0_LIST6_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 11 | VPDMA_INT0_LIST5_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 10 | VPDMA_INT0_LIST5_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 9 | VPDMA_INT0_LIST4_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 8 | VPDMA_INT0_LIST4_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 7 | VPDMA_INT0_LIST3_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 6 | VPDMA_INT0_LIST3_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT0_LIST2_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT0_LIST2_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT0_LIST1_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT0_LIST1_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT0_LIST0_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List0 Notify Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT0_LIST0_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List0 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.4 intc_intr0_status_raw1 Register (offset = 24h) [reset = 0h]

intc_intr0_status_raw1 is shown in Figure 12-257 and described in Table 12-169.

Interrupt0 Raw Register 1

**Figure 12-257. intc_intr0_status_raw1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_RAW | VIP2_CHR_DS_1_UV_ERR_INT_RAW |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_RAW | VIP1_CHR_DS_1_UV_ERR_INT_RAW | NF_CHR_DS_UV_ERR_INT_RAW | COMP_ERR_INT_RAW | GRPX3_INT_RAW | GRPX2_INT_RAW | GRPX1_INT_RAW | DEI_ERROR_INT_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT0_CLIENT_RAW | VPDMA_INT0_CHANNEL_GROUP6_RAW | VPDMA_INT0_CHANNEL_GROUP5_RAW | VPDMA_INT0_CHANNEL_GROUP4_RAW | VPDMA_INT0_CHANNEL_GROUP3_RAW | VPDMA_INT0_CHANNEL_GROUP2_RAW | VPDMA_INT0_CHANNEL_GROUP1_RAW | VPDMA_INT0_CHANNEL_GROUP0_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-169. intc_intr0_status_raw1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_RAW | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_RAW | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_RAW | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_RAW | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_RAW | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 20 | COMP_ERR_INT_RAW | R/W | 0h | COMP Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 19 | GRPX3_INT_RAW | R/W | 0h | GRPX3 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 18 | GRPX2_INT_RAW | R/W | 0h | GRPX2 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 17 | GRPX1_INT_RAW | R/W | 0h | GRPX1 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 16 | DEI_ERROR_INT_RAW | R/W | 0h | DEI Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT0_CLIENT_RAW | R/W | 0h | VPDMA INT0 Client Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-169. intc_intr0_status_raw1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | VPDMA_INT0_CHANNEL_GROUP6_RAW | R/W | 0h | VPDMA INT0 Channel Group6 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT0_CHANNEL_GROUP5_RAW | R/W | 0h | VPDMA INT0 Channel Group5 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT0_CHANNEL_GROUP4_RAW | R/W | 0h | VPDMA INT0 Channel Group4 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT0_CHANNEL_GROUP3_RAW | R/W | 0h | VPDMA INT0 Channel Group3 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT0_CHANNEL_GROUP2_RAW | R/W | 0h | VPDMA INT0 Channel Group2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT0_CHANNEL_GROUP1_RAW | R/W | 0h | VPDMA INT0 Channel Group1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT0_CHANNEL_GROUP0_RAW | R/W | 0h | VPDMA INT0 Channel Group0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.5 intc_intr0_status_ena0 Register (offset = 28h) [reset = 0h]

intc_intr0_status_ena0 is shown in Figure 12-258 and described in Table 12-170.

Interrupt0 Enabled Register 0

**Figure 12-258. intc_intr0_status_ena0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA | DVO2_INT2_ENA | DVO2_INT1_ENA | DVO2_INT0_ENA | Reserved | | | DVO1_INT2_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA | DVO1_INT0_ENA | VIP2_PARSER_INT_ENA | VIP1_PARSER_INT_ENA | Reserved | DEI_FMD_INT_ENA | Reserved | VPDMA_INT0_DESCRIPTOR_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT0_LIST7_NOTIFY_ENA | VPDMA_INT0_LIST7_COMPLETE_ENA | VPDMA_INT0_LIST6_NOTIFY_ENA | VPDMA_INT0_LIST6_COMPLETE_ENA | VPDMA_INT0_LIST5_NOTIFY_ENA | VPDMA_INT0_LIST5_COMPLETE_ENA | VPDMA_INT0_LIST4_NOTIFY_ENA | VPDMA_INT0_LIST4_COMPLETE_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT0_LIST3_NOTIFY_ENA | VPDMA_INT0_LIST3_COMPLETE_ENA | VPDMA_INT0_LIST2_NOTIFY_ENA | VPDMA_INT0_LIST2_COMPLETE_ENA | VPDMA_INT0_LIST1_NOTIFY_ENA | VPDMA_INT0_LIST1_COMPLETE_ENA | VPDMA_INT0_LIST0_NOTIFY_ENA | VPDMA_INT0_LIST0_COMPLETE_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-170. intc_intr0_status_ena0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA | R/W | 0h | SD_VENC Interrupt Enabled Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 30 | DVO2_INT2_ENA | R/W | 0h | DVO2 Enabled Interrupt2 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 29 | DVO2_INT1_ENA | R/W | 0h | DVO2 Enabled Interrupt1 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 28 | DVO2_INT0_ENA | R/W | 0h | DVO2 Enabled Interrupt0 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA | R/W | 0h | DVO1 Enabled Interrupt2 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 23 | DVO1_INT1_ENA | R/W | 0h | DVO1 Enabled Interrupt1 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 22 | DVO1_INT0_ENA | R/W | 0h | DVO1 Enabled Interrupt0 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA | R/W | 0h | VIP2 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA | R/W | 0h | VIP1 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 19 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-170. intc_intr0_status_ena0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | DEI_FMD_INT_ENA | R/W | 0h | DEI Film Mode Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |
| 16 | VPDMA_INT0_DESCRIPTOR_ENA | R/W | 0h | VPDMA INT0 Descriptor Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT0_LIST7_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List7 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT0_LIST7_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List7 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT0_LIST6_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List6 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT0_LIST6_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List6 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT0_LIST5_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List5 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT0_LIST5_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List5 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT0_LIST4_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List4 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT0_LIST4_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List4 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT0_LIST3_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List3 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT0_LIST3_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List3 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT0_LIST2_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List2 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT0_LIST2_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List2 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT0_LIST1_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List1 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT0_LIST1_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List1 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT0_LIST0_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List0 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT0_LIST0_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List0 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.6 intc_intr0_status_ena1 Register (offset = 2Ch) [reset = 0h]

intc_intr0_status_ena1 is shown in Figure 12-259 and described in Table 12-171.

Interrupt0 Enabled Register 1

**Figure 12-259. intc_intr0_status_ena1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_ENA | VIP2_CHR_DS_1_UV_ERR_INT_ENA |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| VIP1_CHR_DS_2_UV_ERR_INT_ENA | VIP1_CHR_DS_1_UV_ERR_INT_ENA | NF_CHR_DS_UV_ERR_INT_ENA | COMP_ERR_INT_ENA | GRPX3_INT_ENA | GRPX2_INT_ENA | GRPX1_INT_ENA | DEI_ERROR_INT_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT0_CLIENT_ENA | VPDMA_INT0_CHANNEL_GROUP6_ENA | VPDMA_INT0_CHANNEL_GROUP5_ENA | VPDMA_INT0_CHANNEL_GROUP4_ENA | VPDMA_INT0_CHANNEL_GROUP3_ENA | VPDMA_INT0_CHANNEL_GROUP2_ENA | VPDMA_INT0_CHANNEL_GROUP1_ENA | VPDMA_INT0_CHANNEL_GROUP0_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-171. intc_intr0_status_ena1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enabled Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA | R/W | 0h | COMP Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 19 | GRPX3_INT_ENA | R/W | 0h | GRPX3 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 18 | GRPX2_INT_ENA | R/W | 0h | GRPX2 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 17 | GRPX1_INT_ENA | R/W | 0h | GRPX1 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA | R/W | 0h | DEI Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-171. intc_intr0_status_ena1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT0_CLIENT_ENA | R/W | 0h | VPDMA INT0 Client Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 6 | VPDMA_INT0_CHANNEL_GROUP6_ENA | R/W | 0h | VPDMA INT0 Channel Group6 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT0_CHANNEL_GROUP5_ENA | R/W | 0h | VPDMA INT0 Channel Group5 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT0_CHANNEL_GROUP4_ENA | R/W | 0h | VPDMA INT0 Channel Group4 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT0_CHANNEL_GROUP3_ENA | R/W | 0h | VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT0_CHANNEL_GROUP2_ENA | R/W | 0h | VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT0_CHANNEL_GROUP1_ENA | R/W | 0h | VPDMA INT0 Channel Group1 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT0_CHANNEL_GROUP0_ENA | R/W | 0h | VPDMA INT0 Channel Group0 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.7 intc_intr0_ena_set0 Register (offset = 30h) [reset = 0h]

intc_intr0_ena_set0 is shown in Figure 12-260 and described in Table 12-172.

Interrupt0 Enable/Set Register 0

**Figure 12-260. intc_intr0_ena_set0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA_SET | DVO2_INT2_ENA_SET | DVO2_INT1_ENA_SET | DVO2_INT0_ENA_SET | Reserved | | | DVO1_INT2_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA_SET | DVO1_INT0_ENA_SET | VIP2_PARSER_INT_ENA_SET | VIP1_PARSER_INT_ENA_SET | Reserved | DEI_FMD_INT_ENA_SET | Reserved | VPDMA_INT0_DESCRIPTOR_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT0_LIST7_NOTIFY_ENA_SET | VPDMA_INT0_LIST7_COMPLETE_ENA_SET | VPDMA_INT0_LIST6_NOTIFY_ENA_SET | VPDMA_INT0_LIST6_COMPLETE_ENA_SET | VPDMA_INT0_LIST5_NOTIFY_ENA_SET | VPDMA_INT0_LIST5_COMPLETE_ENA_SET | VPDMA_INT0_LIST4_NOTIFY_ENA_SET | VPDMA_INT0_LIST4_COMPLETE_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT0_LIST3_NOTIFY_ENA_SET | VPDMA_INT0_LIST3_COMPLETE_ENA_SET | VPDMA_INT0_LIST2_NOTIFY_ENA_SET | VPDMA_INT0_LIST2_COMPLETE_ENA_SET | VPDMA_INT0_LIST1_NOTIFY_ENA_SET | VPDMA_INT0_LIST1_COMPLETE_ENA_SET | VPDMA_INT0_LIST0_NOTIFY_ENA_SET | VPDMA_INT0_LIST0_COMPLETE_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-172. intc_intr0_ena_set0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA_SET | R/W | 0h | SD_VENC Interrupt Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 30 | DVO2_INT2_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 29 | DVO2_INT1_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 28 | DVO2_INT0_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 23 | DVO1_INT1_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 22 | DVO1_INT0_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA_SET | R/W | 0h | VIP2 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA_SET | R/W | 0h | VIP1 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_ENA_SET | R/W | 0h | DEI Film Mode Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-172. intc_intr0_ena_set0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | VPDMA_INT0_DESCRIPTOR_ENA_SET | R/W | 0h | VPDMA INT0 Descriptor Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT0_LIST7_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List7 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT0_LIST7_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List7 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT0_LIST6_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List6 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT0_LIST6_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List6 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT0_LIST5_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List5 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT0_LIST5_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List5 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT0_LIST4_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List4 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT0_LIST4_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List4 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT0_LIST3_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List3 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT0_LIST3_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List3 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT0_LIST2_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List2 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT0_LIST2_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List2 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT0_LIST1_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List1 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT0_LIST1_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List1 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT0_LIST0_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List0 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT0_LIST0_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List0 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.8 intc_intr0_ena_set1 Register (offset = 34h) [reset = 0h]

intc_intr0_ena_set1 is shown in Figure 12-261 and described in Table 12-173.

Interrupt0 Enable/Set Register 1

**Figure 12-261. intc_intr0_ena_set1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET | VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET | VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET | NF_CHR_DS_UV_ERR_INT_ENA_SET | COMP_ERR_INT_ENA_SET | GRPX3_INT_ENA_SET | GRPX2_INT_ENA_SET | GRPX1_INT_ENA_SET | DEI_ERROR_INT_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT0_CLIENT_ENA_SET | VPDMA_INT0_CHANNEL_GROUP6_ENA_SET | VPDMA_INT0_CHANNEL_GROUP5_ENA_SET | VPDMA_INT0_CHANNEL_GROUP4_ENA_SET | VPDMA_INT0_CHANNEL_GROUP3_ENA_SET | VPDMA_INT0_CHANNEL_GROUP2_ENA_SET | VPDMA_INT0_CHANNEL_GROUP1_ENA_SET | VPDMA_INT0_CHANNEL_GROUP0_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-173. intc_intr0_ena_set1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA_SET | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA_SET | R/W | 0h | COMP Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 19 | GRPX3_INT_ENA_SET | R/W | 0h | GRPX3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 18 | GRPX2_INT_ENA_SET | R/W | 0h | GRPX2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | GRPX1_INT_ENA_SET | R/W | 0h | GRPX1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA_SET | R/W | 0h | DEI Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT0_CLIENT_ENA_SET | R/W | 0h | VPDMA INT0 Client Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-173. intc_intr0_ena_set1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | VPDMA_INT0_CHANNEL_GROUP6_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group6 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT0_CHANNEL_GROUP5_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group5 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT0_CHANNEL_GROUP4_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group4 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT0_CHANNEL_GROUP3_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT0_CHANNEL_GROUP2_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT0_CHANNEL_GROUP1_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT0_CHANNEL_GROUP0_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group0 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.9 intc_intr0_ena_clr0 Register (offset = 38h) [reset = 0h]

intc_intr0_ena_clr0 is shown in Figure 12-262 and described in Table 12-174.

Interrupt0 Enable/Clear Register 0

**Figure 12-262. intc_intr0_ena_clr0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA_CLR | DVO2_INT2_ENA_CLR | DVO2_INT1_ENA_CLR | DVO2_INT0_ENA_CLR | Reserved | | | DVO1_INT2_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA_CLR | DVO1_INT0_ENA_CLR | VIP2_PARSER_INT_ENA_CLR | VIP1_PARSER_INT_ENA_CLR | Reserved | DEI_FMD_INT_ENA_CLR | Reserved | VPDMA_INT0_DESCRIPTOR_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT0_LIST7_NOTIFY_ENA_CLR | VPDMA_INT0_LIST7_COMPLETE_ENA_CLR | VPDMA_INT0_LIST6_NOTIFY_ENA_CLR | VPDMA_INT0_LIST6_COMPLETE_ENA_CLR | VPDMA_INT0_LIST5_NOTIFY_ENA_CLR | VPDMA_INT0_LIST5_COMPLETE_ENA_CLR | VPDMA_INT0_LIST4_NOTIFY_ENA_CLR | VPDMA_INT0_LIST4_COMPLETE_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT0_LIST3_NOTIFY_ENA_CLR | VPDMA_INT0_LIST3_COMPLETE_ENA_CLR | VPDMA_INT0_LIST2_NOTIFY_ENA_CLR | VPDMA_INT0_LIST2_COMPLETE_ENA_CLR | VPDMA_INT0_LIST1_NOTIFY_ENA_CLR | VPDMA_INT0_LIST1_COMPLETE_ENA_CLR | VPDMA_INT0_LIST0_NOTIFY_ENA_CLR | VPDMA_INT0_LIST0_COMPLETE_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-174. intc_intr0_ena_clr0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA_CLR | R/W | 0h | SD_VENC Interrupt Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 30 | DVO2_INT2_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 29 | DVO2_INT1_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 28 | DVO2_INT0_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 23 | DVO1_INT1_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 22 | DVO1_INT0_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA_CLR | R/W | 0h | VIP2 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA_CLR | R/W | 0h | VIP1 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_ENA_CLR | R/W | 0h | DEI Film Mode Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-174. intc_intr0_ena_clr0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | VPDMA_INT0_DESCRIPTOR_ENA_CLR | R/W | 0h | VPDMA INT0 Descriptor Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT0_LIST7_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List7 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT0_LIST7_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List7 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT0_LIST6_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List6 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT0_LIST6_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List6 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT0_LIST5_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List5 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT0_LIST5_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List5 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT0_LIST4_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List4 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT0_LIST4_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List4 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT0_LIST3_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List3 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT0_LIST3_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List3 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT0_LIST2_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List2 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT0_LIST2_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List2 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT0_LIST1_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List1 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT0_LIST1_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List1 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT0_LIST0_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List0 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT0_LIST0_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List0 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.10 intc_intr0_ena_clr1 Register (offset = 3Ch) [reset = 0h]

intc_intr0_ena_clr1 is shown in Figure 12-263 and described in Table 12-175.

Interrupt0 Enable/Clear Register 1

**Figure 12-263. intc_intr0_ena_clr1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR | VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR | VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR | NF_CHR_DS_UV_ERR_INT_ENA_CLR | COMP_ERR_INT_ENA_CLR | GRPX3_INT_ENA_CLR | GRPX2_INT_ENA_CLR | GRPX1_INT_ENA_CLR | DEI_ERROR_INT_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT0_CLIENT_ENA_CLR | VPDMA_INT0_CHANNEL_GROUP6_ENA_CLR | VPDMA_INT0_CHANNEL_GROUP5_ENA_CLR | VPDMA_INT0_CHANNEL_GROUP4_ENA_CLR | VPDMA_INT0_CHANNEL_GROUP3_ENA_CLR | VPDMA_INT0_CHANNEL_GROUP2_ENA_CLR | VPDMA_INT0_CHANNEL_GROUP1_ENA_CLR | VPDMA_INT0_CHANNEL_GROUP0_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-175. intc_intr0_ena_clr1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA_CLR | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA_CLR | R/W | 0h | COMP Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 19 | GRPX3_INT_ENA_CLR | R/W | 0h | GRPX3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 18 | GRPX2_INT_ENA_CLR | R/W | 0h | GRPX2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 17 | GRPX1_INT_ENA_CLR | R/W | 0h | GRPX1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA_CLR | R/W | 0h | DEI Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-175. intc_intr0_ena_clr1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT0_CLIENT_ENA_CLR | R/W | 0h | VPDMA INT0 Client Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT0_CHANNEL_GROUP6_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group6 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT0_CHANNEL_GROUP5_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group5 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT0_CHANNEL_GROUP4_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group4 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT0_CHANNEL_GROUP3_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT0_CHANNEL_GROUP2_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT0_CHANNEL_GROUP1_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT0_CHANNEL_GROUP0_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group0 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.11 intc_intr1_status_raw0 Register (offset = 40h) [reset = 0h]

intc_intr1_status_raw0 is shown in Figure 12-264 and described in Table 12-176.

Interrupt1 Raw Register 0

**Figure 12-264. intc_intr1_status_raw0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_RAW | DVO2_INT2_RAW | DVO2_INT1_RAW | DVO2_INT0_RAW | Reserved | | | DVO1_INT2_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_RAW | DVO1_INT0_RAW | VIP2_PARSER_INT_RAW | VIP1_PARSER_INT_RAW | Reserved | DEI_FMD_INT_RAW | Reserved | VPDMA_INT1_DESCRIPTOR_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT1_LIST7_NOTIFY_RAW | VPDMA_INT1_LIST7_COMPLETE_RAW | VPDMA_INT1_LIST6_NOTIFY_RAW | VPDMA_INT1_LIST6_COMPLETE_RAW | VPDMA_INT1_LIST5_NOTIFY_RAW | VPDMA_INT1_LIST5_COMPLETE_RAW | VPDMA_INT1_LIST4_NOTIFY_RAW | VPDMA_INT1_LIST4_COMPLETE_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT1_LIST3_NOTIFY_RAW | VPDMA_INT1_LIST3_COMPLETE_RAW | VPDMA_INT1_LIST2_NOTIFY_RAW | VPDMA_INT1_LIST2_COMPLETE_RAW | VPDMA_INT1_LIST1_NOTIFY_RAW | VPDMA_INT1_LIST1_COMPLETE_RAW | VPDMA_INT1_LIST0_NOTIFY_RAW | VPDMA_INT1_LIST0_COMPLETE_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-176. intc_intr1_status_raw0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_RAW | R/W | 0h | SD VENC Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 30 | DVO2_INT2_RAW | R/W | 0h | DVO2 Interrupt2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 29 | DVO2_INT1_RAW | R/W | 0h | DVO2 Interrupt1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 28 | DVO2_INT0_RAW | R/W | 0h | DVO2 Interrupt0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_RAW | R/W | 0h | DVO1 Interrupt2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 23 | DVO1_INT1_RAW | R/W | 0h | DVO1 Interrupt1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 22 | DVO1_INT0_RAW | R/W | 0h | DVO1 Interrupt0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_RAW | R/W | 0h | VIP2 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_RAW | R/W | 0h | VIP1 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_RAW | R/W | 0h | DEI Film Mode Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 17 | Reserved | R | 0h | |
| 16 | VPDMA_INT1_DESCRIPTOR_RAW | R/W | 0h | VPDMA INT0 Descriptor Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 15 | VPDMA_INT1_LIST7_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 14 | VPDMA_INT1_LIST7_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-176. intc_intr1_status_raw0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | VPDMA_INT1_LIST6_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 12 | VPDMA_INT1_LIST6_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 11 | VPDMA_INT1_LIST5_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 10 | VPDMA_INT1_LIST5_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 9 | VPDMA_INT1_LIST4_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 8 | VPDMA_INT1_LIST4_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 7 | VPDMA_INT1_LIST3_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 6 | VPDMA_INT1_LIST3_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT1_LIST2_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT1_LIST2_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT1_LIST1_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT1_LIST1_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT1_LIST0_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List0 Notify Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT1_LIST0_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List0 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.12 intc_intr1_status_raw1 Register (offset = 44h) [reset = 0h]

intc_intr1_status_raw1 is shown in Figure 12-265 and described in Table 12-177.

Interrupt1 Raw Register 1

**Figure 12-265. intc_intr1_status_raw1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_RAW | VIP2_CHR_DS_1_UV_ERR_INT_RAW |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_RAW | VIP1_CHR_DS_1_UV_ERR_INT_RAW | NF_CHR_DS_UV_ERR_INT_RAW | COMP_ERR_INT_RAW | GRPX3_INT_RAW | GRPX2_INT_RAW | GRPX1_INT_RAW | DEI_ERROR_INT_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT1_CLIENT_RAW | VPDMA_INT1_CHANNEL_GROUP6_RAW | VPDMA_INT1_CHANNEL_GROUP5_RAW | VPDMA_INT1_CHANNEL_GROUP4_RAW | VPDMA_INT1_CHANNEL_GROUP3_RAW | VPDMA_INT1_CHANNEL_GROUP2_RAW | VPDMA_INT1_CHANNEL_GROUP1_RAW | VPDMA_INT1_CHANNEL_GROUP0_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-177. intc_intr1_status_raw1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_RAW | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_RAW | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_RAW | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_RAW | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_RAW | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 20 | COMP_ERR_INT_RAW | R/W | 0h | COMP Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 19 | GRPX3_INT_RAW | R/W | 0h | GRPX3 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 18 | GRPX2_INT_RAW | R/W | 0h | GRPX2 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 17 | GRPX1_INT_RAW | R/W | 0h | GRPX1 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 16 | DEI_ERROR_INT_RAW | R/W | 0h | DEI Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT1_CLIENT_RAW | R/W | 0h | VPDMA INT0 Client Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-177. intc_intr1_status_raw1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | VPDMA_INT1_CHANNEL_GROUP6_RAW | R/W | 0h | VPDMA INT0 Channel Group6 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT1_CHANNEL_GROUP5_RAW | R/W | 0h | VPDMA INT0 Channel Group5 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT1_CHANNEL_GROUP4_RAW | R/W | 0h | VPDMA INT0 Channel Group4 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT1_CHANNEL_GROUP3_RAW | R/W | 0h | VPDMA INT0 Channel Group3 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT1_CHANNEL_GROUP2_RAW | R/W | 0h | VPDMA INT0 Channel Group2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT1_CHANNEL_GROUP1_RAW | R/W | 0h | VPDMA INT0 Channel Group1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT1_CHANNEL_GROUP0_RAW | R/W | 0h | VPDMA INT0 Channel Group0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.13 intc_intr1_status_ena0 Register (offset = 48h) [reset = 0h]

intc_intr1_status_ena0 is shown in Figure 12-266 and described in Table 12-178.

Interrupt1 Enabled Register 0

**Figure 12-266. intc_intr1_status_ena0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA | DVO2_INT2_ENA | DVO2_INT1_ENA | DVO2_INT0_ENA | Reserved | | | DVO1_INT2_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA | DVO1_INT0_ENA | VIP2_PARSER_INT_ENA | VIP1_PARSER_INT_ENA | Reserved | DEI_FMD_INT_ENA | Reserved | VPDMA_INT1_DESCRIPTOR_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT1_LIST7_NOTIFY_ENA | VPDMA_INT1_LIST7_COMPLETE_ENA | VPDMA_INT1_LIST6_NOTIFY_ENA | VPDMA_INT1_LIST6_COMPLETE_ENA | VPDMA_INT1_LIST5_NOTIFY_ENA | VPDMA_INT1_LIST5_COMPLETE_ENA | VPDMA_INT1_LIST4_NOTIFY_ENA | VPDMA_INT1_LIST4_COMPLETE_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT1_LIST3_NOTIFY_ENA | VPDMA_INT1_LIST3_COMPLETE_ENA | VPDMA_INT1_LIST2_NOTIFY_ENA | VPDMA_INT1_LIST2_COMPLETE_ENA | VPDMA_INT1_LIST1_NOTIFY_ENA | VPDMA_INT1_LIST1_COMPLETE_ENA | VPDMA_INT1_LIST0_NOTIFY_ENA | VPDMA_INT1_LIST0_COMPLETE_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-178. intc_intr1_status_ena0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA | R/W | 0h | SD_VENC Interrupt Enabled Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 30 | DVO2_INT2_ENA | R/W | 0h | DVO2 Enabled Interrupt2 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 29 | DVO2_INT1_ENA | R/W | 0h | DVO2 Enabled Interrupt1 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 28 | DVO2_INT0_ENA | R/W | 0h | DVO2 Enabled Interrupt0 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA | R/W | 0h | DVO1 Enabled Interrupt2 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 23 | DVO1_INT1_ENA | R/W | 0h | DVO1 Enabled Interrupt1 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 22 | DVO1_INT0_ENA | R/W | 0h | DVO1 Enabled Interrupt0 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA | R/W | 0h | VIP2 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA | R/W | 0h | VIP1 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 19 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-178. intc_intr1_status_ena0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | DEI_FMD_INT_ENA | R/W | 0h | DEI Film Mode Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |
| 16 | VPDMA_INT1_DESCRIPTOR_ENA | R/W | 0h | VPDMA INT0 Descriptor Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT1_LIST7_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List7 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT1_LIST7_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List7 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT1_LIST6_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List6 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT1_LIST6_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List6 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT1_LIST5_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List5 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT1_LIST5_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List5 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT1_LIST4_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List4 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT1_LIST4_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List4 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT1_LIST3_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List3 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT1_LIST3_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List3 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT1_LIST2_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List2 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT1_LIST2_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List2 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT1_LIST1_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List1 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT1_LIST1_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List1 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT1_LIST0_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List0 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT1_LIST0_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List0 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.14 intc_intr1_status_ena1 Register (offset = 4Ch) [reset = 0h]

intc_intr1_status_ena1 is shown in Figure 12-267 and described in Table 12-179.

Interrupt1 Enabled Register 1

**Figure 12-267. intc_intr1_status_ena1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_ENA | VIP2_CHR_DS_1_UV_ERR_INT_ENA |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| VIP1_CHR_DS_2_UV_ERR_INT_ENA | VIP1_CHR_DS_1_UV_ERR_INT_ENA | NF_CHR_DS_UV_ERR_INT_ENA | COMP_ERR_INT_ENA | GRPX3_INT_ENA | GRPX2_INT_ENA | GRPX1_INT_ENA | DEI_ERROR_INT_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT1_CLIENT_ENA | VPDMA_INT1_CHANNEL_GROUP6_ENA | VPDMA_INT1_CHANNEL_GROUP5_ENA | VPDMA_INT1_CHANNEL_GROUP4_ENA | VPDMA_INT1_CHANNEL_GROUP3_ENA | VPDMA_INT1_CHANNEL_GROUP2_ENA | VPDMA_INT1_CHANNEL_GROUP1_ENA | VPDMA_INT1_CHANNEL_GROUP0_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-179. intc_intr1_status_ena1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enabled Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA | R/W | 0h | COMP Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 19 | GRPX3_INT_ENA | R/W | 0h | GRPX3 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 18 | GRPX2_INT_ENA | R/W | 0h | GRPX2 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 17 | GRPX1_INT_ENA | R/W | 0h | GRPX1 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA | R/W | 0h | DEI Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-179. intc_intr1_status_ena1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT1_CLIENT_ENA | R/W | 0h | VPDMA INT0 Client Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 6 | VPDMA_INT1_CHANNEL_GROUP6_ENA | R/W | 0h | VPDMA INT0 Channel Group6 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT1_CHANNEL_GROUP5_ENA | R/W | 0h | VPDMA INT0 Channel Group5 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT1_CHANNEL_GROUP4_ENA | R/W | 0h | VPDMA INT0 Channel Group4 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT1_CHANNEL_GROUP3_ENA | R/W | 0h | VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT1_CHANNEL_GROUP2_ENA | R/W | 0h | VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT1_CHANNEL_GROUP1_ENA | R/W | 0h | VPDMA INT0 Channel Group1 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT1_CHANNEL_GROUP0_ENA | R/W | 0h | VPDMA INT0 Channel Group0 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.15 intc_intr1_ena_set0 Register (offset = 50h) [reset = 0h]

intc_intr1_ena_set0 is shown in Figure 12-268 and described in Table 12-180.

Interrupt1 Enable/Set Register 0

**Figure 12-268. intc_intr1_ena_set0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA_SET | DVO2_INT2_ENA_SET | DVO2_INT1_ENA_SET | DVO2_INT0_ENA_SET | Reserved | | | DVO1_INT2_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA_SET | DVO1_INT0_ENA_SET | VIP2_PARSER_INT_ENA_SET | VIP1_PARSER_INT_ENA_SET | Reserved | DEI_FMD_INT_ENA_SET | Reserved | VPDMA_INT1_DESCRIPTOR_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT1_LIST7_NOTIFY_ENA_SET | VPDMA_INT1_LIST7_COMPLETE_ENA_SET | VPDMA_INT1_LIST6_NOTIFY_ENA_SET | VPDMA_INT1_LIST6_COMPLETE_ENA_SET | VPDMA_INT1_LIST5_NOTIFY_ENA_SET | VPDMA_INT1_LIST5_COMPLETE_ENA_SET | VPDMA_INT1_LIST4_NOTIFY_ENA_SET | VPDMA_INT1_LIST4_COMPLETE_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT1_LIST3_NOTIFY_ENA_SET | VPDMA_INT1_LIST3_COMPLETE_ENA_SET | VPDMA_INT1_LIST2_NOTIFY_ENA_SET | VPDMA_INT1_LIST2_COMPLETE_ENA_SET | VPDMA_INT1_LIST1_NOTIFY_ENA_SET | VPDMA_INT1_LIST1_COMPLETE_ENA_SET | VPDMA_INT1_LIST0_NOTIFY_ENA_SET | VPDMA_INT1_LIST0_COMPLETE_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-180. intc_intr1_ena_set0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA_SET | R/W | 0h | SD_VENC Interrupt Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 30 | DVO2_INT2_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 29 | DVO2_INT1_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 28 | DVO2_INT0_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 23 | DVO1_INT1_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 22 | DVO1_INT0_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA_SET | R/W | 0h | VIP2 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA_SET | R/W | 0h | VIP1 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_ENA_SET | R/W | 0h | DEI Film Mode Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-180. intc_intr1_ena_set0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | VPDMA_INT1_DESCRIPTOR_ENA_SET | R/W | 0h | VPDMA INT0 Descriptor Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT1_LIST7_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List7 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT1_LIST7_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List7 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT1_LIST6_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List6 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT1_LIST6_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List6 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT1_LIST5_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List5 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT1_LIST5_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List5 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT1_LIST4_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List4 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT1_LIST4_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List4 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT1_LIST3_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List3 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT1_LIST3_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List3 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT1_LIST2_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List2 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT1_LIST2_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List2 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT1_LIST1_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List1 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT1_LIST1_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List1 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT1_LIST0_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List0 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT1_LIST0_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List0 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.16 intc_intr1_ena_set1 Register (offset = 54h) [reset = 0h]

intc_intr1_ena_set1 is shown in Figure 12-269 and described in Table 12-181.

Interrupt1 Enable/Set Register 1

**Figure 12-269. intc_intr1_ena_set1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET | VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET | VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET | NF_CHR_DS_UV_ERR_INT_ENA_SET | COMP_ERR_INT_ENA_SET | GRPX3_INT_ENA_SET | GRPX2_INT_ENA_SET | GRPX1_INT_ENA_SET | DEI_ERROR_INT_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT1_CLIENT_ENA_SET | VPDMA_INT1_CHANNEL_GROUP6_ENA_SET | VPDMA_INT1_CHANNEL_GROUP5_ENA_SET | VPDMA_INT1_CHANNEL_GROUP4_ENA_SET | VPDMA_INT1_CHANNEL_GROUP3_ENA_SET | VPDMA_INT1_CHANNEL_GROUP2_ENA_SET | VPDMA_INT1_CHANNEL_GROUP1_ENA_SET | VPDMA_INT1_CHANNEL_GROUP0_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-181. intc_intr1_ena_set1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA_SET | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA_SET | R/W | 0h | COMP Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 19 | GRPX3_INT_ENA_SET | R/W | 0h | GRPX3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 18 | GRPX2_INT_ENA_SET | R/W | 0h | GRPX2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | GRPX1_INT_ENA_SET | R/W | 0h | GRPX1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA_SET | R/W | 0h | DEI Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT1_CLIENT_ENA_SET | R/W | 0h | VPDMA INT0 Client Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-181. intc_intr1_ena_set1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | VPDMA_INT1_CHANNEL_GROUP6_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group6 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT1_CHANNEL_GROUP5_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group5 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT1_CHANNEL_GROUP4_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group4 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT1_CHANNEL_GROUP3_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT1_CHANNEL_GROUP2_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT1_CHANNEL_GROUP1_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT1_CHANNEL_GROUP0_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group0 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.17 intc_intr1_ena_clr0 Register (offset = 58h) [reset = 0h]

intc_intr1_ena_clr0 is shown in Figure 12-270 and described in Table 12-182.

Interrupt1 Enable/Clear Register 0

**Figure 12-270. intc_intr1_ena_clr0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA_CLR | DVO2_INT2_ENA_CLR | DVO2_INT1_ENA_CLR | DVO2_INT0_ENA_CLR | Reserved | | | DVO1_INT2_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA_CLR | DVO1_INT0_ENA_CLR | VIP2_PARSER_INT_ENA_CLR | VIP1_PARSER_INT_ENA_CLR | Reserved | DEI_FMD_INT_ENA_CLR | Reserved | VPDMA_INT1_DESCRIPTOR_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT1_LIST7_NOTIFY_ENA_CLR | VPDMA_INT1_LIST7_COMPLETE_ENA_CLR | VPDMA_INT1_LIST6_NOTIFY_ENA_CLR | VPDMA_INT1_LIST6_COMPLETE_ENA_CLR | VPDMA_INT1_LIST5_NOTIFY_ENA_CLR | VPDMA_INT1_LIST5_COMPLETE_ENA_CLR | VPDMA_INT1_LIST4_NOTIFY_ENA_CLR | VPDMA_INT1_LIST4_COMPLETE_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT1_LIST3_NOTIFY_ENA_CLR | VPDMA_INT1_LIST3_COMPLETE_ENA_CLR | VPDMA_INT1_LIST2_NOTIFY_ENA_CLR | VPDMA_INT1_LIST2_COMPLETE_ENA_CLR | VPDMA_INT1_LIST1_NOTIFY_ENA_CLR | VPDMA_INT1_LIST1_COMPLETE_ENA_CLR | VPDMA_INT1_LIST0_NOTIFY_ENA_CLR | VPDMA_INT1_LIST0_COMPLETE_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-182. intc_intr1_ena_clr0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA_CLR | R/W | 0h | SD_VENC Interrupt Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 30 | DVO2_INT2_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 29 | DVO2_INT1_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 28 | DVO2_INT0_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 23 | DVO1_INT1_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 22 | DVO1_INT0_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA_CLR | R/W | 0h | VIP2 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA_CLR | R/W | 0h | VIP1 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_ENA_CLR | R/W | 0h | DEI Film Mode Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-182. intc_intr1_ena_clr0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | VPDMA_INT1_DESCRIPTOR_ENA_CLR | R/W | 0h | VPDMA INT0 Descriptor Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT1_LIST7_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List7 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT1_LIST7_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List7 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT1_LIST6_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List6 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT1_LIST6_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List6 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT1_LIST5_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List5 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT1_LIST5_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List5 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT1_LIST4_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List4 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT1_LIST4_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List4 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT1_LIST3_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List3 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT1_LIST3_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List3 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT1_LIST2_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List2 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT1_LIST2_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List2 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT1_LIST1_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List1 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT1_LIST1_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List1 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT1_LIST0_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List0 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT1_LIST0_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List0 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.18 intc_intr1_ena_clr1 Register (offset = 5Ch) [reset = 0h]

intc_intr1_ena_clr1 is shown in Figure 12-271 and described in Table 12-183.

Interrupt1 Enable/Clear Register 1

**Figure 12-271. intc_intr1_ena_clr1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR | VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR | VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR | NF_CHR_DS_UV_ERR_INT_ENA_CLR | COMP_ERR_INT_ENA_CLR | GRPX3_INT_ENA_CLR | GRPX2_INT_ENA_CLR | GRPX1_INT_ENA_CLR | DEI_ERROR_INT_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT1_CLIENT_ENA_CLR | VPDMA_INT1_CHANNEL_GROUP6_ENA_CLR | VPDMA_INT1_CHANNEL_GROUP5_ENA_CLR | VPDMA_INT1_CHANNEL_GROUP4_ENA_CLR | VPDMA_INT1_CHANNEL_GROUP3_ENA_CLR | VPDMA_INT1_CHANNEL_GROUP2_ENA_CLR | VPDMA_INT1_CHANNEL_GROUP1_ENA_CLR | VPDMA_INT1_CHANNEL_GROUP0_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-183. intc_intr1_ena_clr1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA_CLR | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA_CLR | R/W | 0h | COMP Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 19 | GRPX3_INT_ENA_CLR | R/W | 0h | GRPX3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 18 | GRPX2_INT_ENA_CLR | R/W | 0h | GRPX2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 17 | GRPX1_INT_ENA_CLR | R/W | 0h | GRPX1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA_CLR | R/W | 0h | DEI Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-183. intc_intr1_ena_clr1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT1_CLIENT_ENA_CLR | R/W | 0h | VPDMA INT0 Client Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT1_CHANNEL_GROUP6_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group6 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT1_CHANNEL_GROUP5_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group5 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT1_CHANNEL_GROUP4_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group4 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT1_CHANNEL_GROUP3_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT1_CHANNEL_GROUP2_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT1_CHANNEL_GROUP1_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT1_CHANNEL_GROUP0_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group0 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.19 intc_intr2_status_raw0 Register (offset = 60h) [reset = 0h]

intc_intr2_status_raw0 is shown in Figure 12-272 and described in Table 12-184.

Interrupt2 Raw Register 0

**Figure 12-272. intc_intr2_status_raw0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_RAW | DVO2_INT2_RAW | DVO2_INT1_RAW | DVO2_INT0_RAW | Reserved | | | DVO1_INT2_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_RAW | DVO1_INT0_RAW | VIP2_PARSER_INT_RAW | VIP1_PARSER_INT_RAW | Reserved | DEI_FMD_INT_RAW | Reserved | VPDMA_INT2_DESCRIPTOR_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT2_LIST7_NOTIFY_RAW | VPDMA_INT2_LIST7_COMPLETE_RAW | VPDMA_INT2_LIST6_NOTIFY_RAW | VPDMA_INT2_LIST6_COMPLETE_RAW | VPDMA_INT2_LIST5_NOTIFY_RAW | VPDMA_INT2_LIST5_COMPLETE_RAW | VPDMA_INT2_LIST4_NOTIFY_RAW | VPDMA_INT2_LIST4_COMPLETE_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT2_LIST3_NOTIFY_RAW | VPDMA_INT2_LIST3_COMPLETE_RAW | VPDMA_INT2_LIST2_NOTIFY_RAW | VPDMA_INT2_LIST2_COMPLETE_RAW | VPDMA_INT2_LIST1_NOTIFY_RAW | VPDMA_INT2_LIST1_COMPLETE_RAW | VPDMA_INT2_LIST0_NOTIFY_RAW | VPDMA_INT2_LIST0_COMPLETE_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-184. intc_intr2_status_raw0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_RAW | R/W | 0h | SD VENC Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 30 | DVO2_INT2_RAW | R/W | 0h | DVO2 Interrupt2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 29 | DVO2_INT1_RAW | R/W | 0h | DVO2 Interrupt1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 28 | DVO2_INT0_RAW | R/W | 0h | DVO2 Interrupt0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_RAW | R/W | 0h | DVO1 Interrupt2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 23 | DVO1_INT1_RAW | R/W | 0h | DVO1 Interrupt1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 22 | DVO1_INT0_RAW | R/W | 0h | DVO1 Interrupt0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_RAW | R/W | 0h | VIP2 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_RAW | R/W | 0h | VIP1 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_RAW | R/W | 0h | DEI Film Mode Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 17 | Reserved | R | 0h | |
| 16 | VPDMA_INT2_DESCRIPTOR_RAW | R/W | 0h | VPDMA INT0 Descriptor Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 15 | VPDMA_INT2_LIST7_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 14 | VPDMA_INT2_LIST7_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-184. intc_intr2_status_raw0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | VPDMA_INT2_LIST6_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 12 | VPDMA_INT2_LIST6_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 11 | VPDMA_INT2_LIST5_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 10 | VPDMA_INT2_LIST5_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 9 | VPDMA_INT2_LIST4_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 8 | VPDMA_INT2_LIST4_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 7 | VPDMA_INT2_LIST3_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 6 | VPDMA_INT2_LIST3_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT2_LIST2_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT2_LIST2_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT2_LIST1_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT2_LIST1_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT2_LIST0_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List0 Notify Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT2_LIST0_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List0 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.20 intc_intr2_status_raw1 Register (offset = 64h) [reset = 0h]

intc_intr2_status_raw1 is shown in Figure 12-273 and described in Table 12-185.

Interrupt2 Raw Register 1

**Figure 12-273. intc_intr2_status_raw1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_RAW | VIP2_CHR_DS_1_UV_ERR_INT_RAW |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_RAW | VIP1_CHR_DS_1_UV_ERR_INT_RAW | NF_CHR_DS_UV_ERR_INT_RAW | COMP_ERR_INT_RAW | GRPX3_INT_RAW | GRPX2_INT_RAW | GRPX1_INT_RAW | DEI_ERROR_INT_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT2_CLIENT_RAW | VPDMA_INT2_CHANNEL_GROUP6_RAW | VPDMA_INT2_CHANNEL_GROUP5_RAW | VPDMA_INT2_CHANNEL_GROUP4_RAW | VPDMA_INT2_CHANNEL_GROUP3_RAW | VPDMA_INT2_CHANNEL_GROUP2_RAW | VPDMA_INT2_CHANNEL_GROUP1_RAW | VPDMA_INT2_CHANNEL_GROUP0_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-185. intc_intr2_status_raw1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_RAW | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_RAW | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_RAW | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_RAW | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_RAW | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 20 | COMP_ERR_INT_RAW | R/W | 0h | COMP Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 19 | GRPX3_INT_RAW | R/W | 0h | GRPX3 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 18 | GRPX2_INT_RAW | R/W | 0h | GRPX2 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 17 | GRPX1_INT_RAW | R/W | 0h | GRPX1 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 16 | DEI_ERROR_INT_RAW | R/W | 0h | DEI Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT2_CLIENT_RAW | R/W | 0h | VPDMA INT0 Client Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-185. intc_intr2_status_raw1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | VPDMA_INT2_CHANNEL_GROUP6_RAW | R/W | 0h | VPDMA INT0 Channel Group6 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT2_CHANNEL_GROUP5_RAW | R/W | 0h | VPDMA INT0 Channel Group5 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT2_CHANNEL_GROUP4_RAW | R/W | 0h | VPDMA INT0 Channel Group4 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT2_CHANNEL_GROUP3_RAW | R/W | 0h | VPDMA INT0 Channel Group3 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT2_CHANNEL_GROUP2_RAW | R/W | 0h | VPDMA INT0 Channel Group2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT2_CHANNEL_GROUP1_RAW | R/W | 0h | VPDMA INT0 Channel Group1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT2_CHANNEL_GROUP0_RAW | R/W | 0h | VPDMA INT0 Channel Group0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.21 intc_intr2_status_ena0 Register (offset = 68h) [reset = 0h]

intc_intr2_status_ena0 is shown in Figure 12-274 and described in Table 12-186.

Interrupt2 Enabled Register 0

**Figure 12-274. intc_intr2_status_ena0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA | DVO2_INT2_ENA | DVO2_INT1_ENA | DVO2_INT0_ENA | Reserved | | | DVO1_INT2_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA | DVO1_INT0_ENA | VIP2_PARSER_INT_ENA | VIP1_PARSER_INT_ENA | Reserved | DEI_FMD_INT_ENA | Reserved | VPDMA_INT2_DESCRIPTOR_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT2_LIST7_NOTIFY_ENA | VPDMA_INT2_LIST7_COMPLETE_ENA | VPDMA_INT2_LIST6_NOTIFY_ENA | VPDMA_INT2_LIST6_COMPLETE_ENA | VPDMA_INT2_LIST5_NOTIFY_ENA | VPDMA_INT2_LIST5_COMPLETE_ENA | VPDMA_INT2_LIST4_NOTIFY_ENA | VPDMA_INT2_LIST4_COMPLETE_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT2_LIST3_NOTIFY_ENA | VPDMA_INT2_LIST3_COMPLETE_ENA | VPDMA_INT2_LIST2_NOTIFY_ENA | VPDMA_INT2_LIST2_COMPLETE_ENA | VPDMA_INT2_LIST1_NOTIFY_ENA | VPDMA_INT2_LIST1_COMPLETE_ENA | VPDMA_INT2_LIST0_NOTIFY_ENA | VPDMA_INT2_LIST0_COMPLETE_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-186. intc_intr2_status_ena0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA | R/W | 0h | SD_VENC Interrupt Enabled Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 30 | DVO2_INT2_ENA | R/W | 0h | DVO2 Enabled Interrupt2 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 29 | DVO2_INT1_ENA | R/W | 0h | DVO2 Enabled Interrupt1 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 28 | DVO2_INT0_ENA | R/W | 0h | DVO2 Enabled Interrupt0 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA | R/W | 0h | DVO1 Enabled Interrupt2 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 23 | DVO1_INT1_ENA | R/W | 0h | DVO1 Enabled Interrupt1 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 22 | DVO1_INT0_ENA | R/W | 0h | DVO1 Enabled Interrupt0 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA | R/W | 0h | VIP2 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA | R/W | 0h | VIP1 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 19 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-186. intc_intr2_status_ena0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | DEI_FMD_INT_ENA | R/W | 0h | DEI Film Mode Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |
| 16 | VPDMA_INT2_DESCRIPTOR_ENA | R/W | 0h | VPDMA INT0 Descriptor Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT2_LIST7_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List7 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT2_LIST7_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List7 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT2_LIST6_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List6 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT2_LIST6_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List6 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT2_LIST5_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List5 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT2_LIST5_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List5 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT2_LIST4_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List4 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT2_LIST4_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List4 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT2_LIST3_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List3 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT2_LIST3_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List3 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT2_LIST2_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List2 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT2_LIST2_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List2 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT2_LIST1_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List1 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT2_LIST1_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List1 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT2_LIST0_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List0 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT2_LIST0_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List0 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |

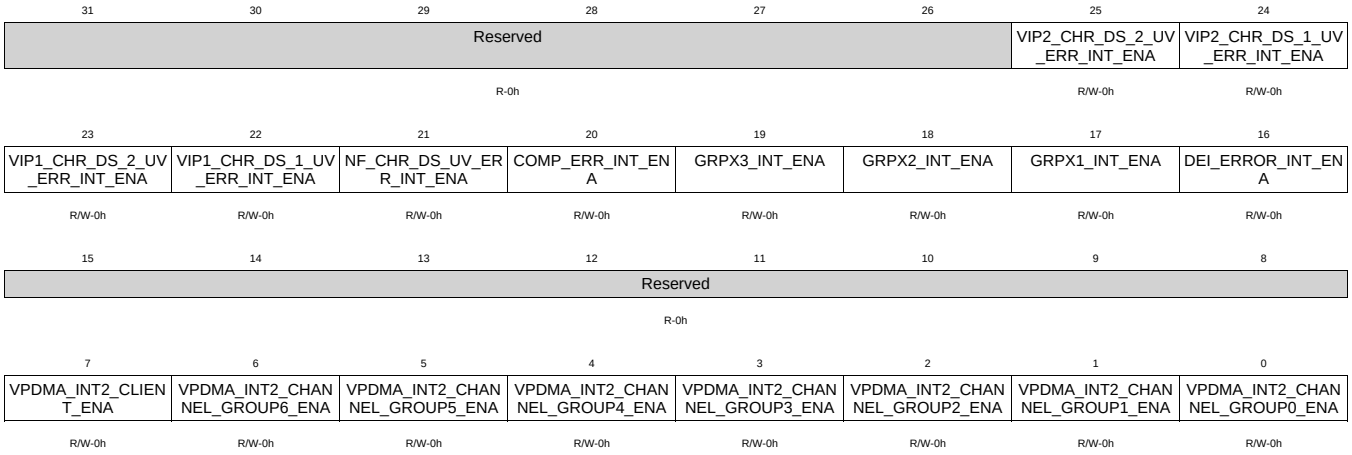

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.22 intc_intr2_status_ena1 Register (offset = 6Ch) [reset = 0h]

intc_intr2_status_ena1 is shown in Figure 12-275 and described in Table 12-187.

Interrupt2 Enabled Register 1

**Figure 12-275. intc_intr2_status_ena1 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-187. intc_intr2_status_ena1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enabled Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA | R/W | 0h | COMP Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 19 | GRPX3_INT_ENA | R/W | 0h | GRPX3 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 18 | GRPX2_INT_ENA | R/W | 0h | GRPX2 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 17 | GRPX1_INT_ENA | R/W | 0h | GRPX1 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA | R/W | 0h | DEI Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-187. intc_intr2_status_ena1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT2_CLIENT_ENA | R/W | 0h | VPDMA INT0 Client Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 6 | VPDMA_INT2_CHANNEL_GROUP6_ENA | R/W | 0h | VPDMA INT0 Channel Group6 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT2_CHANNEL_GROUP5_ENA | R/W | 0h | VPDMA INT0 Channel Group5 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT2_CHANNEL_GROUP4_ENA | R/W | 0h | VPDMA INT0 Channel Group4 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT2_CHANNEL_GROUP3_ENA | R/W | 0h | VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT2_CHANNEL_GROUP2_ENA | R/W | 0h | VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT2_CHANNEL_GROUP1_ENA | R/W | 0h | VPDMA INT0 Channel Group1 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT2_CHANNEL_GROUP0_ENA | R/W | 0h | VPDMA INT0 Channel Group0 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.23 intc_intr2_ena_set0 Register (offset = 70h) [reset = 0h]

intc_intr2_ena_set0 is shown in Figure 12-276 and described in Table 12-188.

Interrupt2 Enable/Set Register 0

**Figure 12-276. intc_intr2_ena_set0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA_SET | DVO2_INT2_ENA_SET | DVO2_INT1_ENA_SET | DVO2_INT0_ENA_SET | Reserved | | | DVO1_INT2_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA_SET | DVO1_INT0_ENA_SET | VIP2_PARSER_INT_ENA_SET | VIP1_PARSER_INT_ENA_SET | Reserved | DEI_FMD_INT_ENA_SET | Reserved | VPDMA_INT2_DESCRIPTOR_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT2_LIST7_NOTIFY_ENA_SET | VPDMA_INT2_LIST7_COMPLETE_ENA_SET | VPDMA_INT2_LIST6_NOTIFY_ENA_SET | VPDMA_INT2_LIST6_COMPLETE_ENA_SET | VPDMA_INT2_LIST5_NOTIFY_ENA_SET | VPDMA_INT2_LIST5_COMPLETE_ENA_SET | VPDMA_INT2_LIST4_NOTIFY_ENA_SET | VPDMA_INT2_LIST4_COMPLETE_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT2_LIST3_NOTIFY_ENA_SET | VPDMA_INT2_LIST3_COMPLETE_ENA_SET | VPDMA_INT2_LIST2_NOTIFY_ENA_SET | VPDMA_INT2_LIST2_COMPLETE_ENA_SET | VPDMA_INT2_LIST1_NOTIFY_ENA_SET | VPDMA_INT2_LIST1_COMPLETE_ENA_SET | VPDMA_INT2_LIST0_NOTIFY_ENA_SET | VPDMA_INT2_LIST0_COMPLETE_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-188. intc_intr2_ena_set0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA_SET | R/W | 0h | SD_VENC Interrupt Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 30 | DVO2_INT2_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 29 | DVO2_INT1_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 28 | DVO2_INT0_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 23 | DVO1_INT1_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 22 | DVO1_INT0_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA_SET | R/W | 0h | VIP2 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA_SET | R/W | 0h | VIP1 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_ENA_SET | R/W | 0h | DEI Film Mode Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-188. intc_intr2_ena_set0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | VPDMA_INT2_DESCRIPTOR_ENA_SET | R/W | 0h | VPDMA INT0 Descriptor Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT2_LIST7_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List7 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT2_LIST7_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List7 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT2_LIST6_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List6 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT2_LIST6_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List6 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT2_LIST5_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List5 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT2_LIST5_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List5 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT2_LIST4_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List4 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT2_LIST4_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List4 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT2_LIST3_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List3 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT2_LIST3_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List3 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT2_LIST2_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List2 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT2_LIST2_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List2 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT2_LIST1_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List1 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT2_LIST1_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List1 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT2_LIST0_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List0 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT2_LIST0_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List0 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.24 intc_intr2_ena_set1 Register (offset = 74h) [reset = 0h]

intc_intr2_ena_set1 is shown in Figure 12-277 and described in Table 12-189.

Interrupt2 Enable/Set Register 1

**Figure 12-277. intc_intr2_ena_set1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET | VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET | VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET | NF_CHR_DS_UV_ERR_INT_ENA_SET | COMP_ERR_INT_ENA_SET | GRPX3_INT_ENA_SET | GRPX2_INT_ENA_SET | GRPX1_INT_ENA_SET | DEI_ERROR_INT_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT2_CLIENT_ENA_SET | VPDMA_INT2_CHANNEL_GROUP6_ENA_SET | VPDMA_INT2_CHANNEL_GROUP5_ENA_SET | VPDMA_INT2_CHANNEL_GROUP4_ENA_SET | VPDMA_INT2_CHANNEL_GROUP3_ENA_SET | VPDMA_INT2_CHANNEL_GROUP2_ENA_SET | VPDMA_INT2_CHANNEL_GROUP1_ENA_SET | VPDMA_INT2_CHANNEL_GROUP0_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-189. intc_intr2_ena_set1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA_SET | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA_SET | R/W | 0h | COMP Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 19 | GRPX3_INT_ENA_SET | R/W | 0h | GRPX3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 18 | GRPX2_INT_ENA_SET | R/W | 0h | GRPX2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | GRPX1_INT_ENA_SET | R/W | 0h | GRPX1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA_SET | R/W | 0h | DEI Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT2_CLIENT_ENA_SET | R/W | 0h | VPDMA INT0 Client Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-189. intc_intr2_ena_set1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | VPDMA_INT2_CHANNEL_GROUP6_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group6 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT2_CHANNEL_GROUP5_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group5 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT2_CHANNEL_GROUP4_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group4 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT2_CHANNEL_GROUP3_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT2_CHANNEL_GROUP2_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT2_CHANNEL_GROUP1_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT2_CHANNEL_GROUP0_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group0 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |

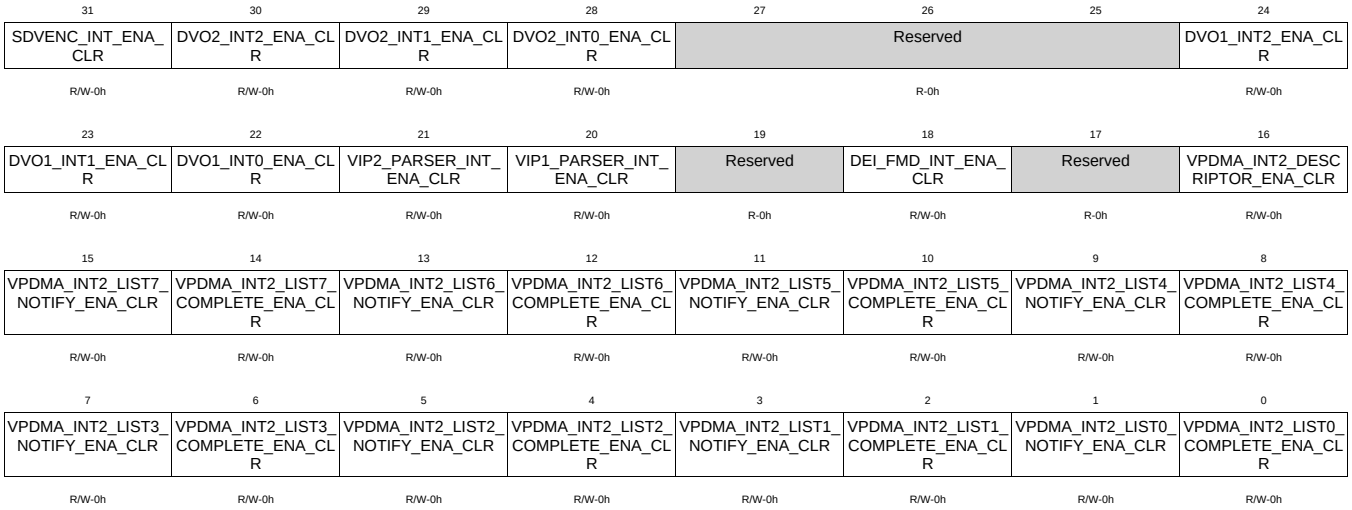

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.25 intc_intr2_ena_clr0 Register (offset = 78h) [reset = 0h]

intc_intr2_ena_clr0 is shown in Figure 12-278 and described in Table 12-190.

Interrupt2 Enable/Clear Register 0

**Figure 12-278. intc_intr2_ena_clr0 Register**



| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA_CLR | DVO2_INT2_ENA_CLR | DVO2_INT1_ENA_CLR | DVO2_INT0_ENA_CLR | Reserved | | | DVO1_INT2_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA_CLR | DVO1_INT0_ENA_CLR | VIP2_PARSER_INT_ENA_CLR | VIP1_PARSER_INT_ENA_CLR | Reserved | DEI_FMD_INT_ENA_CLR | Reserved | VPDMA_INT2_DESCRIPTOR_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT2_LIST7_NOTIFY_ENA_CLR | VPDMA_INT2_LIST7_COMPLETE_ENA_CLR | VPDMA_INT2_LIST6_NOTIFY_ENA_CLR | VPDMA_INT2_LIST6_COMPLETE_ENA_CLR | VPDMA_INT2_LIST5_NOTIFY_ENA_CLR | VPDMA_INT2_LIST5_COMPLETE_ENA_CLR | VPDMA_INT2_LIST4_NOTIFY_ENA_CLR | VPDMA_INT2_LIST4_COMPLETE_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT2_LIST3_NOTIFY_ENA_CLR | VPDMA_INT2_LIST3_COMPLETE_ENA_CLR | VPDMA_INT2_LIST2_NOTIFY_ENA_CLR | VPDMA_INT2_LIST2_COMPLETE_ENA_CLR | VPDMA_INT2_LIST1_NOTIFY_ENA_CLR | VPDMA_INT2_LIST1_COMPLETE_ENA_CLR | VPDMA_INT2_LIST0_NOTIFY_ENA_CLR | VPDMA_INT2_LIST0_COMPLETE_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-190. intc_intr2_ena_clr0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA_CLR | R/W | 0h | SD_VENC Interrupt Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 30 | DVO2_INT2_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 29 | DVO2_INT1_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 28 | DVO2_INT0_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 23 | DVO1_INT1_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 22 | DVO1_INT0_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA_CLR | R/W | 0h | VIP2 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA_CLR | R/W | 0h | VIP1 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_ENA_CLR | R/W | 0h | DEI Film Mode Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-190. intc_intr2_ena_clr0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | VPDMA_INT2_DESCRIPTOR_ENA_CLR | R/W | 0h | VPDMA INT0 Descriptor Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT2_LIST7_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List7 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT2_LIST7_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List7 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT2_LIST6_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List6 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT2_LIST6_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List6 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT2_LIST5_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List5 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT2_LIST5_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List5 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT2_LIST4_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List4 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT2_LIST4_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List4 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT2_LIST3_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List3 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT2_LIST3_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List3 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT2_LIST2_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List2 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT2_LIST2_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List2 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT2_LIST1_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List1 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT2_LIST1_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List1 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT2_LIST0_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List0 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT2_LIST0_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List0 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.26 intc_intr2_ena_clr1 Register (offset = 7Ch) [reset = 0h]

intc_intr2_ena_clr1 is shown in Figure 12-279 and described in Table 12-191.

Interrupt2 Enable/Clear Register 1

**Figure 12-279. intc_intr2_ena_clr1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR | VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR | VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR | NF_CHR_DS_UV_ERR_INT_ENA_CLR | COMP_ERR_INT_ENA_CLR | GRPX3_INT_ENA_CLR | GRPX2_INT_ENA_CLR | GRPX1_INT_ENA_CLR | DEI_ERROR_INT_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT2_CLIENT_ENA_CLR | VPDMA_INT2_CHANNEL_GROUP6_ENA_CLR | VPDMA_INT2_CHANNEL_GROUP5_ENA_CLR | VPDMA_INT2_CHANNEL_GROUP4_ENA_CLR | VPDMA_INT2_CHANNEL_GROUP3_ENA_CLR | VPDMA_INT2_CHANNEL_GROUP2_ENA_CLR | VPDMA_INT2_CHANNEL_GROUP1_ENA_CLR | VPDMA_INT2_CHANNEL_GROUP0_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-191. intc_intr2_ena_clr1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA_CLR | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA_CLR | R/W | 0h | COMP Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 19 | GRPX3_INT_ENA_CLR | R/W | 0h | GRPX3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 18 | GRPX2_INT_ENA_CLR | R/W | 0h | GRPX2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 17 | GRPX1_INT_ENA_CLR | R/W | 0h | GRPX1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA_CLR | R/W | 0h | DEI Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-191. intc_intr2_ena_clr1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT2_CLIENT_ENA_CLR | R/W | 0h | VPDMA INT0 Client Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT2_CHANNEL_GROUP6_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group6 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT2_CHANNEL_GROUP5_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group5 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT2_CHANNEL_GROUP4_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group4 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT2_CHANNEL_GROUP3_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT2_CHANNEL_GROUP2_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT2_CHANNEL_GROUP1_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT2_CHANNEL_GROUP0_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group0 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.27 intc_intr3_status_raw0 Register (offset = 80h) [reset = 0h]

intc_intr3_status_raw0 is shown in Figure 12-280 and described in Table 12-192.

Interrupt3 Raw Register 0

**Figure 12-280. intc_intr3_status_raw0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_RAW | DVO2_INT2_RAW | DVO2_INT1_RAW | DVO2_INT0_RAW | Reserved | | | DVO1_INT2_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_RAW | DVO1_INT0_RAW | VIP2_PARSER_INT_RAW | VIP1_PARSER_INT_RAW | Reserved | DEI_FMD_INT_RAW | Reserved | VPDMA_INT3_DESCRIPTOR_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT3_LIST7_NOTIFY_RAW | VPDMA_INT3_LIST7_COMPLETE_RAW | VPDMA_INT3_LIST6_NOTIFY_RAW | VPDMA_INT3_LIST6_COMPLETE_RAW | VPDMA_INT3_LIST5_NOTIFY_RAW | VPDMA_INT3_LIST5_COMPLETE_RAW | VPDMA_INT3_LIST4_NOTIFY_RAW | VPDMA_INT3_LIST4_COMPLETE_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT3_LIST3_NOTIFY_RAW | VPDMA_INT3_LIST3_COMPLETE_RAW | VPDMA_INT3_LIST2_NOTIFY_RAW | VPDMA_INT3_LIST2_COMPLETE_RAW | VPDMA_INT3_LIST1_NOTIFY_RAW | VPDMA_INT3_LIST1_COMPLETE_RAW | VPDMA_INT3_LIST0_NOTIFY_RAW | VPDMA_INT3_LIST0_COMPLETE_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-192. intc_intr3_status_raw0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_RAW | R/W | 0h | SD VENC Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 30 | DVO2_INT2_RAW | R/W | 0h | DVO2 Interrupt2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 29 | DVO2_INT1_RAW | R/W | 0h | DVO2 Interrupt1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 28 | DVO2_INT0_RAW | R/W | 0h | DVO2 Interrupt0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_RAW | R/W | 0h | DVO1 Interrupt2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 23 | DVO1_INT1_RAW | R/W | 0h | DVO1 Interrupt1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 22 | DVO1_INT0_RAW | R/W | 0h | DVO1 Interrupt0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_RAW | R/W | 0h | VIP2 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_RAW | R/W | 0h | VIP1 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_RAW | R/W | 0h | DEI Film Mode Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 17 | Reserved | R | 0h | |
| 16 | VPDMA_INT3_DESCRIPTOR_RAW | R/W | 0h | VPDMA INT0 Descriptor Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 15 | VPDMA_INT3_LIST7_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 14 | VPDMA_INT3_LIST7_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-192. intc_intr3_status_raw0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | VPDMA_INT3_LIST6_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 12 | VPDMA_INT3_LIST6_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 11 | VPDMA_INT3_LIST5_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 10 | VPDMA_INT3_LIST5_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 9 | VPDMA_INT3_LIST4_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 8 | VPDMA_INT3_LIST4_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 7 | VPDMA_INT3_LIST3_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 6 | VPDMA_INT3_LIST3_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT3_LIST2_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT3_LIST2_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT3_LIST1_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT3_LIST1_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT3_LIST0_NOTIFY_RAW | R/W | 0h | VPDMA INT0 List0 Notify Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT3_LIST0_COMPLETE_RAW | R/W | 0h | VPDMA INT0 List0 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.28 intc_intr3_status_raw1 Register (offset = 84h) [reset = 0h]

intc_intr3_status_raw1 is shown in Figure 12-281 and described in Table 12-193.

Interrupt3 Raw Register 0

**Figure 12-281. intc_intr3_status_raw1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_RAW | VIP2_CHR_DS_1_UV_ERR_INT_RAW |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_RAW | VIP1_CHR_DS_1_UV_ERR_INT_RAW | NF_CHR_DS_UV_ERR_INT_RAW | COMP_ERR_INT_RAW | GRPX3_INT_RAW | GRPX2_INT_RAW | GRPX1_INT_RAW | DEI_ERROR_INT_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT3_CLIENT_RAW | VPDMA_INT3_CHANNEL_GROUP6_RAW | VPDMA_INT3_CHANNEL_GROUP5_RAW | VPDMA_INT3_CHANNEL_GROUP4_RAW | VPDMA_INT3_CHANNEL_GROUP3_RAW | VPDMA_INT3_CHANNEL_GROUP2_RAW | VPDMA_INT3_CHANNEL_GROUP1_RAW | VPDMA_INT3_CHANNEL_GROUP0_RAW |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-193. intc_intr3_status_raw1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_RAW | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_RAW | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_RAW | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_RAW | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_RAW | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 20 | COMP_ERR_INT_RAW | R/W | 0h | COMP Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 19 | GRPX3_INT_RAW | R/W | 0h | GRPX3 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 18 | GRPX2_INT_RAW | R/W | 0h | GRPX2 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 17 | GRPX1_INT_RAW | R/W | 0h | GRPX1 Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 16 | DEI_ERROR_INT_RAW | R/W | 0h | DEI Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT3_CLIENT_RAW | R/W | 0h | VPDMA INT0 Client Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-193. intc_intr3_status_raw1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | VPDMA_INT3_CHANNEL_GROUP6_RAW | R/W | 0h | VPDMA INT0 Channel Group6 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT3_CHANNEL_GROUP5_RAW | R/W | 0h | VPDMA INT0 Channel Group5 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT3_CHANNEL_GROUP4_RAW | R/W | 0h | VPDMA INT0 Channel Group4 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT3_CHANNEL_GROUP3_RAW | R/W | 0h | VPDMA INT0 Channel Group3 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT3_CHANNEL_GROUP2_RAW | R/W | 0h | VPDMA INT0 Channel Group2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT3_CHANNEL_GROUP1_RAW | R/W | 0h | VPDMA INT0 Channel Group1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT3_CHANNEL_GROUP0_RAW | R/W | 0h | VPDMA INT0 Channel Group0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.29 intc_intr3_status_ena0 Register (offset = 88h) [reset = 0h]

intc_intr3_status_ena0 is shown in Figure 12-282 and described in Table 12-194.

Interrupt3 Enabled Register 0

**Figure 12-282. intc_intr3_status_ena0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA | DVO2_INT2_ENA | DVO2_INT1_ENA | DVO2_INT0_ENA | Reserved | | | DVO1_INT2_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA | DVO1_INT0_ENA | VIP2_PARSER_INT_ENA | VIP1_PARSER_INT_ENA | Reserved | DEI_FMD_INT_ENA | Reserved | VPDMA_INT3_DESCRIPTOR_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT3_LIST7_NOTIFY_ENA | VPDMA_INT3_LIST7_COMPLETE_ENA | VPDMA_INT3_LIST6_NOTIFY_ENA | VPDMA_INT3_LIST6_COMPLETE_ENA | VPDMA_INT3_LIST5_NOTIFY_ENA | VPDMA_INT3_LIST5_COMPLETE_ENA | VPDMA_INT3_LIST4_NOTIFY_ENA | VPDMA_INT3_LIST4_COMPLETE_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT3_LIST3_NOTIFY_ENA | VPDMA_INT3_LIST3_COMPLETE_ENA | VPDMA_INT3_LIST2_NOTIFY_ENA | VPDMA_INT3_LIST2_COMPLETE_ENA | VPDMA_INT3_LIST1_NOTIFY_ENA | VPDMA_INT3_LIST1_COMPLETE_ENA | VPDMA_INT3_LIST0_NOTIFY_ENA | VPDMA_INT3_LIST0_COMPLETE_ENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-194. intc_intr3_status_ena0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA | R/W | 0h | SD_VENC Interrupt Enabled Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 30 | DVO2_INT2_ENA | R/W | 0h | DVO2 Enabled Interrupt2 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 29 | DVO2_INT1_ENA | R/W | 0h | DVO2 Enabled Interrupt1 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 28 | DVO2_INT0_ENA | R/W | 0h | DVO2 Enabled Interrupt0 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA | R/W | 0h | DVO1 Enabled Interrupt2 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 23 | DVO1_INT1_ENA | R/W | 0h | DVO1 Enabled Interrupt1 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 22 | DVO1_INT0_ENA | R/W | 0h | DVO1 Enabled Interrupt0 Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA | R/W | 0h | VIP2 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA | R/W | 0h | VIP1 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 19 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-194. intc_intr3_status_ena0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | DEI_FMD_INT_ENA | R/W | 0h | DEI Film Mode Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |
| 16 | VPDMA_INT3_DESCRIPTOR_ENA | R/W | 0h | VPDMA INT0 Descriptor Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT3_LIST7_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List7 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT3_LIST7_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List7 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT3_LIST6_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List6 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT3_LIST6_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List6 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT3_LIST5_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List5 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT3_LIST5_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List5 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT3_LIST4_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List4 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT3_LIST4_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List4 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT3_LIST3_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List3 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT3_LIST3_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List3 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT3_LIST2_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List2 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT3_LIST2_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List2 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT3_LIST1_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List1 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT3_LIST1_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List1 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT3_LIST0_NOTIFY_ENA | R/W | 0h | VPDMA INT0 List0 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT3_LIST0_COMPLETE_ENA | R/W | 0h | VPDMA INT0 List0 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |

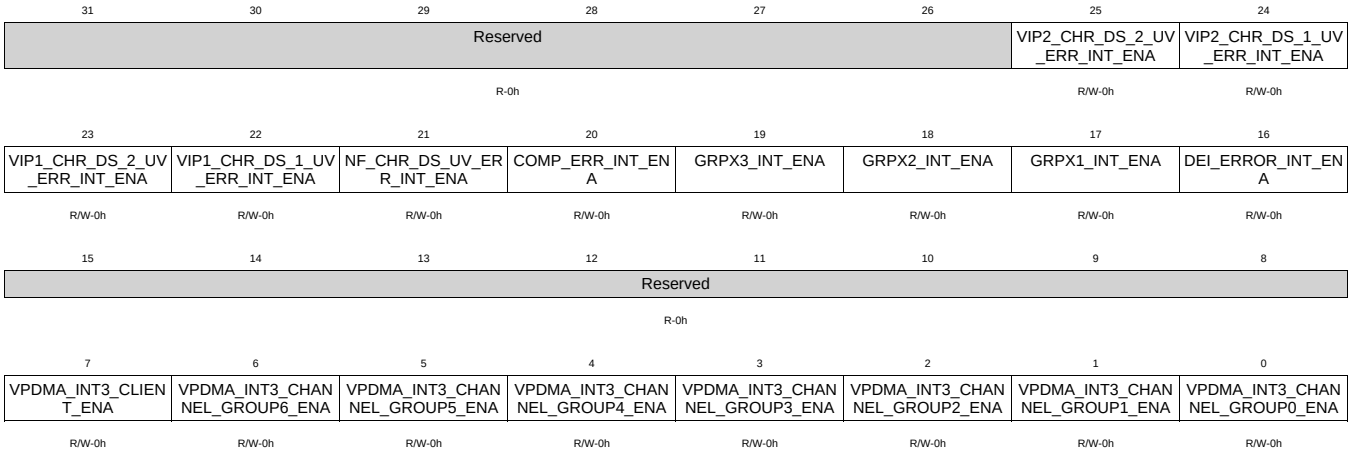

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.30 intc_intr3_status_ena1 Register (offset = 8Ch) [reset = 0h]

intc_intr3_status_ena1 is shown in Figure 12-283 and described in Table 12-195.

Interrupt3 Enabled Register 1

**Figure 12-283. intc_intr3_status_ena1 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-195. intc_intr3_status_ena1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enabled Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA | R/W | 0h | COMP Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 19 | GRPX3_INT_ENA | R/W | 0h | GRPX3 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 18 | GRPX2_INT_ENA | R/W | 0h | GRPX2 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 17 | GRPX1_INT_ENA | R/W | 0h | GRPX1 Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA | R/W | 0h | DEI Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-195. intc_intr3_status_ena1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT3_CLIENT_ENA | R/W | 0h | VPDMA INT0 Client Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 6 | VPDMA_INT3_CHANNEL_GROUP6_ENA | R/W | 0h | VPDMA INT0 Channel Group6 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 5 | VPDMA_INT3_CHANNEL_GROUP5_ENA | R/W | 0h | VPDMA INT0 Channel Group5 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 4 | VPDMA_INT3_CHANNEL_GROUP4_ENA | R/W | 0h | VPDMA INT0 Channel Group4 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 3 | VPDMA_INT3_CHANNEL_GROUP3_ENA | R/W | 0h | VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 2 | VPDMA_INT3_CHANNEL_GROUP2_ENA | R/W | 0h | VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 1 | VPDMA_INT3_CHANNEL_GROUP1_ENA | R/W | 0h | VPDMA INT0 Channel Group1 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |
| 0 | VPDMA_INT3_CHANNEL_GROUP0_ENA | R/W | 0h | VPDMA INT0 Channel Group0 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.31 intc_intr3_ena_set0 Register (offset = 90h) [reset = 0h]

intc_intr3_ena_set0 is shown in Figure 12-284 and described in Table 12-196.

Interrupt3 Enable/Set Register 0

**Figure 12-284. intc_intr3_ena_set0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SDVENC_INT_ENA_SET | DVO2_INT2_ENA_SET | DVO2_INT1_ENA_SET | DVO2_INT0_ENA_SET | Reserved | | | DVO1_INT2_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DVO1_INT1_ENA_SET | DVO1_INT0_ENA_SET | VIP2_PARSER_INT_ENA_SET | VIP1_PARSER_INT_ENA_SET | Reserved | DEI_FMD_INT_ENA_SET | Reserved | VPDMA_INT3_DESCRIPTOR_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VPDMA_INT3_LIST7_NOTIFY_ENA_SET | VPDMA_INT3_LIST7_COMPLETE_ENA_SET | VPDMA_INT3_LIST6_NOTIFY_ENA_SET | VPDMA_INT3_LIST6_COMPLETE_ENA_SET | VPDMA_INT3_LIST5_NOTIFY_ENA_SET | VPDMA_INT3_LIST5_COMPLETE_ENA_SET | VPDMA_INT3_LIST4_NOTIFY_ENA_SET | VPDMA_INT3_LIST4_COMPLETE_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VPDMA_INT3_LIST3_NOTIFY_ENA_SET | VPDMA_INT3_LIST3_COMPLETE_ENA_SET | VPDMA_INT3_LIST2_NOTIFY_ENA_SET | VPDMA_INT3_LIST2_COMPLETE_ENA_SET | VPDMA_INT3_LIST1_NOTIFY_ENA_SET | VPDMA_INT3_LIST1_COMPLETE_ENA_SET | VPDMA_INT3_LIST0_NOTIFY_ENA_SET | VPDMA_INT3_LIST0_COMPLETE_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-196. intc_intr3_ena_set0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA_SET | R/W | 0h | SD_VENC Interrupt Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 30 | DVO2_INT2_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 29 | DVO2_INT1_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 28 | DVO2_INT0_ENA_SET | R/W | 0h | DVO2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 23 | DVO1_INT1_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 22 | DVO1_INT0_ENA_SET | R/W | 0h | DVO1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA_SET | R/W | 0h | VIP2 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA_SET | R/W | 0h | VIP1 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_ENA_SET | R/W | 0h | DEI Film Mode Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-196. intc_intr3_ena_set0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | VPDMA_INT3_DESCRIPTOR_ENA_SET | R/W | 0h | VPDMA INT0 Descriptor Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT3_LIST7_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List7 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT3_LIST7_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List7 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT3_LIST6_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List6 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT3_LIST6_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List6 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT3_LIST5_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List5 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT3_LIST5_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List5 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT3_LIST4_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List4 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT3_LIST4_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List4 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT3_LIST3_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List3 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT3_LIST3_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List3 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT3_LIST2_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List2 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT3_LIST2_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List2 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT3_LIST1_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List1 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT3_LIST1_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List1 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT3_LIST0_NOTIFY_ENA_SET | R/W | 0h | VPDMA INT0 List0 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT3_LIST0_COMPLETE_ENA_SET | R/W | 0h | VPDMA INT0 List0 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.32 intc_intr3_ena_set1 Register (offset = 94h) [reset = 0h]

intc_intr3_ena_set1 is shown in Figure 12-285 and described in Table 12-197.

Interrupt3 Enable/Set Register 1

**Figure 12-285. intc_intr3_ena_set1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET | VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET | VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET | NF_CHR_DS_UV_ERR_INT_ENA_SET | COMP_ERR_INT_ENA_SET | GRPX3_INT_ENA_SET | GRPX2_INT_ENA_SET | GRPX1_INT_ENA_SET | DEI_ERROR_INT_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT3_CLIENT_ENA_SET | VPDMA_INT3_CHANNEL_GROUP6_ENA_SET | VPDMA_INT3_CHANNEL_GROUP5_ENA_SET | VPDMA_INT3_CHANNEL_GROUP4_ENA_SET | VPDMA_INT3_CHANNEL_GROUP3_ENA_SET | VPDMA_INT3_CHANNEL_GROUP2_ENA_SET | VPDMA_INT3_CHANNEL_GROUP1_ENA_SET | VPDMA_INT3_CHANNEL_GROUP0_ENA_SET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-197. intc_intr3_ena_set1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA_SET | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA_SET | R/W | 0h | COMP Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 19 | GRPX3_INT_ENA_SET | R/W | 0h | GRPX3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 18 | GRPX2_INT_ENA_SET | R/W | 0h | GRPX2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 17 | GRPX1_INT_ENA_SET | R/W | 0h | GRPX1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA_SET | R/W | 0h | DEI Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT3_CLIENT_ENA_SET | R/W | 0h | VPDMA INT0 Client Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-197. intc_intr3_ena_set1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | VPDMA_INT3_CHANNEL_GROUP6_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group6 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT3_CHANNEL_GROUP5_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group5 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT3_CHANNEL_GROUP4_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group4 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT3_CHANNEL_GROUP3_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT3_CHANNEL_GROUP2_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT3_CHANNEL_GROUP1_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT3_CHANNEL_GROUP0_ENA_SET | R/W | 0h | VPDMA INT0 Channel Group0 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect |

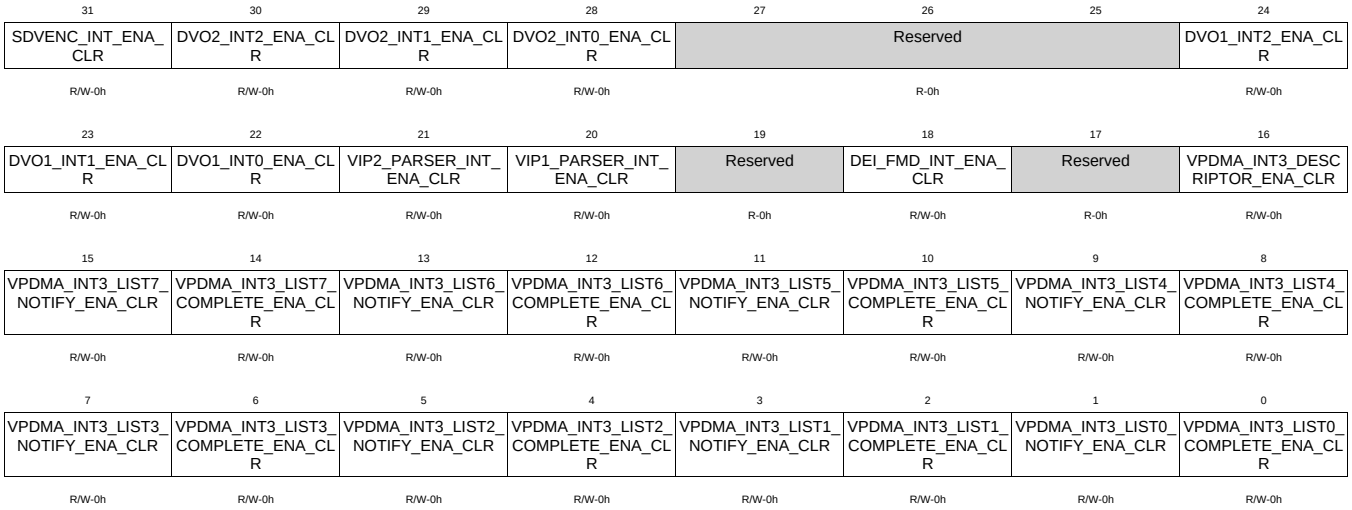

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.33 intc_intr3_ena_clr0 Register (offset = 98h) [reset = 0h]

intc_intr3_ena_clr0 is shown in Figure 12-286 and described in Table 12-198.

Interrupt3 Enable/Clear Register 0

**Figure 12-286. intc_intr3_ena_clr0 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-198. intc_intr3_ena_clr0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | SDVENC_INT_ENA_CLR | R/W | 0h | SD_VENC Interrupt Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 30 | DVO2_INT2_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 29 | DVO2_INT1_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 28 | DVO2_INT0_ENA_CLR | R/W | 0h | DVO2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 27-25 | Reserved | R | 0h | |
| 24 | DVO1_INT2_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 23 | DVO1_INT1_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 22 | DVO1_INT0_ENA_CLR | R/W | 0h | DVO1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 21 | VIP2_PARSER_INT_ENA_CLR | R/W | 0h | VIP2 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 20 | VIP1_PARSER_INT_ENA_CLR | R/W | 0h | VIP1 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 19 | Reserved | R | 0h | |
| 18 | DEI_FMD_INT_ENA_CLR | R/W | 0h | DEI Film Mode Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 17 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-198. intc_intr3_ena_clr0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | VPDMA_INT3_DESCRIPTOR_ENA_CLR | R/W | 0h | VPDMA INT0 Descriptor Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 15 | VPDMA_INT3_LIST7_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List7 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 14 | VPDMA_INT3_LIST7_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List7 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 13 | VPDMA_INT3_LIST6_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List6 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 12 | VPDMA_INT3_LIST6_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List6 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 11 | VPDMA_INT3_LIST5_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List5 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 10 | VPDMA_INT3_LIST5_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List5 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 9 | VPDMA_INT3_LIST4_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List4 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 8 | VPDMA_INT3_LIST4_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List4 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 7 | VPDMA_INT3_LIST3_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List3 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT3_LIST3_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List3 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT3_LIST2_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List2 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT3_LIST2_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List2 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT3_LIST1_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List1 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT3_LIST1_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List1 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT3_LIST0_NOTIFY_ENA_CLR | R/W | 0h | VPDMA INT0 List0 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT3_LIST0_COMPLETE_ENA_CLR | R/W | 0h | VPDMA INT0 List0 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.34 intc_intr3_ena_clr1 Register (offset = 9Ch) [reset = 0h]

intc_intr3_ena_clr1 is shown in Figure 12-287 and described in Table 12-199.

Interrupt3 Enable/Clear Register 1

**Figure 12-287. intc_intr3_ena_clr1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR | VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR | VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR | NF_CHR_DS_UV_ERR_INT_ENA_CLR | COMP_ERR_INT_ENA_CLR | GRPX3_INT_ENA_CLR | GRPX2_INT_ENA_CLR | GRPX1_INT_ENA_CLR | DEI_ERROR_INT_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPDMA_INT3_CLIENT_ENA_CLR | VPDMA_INT3_CHANNEL_GROUP6_ENA_CLR | VPDMA_INT3_CHANNEL_GROUP5_ENA_CLR | VPDMA_INT3_CHANNEL_GROUP4_ENA_CLR | VPDMA_INT3_CHANNEL_GROUP3_ENA_CLR | VPDMA_INT3_CHANNEL_GROUP2_ENA_CLR | VPDMA_INT3_CHANNEL_GROUP1_ENA_CLR | VPDMA_INT3_CHANNEL_GROUP0_ENA_CLR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-199. intc_intr3_ena_clr1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP2 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 24 | VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP2 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 23 | VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP1 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 22 | VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR | R/W | 0h | VIP1 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 21 | NF_CHR_DS_UV_ERR_INT_ENA_CLR | R/W | 0h | Noise Filter Chroma Downsampler UV error Interrupt Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 20 | COMP_ERR_INT_ENA_CLR | R/W | 0h | COMP Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 19 | GRPX3_INT_ENA_CLR | R/W | 0h | GRPX3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 18 | GRPX2_INT_ENA_CLR | R/W | 0h | GRPX2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 17 | GRPX1_INT_ENA_CLR | R/W | 0h | GRPX1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 16 | DEI_ERROR_INT_ENA_CLR | R/W | 0h | DEI Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-199. intc_intr3_ena_clr1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | Reserved | R | 0h | |
| 7 | VPDMA_INT3_CLIENT_ENA_CLR | R/W | 0h | VPDMA INT0 Client Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 6 | VPDMA_INT3_CHANNEL_GROUP6_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group6 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 5 | VPDMA_INT3_CHANNEL_GROUP5_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group5 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 4 | VPDMA_INT3_CHANNEL_GROUP4_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group4 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 3 | VPDMA_INT3_CHANNEL_GROUP3_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 2 | VPDMA_INT3_CHANNEL_GROUP2_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 1 | VPDMA_INT3_CHANNEL_GROUP1_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |
| 0 | VPDMA_INT3_CHANNEL_GROUP0_ENA_CLR | R/W | 0h | VPDMA INT0 Channel Group0 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.35 intc_eoi Register (offset = A0h) [reset = 0h]

intc_eoi is shown in Figure 12-288 and described in Table 12-200.

End of Interrupt Register

**Figure 12-288. intc_eoi Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EOI_VECTOR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-200. intc_eoi Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | EOI_VECTOR | R/W | 0h | Number associated with the ipgenericirq for intr output. There are 4 interrupt outputs Write 0x0 : Write to intr0 IP Generic Write 0x1 : Write to intr1 IP Generic Write 0x2 : Write to intr2 IP Generic Write 0x3 : Write to intr3 IP Generic Any other write value is ignored. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.36 clkc_clken Register (offset = 100h) [reset = 0h]

clkc_clken is shown in Figure 12-289 and described in Table 12-201.

Clock Enable Register

**Figure 12-289. clkc_clken Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | NF_DP_EN |
| R-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | VIP2_DP_EN | VIP1_DP_EN |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | SDVENC_EN | DVO2_EN | | HDMI_DVO1_EN | IND_TRANS2_DP_EN |
| R-0h | | | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IND_TRANS1_DP_EN | COMP_DP_EN | GRPX3_DP_EN | GRPX2_DP_EN | GRPX1_DP_EN | AUX_DP_EN | PRIM_DP_EN | VPDMA_EN |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-201. clkc_clken Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | Reserved | R | 0h | |
| 24 | NF_DP_EN | R/W | 0h | Noise Filter Data Path Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 23-18 | Reserved | R | 0h | |
| 17 | VIP2_DP_EN | R/W | 0h | Video Input Port 2 Data Path Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 16 | VIP1_DP_EN | R/W | 0h | Video Input Port 1 Data Path Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 15-13 | Reserved | R | 0h | |
| 12 | SDVENC_EN | R/W | 0h | SD VENC Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 11 | DVO2_EN | R/W | 0h | DVO2 VENC Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 10 | | R | 0h | |
| 9 | HDMI_DVO1_EN | R/W | 0h | HDMI/DVO1 VENC Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 8 | IND_TRANS2_DP_EN | R/W | 0h | Independent Transcode 2 (to VIP1) Data Path Clock .. 1 = Clock Enabled.. 0 = Clock Disabled |
| 7 | IND_TRANS1_DP_EN | R/W | 0h | Independent Transcode 1 (to VIP2) Data Path Clock .. 1 = Clock Enabled.. 0 = Clock Disabled |
| 6 | COMP_DP_EN | R/W | 0h | Compositor Data Path Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 5 | GRPX3_DP_EN | R/W | 0h | Graphics 3 Data Path Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 4 | GRPX2_DP_EN | R/W | 0h | Graphics 2 Data Path Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 3 | GRPX1_DP_EN | R/W | 0h | Graphics 1 Data Path Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 2 | AUX_DP_EN | R/W | 0h | Auxiliary Video Data Path Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-201. clkc_clken Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | PRIM_DP_EN | R/W | 0h | Primary Video Data Path Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |
| 0 | VPDMA_EN | R/W | 0h | VPDMA Clock Enable.. 1 = Clock Enabled.. 0 = Clock Disabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.37 clkc_dps Register (offset = 108h) [reset = 0h]

clkc_dps is shown in Figure 12-290 and described in Table 12-202.

Main Datapath Select Register

**Figure 12-290. clkc_dps Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MAIN_DATAPATH_SELECT | | | | Reserved | | | |
| R/W-0h | | | | R-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | NF_BYPASS_SELECT | Reserved |
| R-0h | | | | | | R/W-0h | R-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | IND_TRANS2_SELECT | IND_TRANS1_SELECT | Reserved | SC_M_WRBK_SELECT | | | SDVENC_SELECT |
| R-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | | | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SDVENC_SELECT | | | HDCOMP_DVO2_SELECT | | VCOMP_MAIN_DISABLE | VCOMP_PIP_SELECT | |
| R/W-0h | | | R/W-0h | | R/W-0h | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-202. clkc_dps Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | MAIN_DATAPATH_SELECT | R/W | 0h | Main Datapath Register Field Enable<br>0000 : All fields written<br>0001 : Only vcomp_pip_select written<br>0010 : Only vcomp_main_disable written<br>0011 : Only hdcomp_dvo2_select written<br>0100 : Only sdvenc_select written<br>0101 : Only sc_m_wrbk_select written<br>0110 : Only ind_trans1_select written<br>0111 : Only ind_trans2_select written<br>1000 : Only comp_decomp_hq_bypass written<br>1001 : Only comp_decomp_lc_bypass written<br>1010 : Only nf_bypass_select written<br>1011 : Reserved<br>1100 : Reserved<br>1101 : Reserved<br>1110 : Reserved<br>1111 : Reserved |
| 27-18 | Reserved | R | 0h | |
| 17 | NF_BYPASS_SELECT | R/W | 0h | Noise Filter Bypass.<br>0 : Noise Filter Selected<br>1 : Noise Filter Bypassed If Noise Filter is bypassed.. 422 Private Data Store data is chroma downsampled and written back out as 420 Tiled Data. |
| 16-15 | Reserved | R | 0h | |
| 14 | IND_TRANS2_SELECT | R/W | 0h | Independent Transcode Path Select<br>0 : DEI_H Path selected for VIP1 Scaler Output<br>1 : Independent Transcode Path2 selected for VIP2 Scaler Output |
| 13 | IND_TRANS1_SELECT | R/W | 0h | Independent Transcode Path Select<br>0 : DEI Path selected for VIP2 Scaler Output<br>1 : Independent Transcode Path1 selected for VIP1 Scaler Output |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-202. clkc_dps Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | Reserved | R | 0h | |
| 11-9 | SC_M_WRBK_SELECT | R/W | 0h | SC_M_WRBK Path Select<br>000 : Path Disabled<br>001 : HDMI Composited Path Input (from COMP)<br>010 : EDE Path Input (from EDE)<br>011 : VCOMP Path Input (from VCOMP)<br>100 : Auxiliary (PIP) Memory Input (from VPDMA)<br>101 : Primary (HQ) Memory Input (from VPDMA)<br>110 : Reserved<br>111 : Reserved |
| 8-6 | SDVENC_SELECT | R/W | 0h | SD VENC Path Select<br>000 : Path Disabled<br>001 : Auxiliary (PIP) Path Input (from SC_M)<br>010 : Auxiliary (PIP) Memory Input (from VPDMA)<br>011 : Primary (HQ) Memory Input (from VPDMA)<br>100 : Independent Transcode Input (from Independent Transcode)<br>101 : Reserved<br>110 : Reserved<br>111 : Reserved |
| 5-3 | HDCOMP_DVO2_SELECT | R/W | 0h | HDCOMP/DVO2 Path Input Select<br>000 : Path Disabled<br>001 : Auxiliary (PIP) Path Input (from SC_M)<br>010 : Auxiliary (PIP) Memory Input (from VPDMA)<br>011 : Primary (HQ) Memory Input (from VPDMA)<br>100 : Reserved<br>101 : Reserved<br>110 : Reserved<br>111 : Reserved |
| 2 | VCOMP_MAIN_DISABLE | R/W | 0h | VCOMP Main Input Disable (drives main VENC)<br>0 : Path Enabled<br>1 : Path Disabled If VCOMP Main Input is disabled.. the VCOMP will not receive input from the main path. Upstream paths would be allowed to run (transcode.. memory to memory). Disabling this input implies VCOMP will not be internally configured to use this input. |
| 1-0 | VCOMP_PIP_SELECT | R/W | 0h | VCOMP PIP Input Select (drives main VENC)<br>00 : Path Disabled<br>01 : Auxiliary (PIP) Path Input (from SC_M)<br>10 : Auxiliary (PIP) Memory Input (from VPDMA)<br>11 : Primary (HQ) Memory Input (from VPDMA) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.38 clkc_vip1dps Register (offset = 10Ch) [reset = 0h]

clkc_vip1dps is shown in Figure 12-291 and described in Table 12-203.

VIP1 Datapath Select Register

**Figure 12-291. clkc_vip1dps Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| VIP1_DATAPATH_SELECT | | | | Reserved | | | |
| R/W-0h | | | | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | VIP1_CHR_DS_2_BYPASS | VIP1_CHR_DS_1_BYPASS |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VIP1_MULTI_CHANNEL_SELECT | VIP1_CHR_DS_2_SRC_SELECT | | | VIP1_CHR_DS_1_SRC_SELECT | | | VIP1_RGB_OUT_HI_SELECT |
| R/W-0h | R/W-0h | | | R/W-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VIP1_RGB_OUT_LO_SELECT | VIP1_RGB_SRC_SELECT | VIP1_SC_SRC_SELECT | | | VIP1_CSC_SRC_SELECT | | |
| R/W-0h | R/W-0h | R/W-0h | | | R/W-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-203. clkc_vip1dps Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | VIP1_DATAPATH_SELECT | R/W | 0h | VIP1 Datapath Register Field Enable<br>0000 : All fields written<br>0001 : Only vip1_csc_src_select written<br>0010 : Only vip1_sc_src_select written<br>0011 : Only vip1_rgb_src_select written<br>0100 : Only vip1_rgb_out_lo_select written<br>0101 : Only vip1_rgb_out_hi_select written<br>0110 : Only vip1_chr_ds_1_src_select written<br>0111 : Only vip1_chr_ds_2_src_select written<br>1000 : Only vip1_multi_channel_select written<br>1001 : Only vip1_chr_ds_1_bypass written<br>1010 : Only vip1_chr_ds_2_bypass written<br>1011 : Reserved<br>1100 : Reserved<br>1101 : Reserved<br>1110 : Reserved<br>1111 : Reserved |
| 27-18 | Reserved | R | 0h | |
| 17 | VIP1_CHR_DS_2_BYPASS | R/W | 0h | Video Input Port 1 Chroma Downsampler 2 Bypass<br>0 : VIP Chroma Downsampler 1 selected<br>1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420 |
| 16 | VIP1_CHR_DS_1_BYPASS | R/W | 0h | Video Input Port 1 Chroma Downsampler 1 Bypass<br>0 : VIP Chroma Downsampler 1 selected<br>1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-203. clkc_vip1dps Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | VIP1_MULTI_CHANNEL_SELECT | R/W | 0h | Video Input Port 1 Multi Channel Select<br>0 : VIP_PARSER A and B channels operate in single channel mode<br>1 : VIP_PARSER A and B channels directly drive VPDMA (multi-channel case) Multi-Channel means that the A and B sources are from multiple channels and used in a multiplexed stream mode. The VIP Parser extracts the channel ID from each source and outputs this information to VPDMA.. and thus to memory. When operating in a multiplexed stream mode.. this bit must be set to 1 to enable the channel information to be passed to memory. If this is not set.. the channel number (or source number) will be 0 for all streams. If vip1_rgb_out_select = 1.. then VIP_PARSER A port is connected to VPDMA |
| 14-12 | VIP1_CHR_DS_2_SRC_SELECT | R/W | 0h | Video Input Port 1 Chroma Downsampler 2 Source Select<br>000 : Path Disabled (no input to CHR_DS)<br>001 : Source from Scaler (SC_M)<br>010 : Source from Color Space Converter (CSC)<br>011 : Source from VIP_PARSER A port<br>100 : Source from VIP_PARSER B port<br>101 : Source from Transcode (422)<br>110 : Reserved<br>111 : Reserved |
| 11-9 | VIP1_CHR_DS_1_SRC_SELECT | R/W | 0h | Video Input Port 1 Chroma Downsampler 1 Source Select<br>000 : Path Disabled (no input to CHR_DS)<br>001 : Source from Scaler (SC_M)<br>010 : Source from Color Space Converter (CSC)<br>011 : Source from VIP_PARSER A port<br>100 : Source from VIP_PARSER B port<br>101 : Source from Transcode (422)<br>110 : Reserved<br>111 : Reserved |
| 8 | VIP1_RGB_OUT_HI_SELECT | R/W | 0h | Video Input Port 1 HI RGB Output Select<br>0 : Output Type is 420/422<br>1 : Output Type is RGB |
| 7 | VIP1_RGB_OUT_LO_SELECT | R/W | 0h | Video Input Port 1 LO RGB Output Select<br>0 : Output Type is 420/422<br>1 : Output Type is RGB |
| 6 | VIP1_RGB_SRC_SELECT | R/W | 0h | Video Input Port 1 RGB Output Path Select<br>0 : Source from Compositor RGB input<br>1 : Source from CSC |
| 5-3 | VIP1_SC_SRC_SELECT | R/W | 0h | Video Input Port 1 SC_M Source Select<br>000 : Path Disabled<br>001 : Source from Color Space Converter (CSC)<br>010 : Source from VIP_PARSER A port<br>011 : Source from VIP_PARSER B port<br>100 : Source from Transcode (422)<br>101 : Reserved<br>110 : Reserved<br>111 : Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-203. clkc_vip1dps Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2-0 | VIP1_CSC_SRC_SELECT | R/W | 0h | Video Input Port 1 CSC Source Select<br>000 : Path Disabled<br>001 : Source from VIP_PARSER A (422) port<br>010 : Source from VIP_PARSER B port<br>011 : Source from Transcode (422)<br>100 : Source from VIP_PARSER A (RGB) port<br>101 : Source from Compositor (RGB)<br>110 : Reserved<br>111 : Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.39 clkc_vip2dps Register (offset = 110h) [reset = 0h]

clkc_vip2dps is shown in Figure 12-292 and described in Table 12-204.

VIP2 Datapath Select Register

**Figure 12-292. clkc_vip2dps Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| VIP2_DATAPATH_SELECT | | | | Reserved | | | |
| R/W-0h | | | | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | VIP2_CHR_DS_2_BYPASS | VIP2_CHR_DS_1_BYPASS |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VIP2_MULTI_CHANNEL_SELECT | VIP2_CHR_DS_2_SRC_SELECT | | | VIP2_CHR_DS_1_SRC_SELECT | | | VIP2_RGB_OUT_HI_SELECT |
| R/W-0h | R/W-0h | | | R/W-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VIP2_RGB_OUT_LO_SELECT | VIP2_RGB_SRC_SELECT | VIP2_SC_SRC_SELECT | | | VIP2_CSC_SRC_SELECT | | |
| R/W-0h | R/W-0h | R/W-0h | | | R/W-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-204. clkc_vip2dps Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | VIP2_DATAPATH_SELECT | R/W | 0h | VIP2 Datapath Register Field Enable<br>0000 : All fields written<br>0001 : Only vip2_csc_src_select written<br>0010 : Only vip2_sc_src_select written<br>0011 : Only vip2_rgb_src_select written<br>0100 : Only vip2_rgb_out_lo_select written<br>0101 : Only vip2_rgb_out_hi_select written<br>0110 : Only vip2_chr_ds_1_src_select written<br>0111 : Only vip2_chr_ds_2_src_select written<br>1000 : Only vip2_multi_channel_select written<br>1001 : Only vip2_chr_ds_1_bypass written<br>1010 : Only vip2_chr_ds_2_bypass written<br>1011 : Reserved<br>1100 : Reserved<br>1101 : Reserved<br>1110 : Reserved<br>1111 : Reserved |
| 27-18 | Reserved | R | 0h | |
| 17 | VIP2_CHR_DS_2_BYPASS | R/W | 0h | Video Input Port 2 Chroma Downsampler 2 Bypass<br>0 : VIP Chroma Downsampler 1 selected<br>1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420 |
| 16 | VIP2_CHR_DS_1_BYPASS | R/W | 0h | Video Input Port 2 Chroma Downsampler 1 Bypass<br>0 : VIP Chroma Downsampler 1 selected<br>1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-204. clkc_vip2dps Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | VIP2_MULTI_CHANNEL_SELECT | R/W | 0h | Video Input Port 2 Multi Channel Select<br>0 : VIP_PARSER A and B channels operate in single channel mode<br>1 : VIP_PARSER A and B channels directly drive VPDMA (multi-channel case) Multi-Channel means that the A and B sources are from multiple channels and used in a multiplexed stream mode. The VIP Parser extracts the channel ID from each source and outputs this information to VPDMA.. and thus to memory. When operating in a multiplexed stream mode.. this bit must be set to 1 to enable the channel information to be passed to memory. If this is not set.. the channel number (or source number) will be 0 for all streams. If vip2_rgb_out_select = 1.. then VIP_PARSER A port is connected to VPDMA |
| 14-12 | VIP2_CHR_DS_2_SRC_SELECT | R/W | 0h | Video Input Port 2 Chroma Downsampler 2 Source Select<br>000 : Path Disabled (no input to CHR_DS)<br>001 : Source from Scaler (SC_M)<br>010 : Source from Color Space Converter (CSC)<br>011 : Source from VIP_PARSER A port<br>100 : Source from VIP_PARSER B port<br>101 : Source from Transcode (422)<br>110 : Reserved<br>111 : Reserved |
| 11-9 | VIP2_CHR_DS_1_SRC_SELECT | R/W | 0h | Video Input Port 2 Chroma Downsampler 1 Source Select<br>000 : Path Disabled (no input to CHR_DS)<br>001 : Source from Scaler (SC_M)<br>010 : Source from Color Space Converter (CSC)<br>011 : Source from VIP_PARSER A port<br>100 : Source from VIP_PARSER B port<br>101 : Source from Transcode (422)<br>110 : Reserved<br>111 : Reserved |
| 8 | VIP2_RGB_OUT_HI_SELECT | R/W | 0h | Video Input Port 2 HI RGB Output Select<br>0 : Output Type is 420/422<br>1 : Output Type is RGB |
| 7 | VIP2_RGB_OUT_LO_SELECT | R/W | 0h | Video Input Port 2 LO RGB Output Select<br>0 : Output Type is 420/422<br>1 : Output Type is RGB |
| 6 | VIP2_RGB_SRC_SELECT | R/W | 0h | Video Input Port 2 RGB Output Path Select<br>0 : Source from Compositor RGB input<br>1 : Source from CSC |
| 5-3 | VIP2_SC_SRC_SELECT | R/W | 0h | Video Input Port 2 SC_M Source Select<br>000 : Path Disabled<br>001 : Source from Color Space Converter (CSC)<br>010 : Source from VIP_PARSER A port<br>011 : Source from VIP_PARSER B port<br>100 : Source from Transcode (422)<br>101 : Reserved<br>110 : Reserved<br>111 : Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-204. clkc_vip2dps Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2-0 | VIP2_CSC_SRC_SELECT | R/W | 0h | Video Input Port 2 CSC Source Select<br>000 : Path Disabled<br>001 : Source from VIP_PARSER A (422) port<br>010 : Source from VIP_PARSER B port<br>011 : Source from Transcode (422)<br>100 : Source from VIP_PARSER A (RGB) port<br>101 : Source from Compositor (RGB)<br>110 : Reserved<br>111 : Reserved |

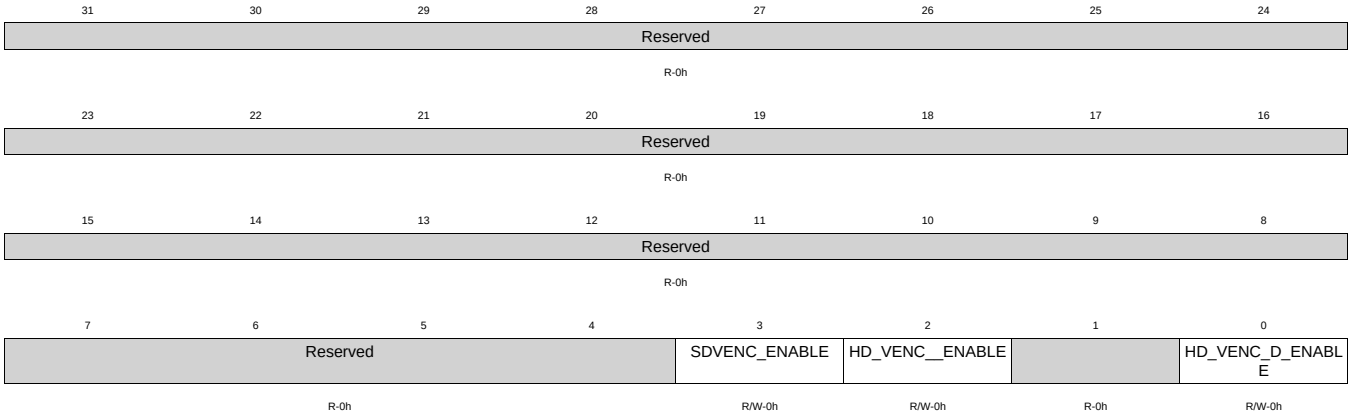

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.40 clkc_venc_ena Register (offset = 118h) [reset = 0h]

clkc_venc_ena is shown in Figure 12-293 and described in Table 12-205.

VENC Enable Register

**Figure 12-293. clkc_venc_ena Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-205. clkc_venc_ena Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | Reserved | R | 0h | |
| 3 | SDVENC_ENABLE | R/W | 0h | SD VENC Enable 0 : Disabled 1 : Enabled |
| 2 | HD_VENC__ENABLE | R/W | 0h | Digital Video Output 2 VENC Enable 0 : Disabled 1 : Enabled |
| 1 | | R | 0h | |
| 0 | HD_VENC_D_ENABLE | R/W | 0h | HDMI/Digital Video Output 1 VENC Enable 0 : Disabled 1 : Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.41 clkc_range_map Register (offset = 11Ch) [reset = 0h]

clkc_range_map is shown in Figure 12-294 and described in Table 12-206.

VC1 Range Map Register

**Figure 12-294. clkc_range_map Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RANGE_REDUCTION_IND2_ON | RANGE_REDUCTION_IND1_ON | RANGE_REDUCTION_AUX_ON | RANGE_REDUCTION_PRIM_ON | RANGE_MAP_IND2_ON | RANGE_MAPUV_IND2 | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RANGE_MAPY_IND2 | | | RANGE_MAP_IND1_ON | RANGE_MAPUV_IND1 | | | RANGE_MAPY_IND1 |
| R/W-0h | | | R/W-0h | R/W-0h | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RANGE_MAPY_IND1 | | RANGE_MAP_AUX_ON | RANGE_MAPUV_AUX | | | RANGE_MAPY_AUX | |
| R/W-0h | | R/W-0h | R/W-0h | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RANGE_MAPY_AUX | RANGE_MAP_PRIM_ON | RANGE_MAPUV_PRIM | | | RANGE_MAPY_PRIM | | |
| R/W-0h | R/W-0h | R/W-0h | | | R/W-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-206. clkc_range_map Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | RANGE_REDUCTION_IND2_ON | R/W | 0h | Range Reduction ON for IND2 transcode input |
| 30 | RANGE_REDUCTION_IND1_ON | R/W | 0h | Range Reduction ON for IND1 transcode input |
| 29 | RANGE_REDUCTION_AUX_ON | R/W | 0h | Range Reduction ON for Auxiliary input |
| 28 | RANGE_REDUCTION_PRIM_ON | R/W | 0h | Range Reduction ON for Primary input |
| 27 | RANGE_MAP_IND2_ON | R/W | 0h | Range Mapping ON for IND2 transcode input |
| 26-24 | RANGE_MAPUV_IND2 | R/W | 0h | Range Map UV for IND2 transcode input |
| 23-21 | RANGE_MAPY_IND2 | R/W | 0h | Range Map Y for IND2 transcode input |
| 20 | RANGE_MAP_IND1_ON | R/W | 0h | Range Mapping ON for IND1 transcode input |
| 19-17 | RANGE_MAPUV_IND1 | R/W | 0h | Range Map UV for IND1 transcode input |
| 16-14 | RANGE_MAPY_IND1 | R/W | 0h | Range Map Y for IND1 transcode input |
| 13 | RANGE_MAP_AUX_ON | R/W | 0h | Range Mapping ON for Auxiliary input |
| 12-10 | RANGE_MAPUV_AUX | R/W | 0h | Range Map UV for Auxiliary input |
| 9-7 | RANGE_MAPY_AUX | R/W | 0h | Range Map Y for Auxiliary input |
| 6 | RANGE_MAP_PRIM_ON | R/W | 0h | Range Mapping ON for Primary input |
| 5-3 | RANGE_MAPUV_PRIM | R/W | 0h | Range Map UV for Primary input |
| 2-0 | RANGE_MAPY_PRIM | R/W | 0h | Range Map Y for Primary input |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.7.42 clkc_underflow Register (offset = 120h) [reset = 0h]

clkc_underflow is shown in Figure 12-295 and described in Table 12-207.

VENC Underflow Status Register

**Figure 12-295. clkc_underflow Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | SD_UNDERFLOW_STAT | DVO2_UNDERFLOW_STAT | | HDMI_UNDERFLOW_STAT |
| R-0h | | | | R/W-0h | R/W-0h | R-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-207. clkc_underflow Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | Reserved | R | 0h | |
| 3 | SD_UNDERFLOW_STAT | R/W | 0h | SD VENC Underflow status Read 1 : Underflow Read 0 : NOT Underflow Write 1 to clear |
| 2 | DVO2_UNDERFLOW_STAT | R/W | 0h | DVO2 VENC Underflow status Read 1 : Underflow Read 0 : NOT Underflow Write 1 to clear |
| 1 | | | 0h | |
| 0 | HDMI_UNDERFLOW_STAT | R/W | 0h | HDMI/DVO1 VENC Underflow status Read 1 : HDMI/DVO1 Underflow Read 0 : HDMI/DVO1 NOT Underflow Write 1 to clear |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8 VPDMA Registers

Table 12-208 lists the memory-mapped registers for the VPDMA. All register offset addresses not listed in Table 12-208 should be considered as reserved locations and the register contents should not be modified.

**Table 12-208. VPDMA REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | VPDMA_pid | VPDMA Peripheral ID Register | Section 12.3.8.1 |
| 4h | VPDMA_list_addr | VPDMA List Address Register | Section 12.3.8.2 |
| 8h | VPDMA_list_attr | VPDMA List Attribute Register | Section 12.3.8.3 |
| Ch | VPDMA_list_stat_sync | VPDMA List Status Register | Section 12.3.8.4 |
| 10h | VPDMA_vpi_ctl_address | VPDMA VPI Control Address Register (Reserved) | Section 12.3.8.5 |
| 14h | VPDMA_vpi_ctl_data | VPDMA VPI Control Data Register (Reserved) | Section 12.3.8.6 |
| 18h | VPDMA_bg_rgb | VPDMA Background RGB Register | Section 12.3.8.7 |
| 1Ch | VPDMA_bg_yuv | VPDMA Background YUV Register | Section 12.3.8.8 |
| 20h | VPDMA_descriptor_top | Reserved | Section 12.3.8.9 |
| 24h | VPDMA_descriptor_bottom | Reserved | Section 12.3.8.10 |
| 28h | VPDMA_current_descriptor | Reserved | Section 12.3.8.11 |
| 2Ch | VPDMA_descriptor_status_control | Reserved | Section 12.3.8.12 |
| 40h | VPDMA_int0_channel0_int_stat | VPDMA Interrupt 0 Channel 0 Status Register | Section 12.3.8.13 |
| 44h | VPDMA_int0_channel0_int_mask | VPDMA Interrupt 0 Channel 0 Mask Register | Section 12.3.8.14 |
| 48h | VPDMA_int0_channel1_int_stat | VPDMA Interrupt 0 Channel 1 Status Register | Section 12.3.8.15 |
| 4Ch | VPDMA_int0_channel1_int_mask | VPDMA Interrupt 0 Channel 1 Mask Register | Section 12.3.8.16 |
| 50h | VPDMA_int0_channel2_int_stat | VPDMA Interrupt 0 Channel 2 Status Register | Section 12.3.8.17 |
| 54h | VPDMA_int0_channel2_int_mask | VPDMA Interrupt 0 Channel 2 Mask Register | Section 12.3.8.18 |
| 58h | VPDMA_int0_channel3_int_stat | VPDMA Interrupt 0 Channel 3 Status Register | Section 12.3.8.19 |
| 5Ch | VPDMA_int0_channel3_int_mask | VPDMA Interrupt 0 Channel 3 Mask Register | Section 12.3.8.20 |
| 60h | VPDMA_int0_channel4_int_stat | VPDMA Interrupt 0 Channel 4 Status Register | Section 12.3.8.21 |
| 64h | VPDMA_int0_channel4_int_mask | VPDMA Interrupt 0 Channel 4 Mask Register | Section 12.3.8.22 |
| 68h | VPDMA_int0_channel5_int_stat | VPDMA Interrupt 0 Channel 5 Status Register | Section 12.3.8.23 |
| 6Ch | VPDMA_int0_channel5_int_mask | VPDMA Interrupt 0 Channel 5 Mask Register | Section 12.3.8.24 |
| 70h | VPDMA_int0_channel6_int_stat | VPDMA Interrupt 0 Channel 6 Status Register | Section 12.3.8.25 |
| 74h | VPDMA_int0_channel6_int_mask | VPDMA Interrupt 0 Channel 6 Mask Register | Section 12.3.8.26 |
| 78h | VPDMA_int0_client0_int_stat | VPDMA Interrupt 0 Client 0 Status Register | Section 12.3.8.27 |
| 7Ch | VPDMA_int0_client0_int_mask | VPDMA Interrupt 0 Client 0 Mask Register | Section 12.3.8.28 |
| 80h | VPDMA_int0_client1_int_stat | VPDMA Interrupt 0 Client 1 Status Register | Section 12.3.8.29 |
| 84h | VPDMA_int0_client1_int_mask | VPDMA Interrupt 0 Client 1 Mask Register | Section 12.3.8.30 |
| 88h | VPDMA_int0_list0_int_stat | VPDMA Interrupt 0 List 0 Status Register | Section 12.3.8.31 |
| 8Ch | VPDMA_int0_list0_int_mask | VPDMA Interrupt 0 List 0 Mask Register | Section 12.3.8.32 |
| 90h | VPDMA_int1_channel0_int_stat | VPDMA Interrupt 1 Channel 0 Status Register | Section 12.3.8.33 |
| 94h | VPDMA_int1_channel0_int_mask | VPDMA Interrupt 1 Channel 0 Mask Register | Section 12.3.8.34 |
| 98h | VPDMA_int1_channel1_int_stat | VPDMA Interrupt 1 Channel 1 Status Register | Section 12.3.8.35 |
| 9Ch | VPDMA_int1_channel1_int_mask | VPDMA Interrupt 1 Channel 1 Mask Register | Section 12.3.8.36 |
| A0h | VPDMA_int1_channel2_int_stat | VPDMA Interrupt 1 Channel 2 Status Register | Section 12.3.8.37 |
| A4h | VPDMA_int1_channel2_int_mask | VPDMA Interrupt 1 Channel 2 Mask Register | Section 12.3.8.38 |
| A8h | VPDMA_int1_channel3_int_stat | VPDMA Interrupt 1 Channel 3 Status Register | Section 12.3.8.39 |
| ACh | VPDMA_int1_channel3_int_mask | VPDMA Interrupt 1 Channel 3 Mask Register | Section 12.3.8.40 |
| B0h | VPDMA_int1_channel4_int_stat | VPDMA Interrupt 1 Channel 4 Status Register | Section 12.3.8.41 |
| B4h | VPDMA_int1_channel4_int_mask | VPDMA Interrupt 1 Channel 4 Mask Register | Section 12.3.8.42 |
| B8h | VPDMA_int1_channel5_int_stat | VPDMA Interrupt 1 Channel 5 Status Register | Section 12.3.8.43 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-208. VPDMA REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| BCh | VPDMA_int1_channel5_int_mask | VPDMA Interrupt 1 Channel 5 Mask Register | Section 12.3.8.44 |
| C0h | VPDMA_int1_channel6_int_stat | VPDMA Interrupt 1 Channel 6 Status Register | Section 12.3.8.45 |
| C4h | VPDMA_int1_channel6_int_mask | VPDMA Interrupt 1 Channel 6 Mask Register | Section 12.3.8.46 |
| C8h | VPDMA_int1_client0_int_stat | VPDMA Interrupt 1 Client 0 Status Register | Section 12.3.8.47 |
| CCh | VPDMA_int1_client0_int_mask | VPDMA Interrupt 1 Client 0 Mask Register | Section 12.3.8.48 |
| D0h | VPDMA_int1_client1_int_stat | VPDMA Interrupt 1 Client 1 Status Register | Section 12.3.8.49 |
| D4h | VPDMA_int1_client1_int_mask | VPDMA Interrupt 1 Client 1 Mask Register | Section 12.3.8.50 |
| D8h | VPDMA_int1_list0_int_stat | VPDMA Interrupt 1 List 0 Status Register | Section 12.3.8.51 |
| DCh | VPDMA_int1_list0_int_mask | VPDMA Interrupt 1 List 0 Mask Register | Section 12.3.8.52 |
| E0h | VPDMA_int2_channel0_int_stat | VPDMA Interrupt 2 Channel 0 Status Register | Section 12.3.8.53 |
| E4h | VPDMA_int2_channel0_int_mask | VPDMA Interrupt 2 Channel 0 Mask Register | Section 12.3.8.54 |
| E8h | VPDMA_int2_channel1_int_stat | VPDMA Interrupt 2 Channel 1 Status Register | Section 12.3.8.55 |
| ECh | VPDMA_int2_channel1_int_mask | VPDMA Interrupt 2 Channel 1 Mask Register | Section 12.3.8.56 |
| F0h | VPDMA_int2_channel2_int_stat | VPDMA Interrupt 2 Channel 2 Status Register | Section 12.3.8.57 |
| F4h | VPDMA_int2_channel2_int_mask | VPDMA Interrupt 2 Channel 2 Mask Register | Section 12.3.8.58 |
| F8h | VPDMA_int2_channel3_int_stat | VPDMA Interrupt 2 Channel 3 Status Register | Section 12.3.8.59 |
| FCh | VPDMA_int2_channel3_int_mask | VPDMA Interrupt 2 Channel 3 Mask Register | Section 12.3.8.60 |
| 100h | VPDMA_int2_channel4_int_stat | VPDMA Interrupt 2 Channel 4 Status Register | Section 12.3.8.61 |
| 104h | VPDMA_int2_channel4_int_mask | VPDMA Interrupt 2 Channel 4 Mask Register | Section 12.3.8.62 |
| 108h | VPDMA_int2_channel5_int_stat | VPDMA Interrupt 2 Channel 5 Status Register | Section 12.3.8.63 |
| 10Ch | VPDMA_int2_channel5_int_mask | VPDMA Interrupt 2 Channel 5 Mask Register | Section 12.3.8.64 |
| 110h | VPDMA_int2_channel6_int_stat | VPDMA Interrupt 2 Channel 6 Status Register | Section 12.3.8.65 |
| 114h | VPDMA_int2_channel6_int_mask | VPDMA Interrupt 2 Channel 6 Mask Register | Section 12.3.8.66 |
| 118h | VPDMA_int2_client0_int_stat | VPDMA Interrupt 2 Client 0 Status Register | Section 12.3.8.67 |
| 11Ch | VPDMA_int2_client0_int_mask | VPDMA Interrupt 2 Client 0 Mask Register | Section 12.3.8.68 |
| 120h | VPDMA_int2_client1_int_stat | VPDMA Interrupt 2 Client 1 Status Register | Section 12.3.8.69 |
| 124h | VPDMA_int2_client1_int_mask | VPDMA Interrupt 2 Client 1 Mask Register | Section 12.3.8.70 |
| 128h | VPDMA_int2_list0_int_stat | VPDMA Interrupt 2 List 0 Status Register | Section 12.3.8.71 |
| 12Ch | VPDMA_int2_list0_int_mask | VPDMA Interrupt 2 List 0 Mask Register | Section 12.3.8.72 |
| 130h | VPDMA_int3_channel0_int_stat | VPDMA Interrupt 3 Channel 0 Status Register | Section 12.3.8.73 |
| 134h | VPDMA_int3_channel0_int_mask | VPDMA Interrupt 3 Channel 0 Mask Register | Section 12.3.8.74 |
| 138h | VPDMA_int3_channel1_int_stat | VPDMA Interrupt 3 Channel 1 Status Register | Section 12.3.8.75 |
| 13Ch | VPDMA_int3_channel1_int_mask | VPDMA Interrupt 3 Channel 1 Mask Register | Section 12.3.8.76 |
| 140h | VPDMA_int3_channel2_int_stat | VPDMA Interrupt 3 Channel 2 Status Register | Section 12.3.8.77 |
| 144h | VPDMA_int3_channel2_int_mask | VPDMA Interrupt 3 Channel 2 Mask Register | Section 12.3.8.78 |
| 148h | VPDMA_int3_channel3_int_stat | VPDMA Interrupt 3 Channel 3 Status Register | Section 12.3.8.79 |
| 14Ch | VPDMA_int3_channel3_int_mask | VPDMA Interrupt 3 Channel 3 Mask Register | Section 12.3.8.80 |
| 150h | VPDMA_int3_channel4_int_stat | VPDMA Interrupt 3 Channel 4 Status Register | Section 12.3.8.81 |
| 154h | VPDMA_int3_channel4_int_mask | VPDMA Interrupt 3 Channel 4 Mask Register | Section 12.3.8.82 |
| 158h | VPDMA_int3_channel5_int_stat | VPDMA Interrupt 3 Channel 5 Status Register | Section 12.3.8.83 |
| 15Ch | VPDMA_int3_channel5_int_mask | VPDMA Interrupt 3 Channel 5 Mask Register | Section 12.3.8.84 |
| 160h | VPDMA_int3_channel6_int_stat | VPDMA Interrupt 3 Channel 6 Status Register | Section 12.3.8.85 |
| 164h | VPDMA_int3_channel6_int_mask | VPDMA Interrupt 3 Channel 6 Mask Register | Section 12.3.8.86 |
| 168h | VPDMA_int3_client0_int_stat | VPDMA Interrupt 3 Client 0 Status Register | Section 12.3.8.87 |
| 16Ch | VPDMA_int3_client0_int_mask | VPDMA Interrupt 3 Client 0 Mask Register | Section 12.3.8.88 |
| 170h | VPDMA_int3_client1_int_stat | VPDMA Interrupt 3 Client 1 Status Register | Section 12.3.8.89 |
| 174h | VPDMA_int3_client1_int_mask | VPDMA Interrupt 3 Client 1 Mask Register | Section 12.3.8.90 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-208. VPDMA REGISTERS (continued)**

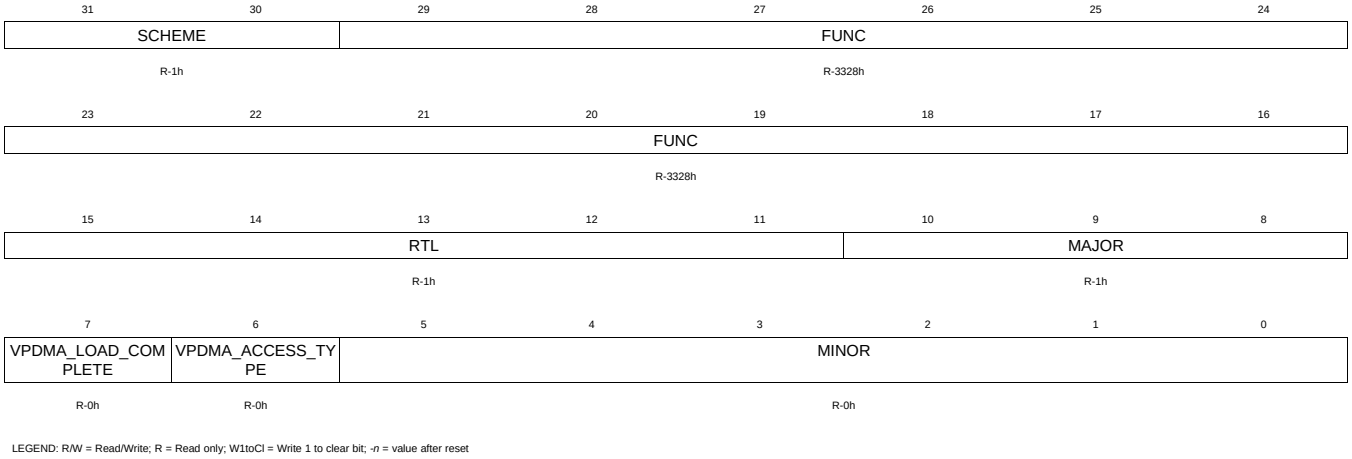
| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 178h | VPDMA_int3_list0_int_stat | VPDMA Interrupt 3 List 0 Status Register | Section 12.3.8.91 |
| 17Ch | VPDMA_int3_list0_int_mask | VPDMA Interrupt 3 List 0 Mask Register | Section 12.3.8.92 |
| 300h | VPDMA_dei_hq_1_chroma_cstat | VPDMA DEI Field 1 Chroma cstat | Section 12.3.8.93 |
| 304h | VPDMA_dei_hq_1_luma_cstat | VPDMA DEI Field 1 Luma cstat | Section 12.3.8.94 |
| 308h | VPDMA_dei_hq_2_luma_cstat | VPDMA DEI Field 2 Luma cstat | Section 12.3.8.95 |
| 30Ch | VPDMA_dei_hq_2_chroma_cstat | VPDMA DEI Field 2 Chroma cstat | Section 12.3.8.96 |
| 310h | VPDMA_dei_hq_3_luma_cstat | VPDMA DEI Field 3 Luma cstat | Section 12.3.8.97 |
| 314h | VPDMA_dei_hq_3_chroma_cstat | VPDMA DEI Field 3 Chroma cstat | Section 12.3.8.98 |
| 330h | VPDMA_dei_hq_mv_in_cstat | VPDMA DEI Motion Vector In cstat | Section 12.3.8.99 |
| 33Ch | VPDMA_dei_hq_mv_out_cstat | VPDMA DEI Motion Vector Out cstat | Section 12.3.8.100 |
| 344h | VPDMA_dei_sc_out_cstat | VPDMA DEI Scaler (SC_1) Out cstat | Section 12.3.8.101 |
| 348h | VPDMA_pip_wrbk_cstat | VPDMA PIP WRBK (BP0) cstat | Section 12.3.8.102 |
| 34Ch | VPDMA_sc_in_chroma_cstat | VPDMA Aux (SC_2) Chroma cstat | Section 12.3.8.103 |
| 350h | VPDMA_sc_in_luma_cstat | VPDMA Aux (SC_2) Luma cstat | Section 12.3.8.104 |
| 374h | VPDMA_sc_out_cstat | VPDMA Aux Scaler (SC_2) Out cstat | Section 12.3.8.105 |
| 378h | VPDMA_comp_wrbk_cstat | VPDMA COMP WRBK (BP1) cstat | Section 12.3.8.106 |
| 37Ch | VPDMA_grpx1_data_cstat | VPDMA GRPX1 Data cstat | Section 12.3.8.107 |
| 380h | VPDMA_grpx2_data_cstat | VPDMA GRPX2 Data cstat | Section 12.3.8.108 |
| 384h | VPDMA_grpx3_data_cstat | VPDMA GRPX3 Data cstat | Section 12.3.8.109 |
| 388h | VPDMA_vip1_lo_y_cstat | VPDMA VIP1 lo y cstat | Section 12.3.8.110 |
| 38Ch | VPDMA_vip1_lo_uv_cstat | VPDMA VIP1 lo uv cstat | Section 12.3.8.111 |
| 390h | VPDMA_vip1_up_y_cstat | VPDMA VIP1 up y cstat | Section 12.3.8.112 |
| 394h | VPDMA_vip1_up_uv_cstat | VPDMA VIP1 up uv cstat | Section 12.3.8.113 |
| 398h | VPDMA_vip2_lo_y_cstat | VPDMA VIP2 lo y cstat | Section 12.3.8.114 |
| 39Ch | VPDMA_vip2_lo_uv_cstat | VPDMA VIP2 lo uv cstat | Section 12.3.8.115 |
| 3A0h | VPDMA_vip2_up_y_cstat | VPDMA VIP2 up y cstat | Section 12.3.8.116 |
| 3A4h | VPDMA_vip2_up_uv_cstat | VPDMA VIP2 up uv cstat | Section 12.3.8.117 |
| 3A8h | VPDMA_grpx1_st_cstat | VPDMA GRPX1 Stencil cstat | Section 12.3.8.118 |
| 3ACh | VPDMA_grpx2_st_cstat | VPDMA GRPX2 Stencil cstat | Section 12.3.8.119 |
| 3B0h | VPDMA_grpx3_st_cstat | VPDMA GRPX3 Stencil cstat | Section 12.3.8.120 |
| 3B4h | VPDMA_nf_422_in_cstat | VPDMA NF 422 In cstat | Section 12.3.8.121 |
| 3B8h | VPDMA_nf_420_y_in_cstat | VPDMA NF 420 Luma In cstat | Section 12.3.8.122 |
| 3BCh | VPDMA_nf_420_uv_in_cstat | VPDMA NF 420 Chroma In cstat | Section 12.3.8.123 |
| 3C0h | VPDMA_nf_420_y_out_cstat | VPDMA NF 420 Luma Out cstat | Section 12.3.8.124 |
| 3C4h | VPDMA_nf_420_uv_out_cstat | VPDMA NF 420 Chroma Out cstat | Section 12.3.8.125 |
| 3CCh | VPDMA_vbi_sdvenc_cstat | VPDMA VBI SDVENC cstat | Section 12.3.8.126 |
| 3D0h | VPDMA_vpi_ctl_cstat | VPDMA VPI Control cstat | Section 12.3.8.127 |
| 3D4h | VPDMA_hdmi_wrbk_out_cstat | VPDMA HDMI WRBK (SC_5) Out cstat | Section 12.3.8.128 |
| 3D8h | VPDMA_trans1_chroma_cstat | VPDMA Transcode 1 Chroma cstat | Section 12.3.8.129 |
| 3DCh | VPDMA_trans1_luma_cstat | VPDMA Transcode 1 Luma cstat | Section 12.3.8.130 |
| 3E0h | VPDMA_trans2_chroma_cstat | VPDMA Transcode 2 Chroma cstat | Section 12.3.8.131 |
| 3E4h | VPDMA_trans2_luma_cstat | VPDMA Transcode 2 Luma cstat | Section 12.3.8.132 |
| 3E8h | VPDMA_vip1_anc_a_cstat | VPDMA VIP1 Ancillary A cstat | Section 12.3.8.133 |
| 3ECh | VPDMA_vip1_anc_b_cstat | VPDMA VIP1 Ancillary B cstat | Section 12.3.8.134 |
| 3F0h | VPDMA_vip2_anc_a_cstat | VPDMA VIP2 Ancillary A cstat | Section 12.3.8.135 |
| 3F4h | VPDMA_vip2_anc_b_cstat | VPDMA VIP2 Ancillary B cstat | Section 12.3.8.136 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.1 VPDMA_pid Register (offset = 0h) [reset = 73280900h]

VPDMA_pid is shown in Figure 12-296 and described in Table 12-209.

**Figure 12-296. VPDMA_pid Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-209. VPDMA_pid Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | SCHEME | R | 1h | The scheme of the register used. Currently this is PDR 3.5 Scheme |
| 29-16 | FUNC | R | 3328h | The funcition of the module being used. The value is for vpdma. |
| 15-11 | RTL | R | 1h | RTL Release Version The PDR release number of this IP. After Bootup this value becomes the firmware Revision ID |
| 10-8 | MAJOR | R | 1h | Major Release Number |
| 7 | VPDMA_LOAD_COMPLETE | R | 0h | This bit will be 1 when the VPDMA state machines image and data image have successfuly been fetched and loaded. |
| 6 | VPDMA_ACCESS_TYPE | R | 0h | After bootup this bit states how DMA transaction are setup by lists or through register access. 0: Lists 1: Register Access |
| 5-0 | MINOR | R | 0h | Minor Release Number |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.2 VPDMA_list_addr Register (offset = 4h) [reset = 0h]

VPDMA_list_addr is shown in Figure 12-297 and described in Table 12-210.

**Figure 12-297. VPDMA_list_addr Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LIST_ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-210. VPDMA_list_addr Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | LIST_ADDR | R/W | 0h | Location of a new list of descriptors. This register must be written with the VPDMA configuration location after reset. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.3 VPDMA_list_attr Register (offset = 8h) [reset = 0h]

VPDMA_list_attr is shown in Figure 12-298 and described in Table 12-211.

**Figure 12-298. VPDMA_list_attr Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | LIST_NUM | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | STOP | RDY | LIST_TYPE | | |
| R-0h | | | W-0h | R-0h | R/W-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LIST_SIZE | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LIST_SIZE | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-211. VPDMA_list_attr Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-24 | LIST_NUM | R/W | 0h | The list number that should be assigned to the list located at LIST_ADDR. If the list is still active this will block all future list writes until the list is available. |
| 23-21 | Reserved | R | 0h | |
| 20 | STOP | W | 0h | This bit is written with the LIST_NUMBER field to stop a list. When this bit is written a one the list specified by the LIST_NUMBER is sent a stop signal and will finish the current frame of transfers and then free the list resources. |
| 19 | RDY | R | 0h | This bit is low when a new list cannot be written to the LIST_ADDR register. The reasons this bit would be low are at initial startup if the LIST_MANAGER State Machine image has not completed loading. It also would be low if the last write to the LIST_ATTR attempted to start a list that is currently active. When this bit is low any writes to the list address register will cause access to not be accepted until this bit has set by the previous list having completed. |
| 18-16 | LIST_TYPE | R/W | 0h | The type of list that has been generated. 0: Normal List 1: Self-Modifying List 2: List Doorbell Others Reserved for future use |
| 15-0 | LIST_SIZE | R/W | 0h | Number of 128 bit word in the new list of descriptors. Writes to this register will activate the list in the list stack of the list manager and begin transfer of the list into VPDMA. This size can not be 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.4 VPDMA_list_stat_sync Register (offset = Ch) [reset = 0h]

VPDMA_list_stat_sync is shown in Figure 12-299 and described in Table 12-212.

**Figure 12-299. VPDMA_list_stat_sync Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| LIST7_BUSY | LIST6_BUSY | LIST5_BUSY | LIST4_BUSY | LIST3_BUSY | LIST2_BUSY | LIST1_BUSY | LIST0_BUSY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SYNC_LISTS7 | SYNC_LISTS6 | SYNC_LISTS5 | SYNC_LISTS4 | SYNC_LISTS3 | SYNC_LISTS2 | SYNC_LISTS1 | SYNC_LISTS0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-212. VPDMA_list_stat_sync Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23 | LIST7_BUSY | R | 0h | The list 7 is currently running. Any attempt to load a new list to list 7 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0. |
| 22 | LIST6_BUSY | R | 0h | The list 6 is currently running. Any attempt to load a new list to list 6 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0. |
| 21 | LIST5_BUSY | R | 0h | The list 5 is currently running. Any attempt to load a new list to list 5 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0. |
| 20 | LIST4_BUSY | R | 0h | The list 4 is currently running. Any attempt to load a new list to list 4 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0. |
| 19 | LIST3_BUSY | R | 0h | The list 3 is currently running. Any attempt to load a new list to list 3 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0. |
| 18 | LIST2_BUSY | R | 0h | The list 2 is currently running. Any attempt to load a new list to list 2 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0. |
| 17 | LIST1_BUSY | R | 0h | The list 1 is currently running. Any attempt to load a new list to list 1 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0. |
| 16 | LIST0_BUSY | R | 0h | The list 0 is currently running. Any attempt to load a new list to list 0 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0. |
| 15-8 | Reserved | R | 0h | |
| 7 | SYNC_LISTS7 | W | 0h | Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 7 waiting on it. |
| 6 | SYNC_LISTS6 | W | 0h | Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 6 waiting on it. |
| 5 | SYNC_LISTS5 | W | 0h | Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 5 waiting on it. |
| 4 | SYNC_LISTS4 | W | 0h | Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 4 waiting on it. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-212. VPDMA_list_stat_sync Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | SYNC_LISTS3 | W | 0h | Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 3 waiting on it. |
| 2 | SYNC_LISTS2 | W | 0h | Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 2 waiting on it. |
| 1 | SYNC_LISTS1 | W | 0h | Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 1 waiting on it. |
| 0 | SYNC_LISTS0 | W | 0h | Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 0 waiting on it. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.5 VPDMA_vpi_ctl_address Register (offset = 10h) [reset = 0h]

VPDMA_vpi_ctl_address is shown in Figure 12-300 and described in Table 12-213.

**Figure 12-300. VPDMA_vpi_ctl_address Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-213. VPDMA_vpi_ctl_address Register Field Descriptions**

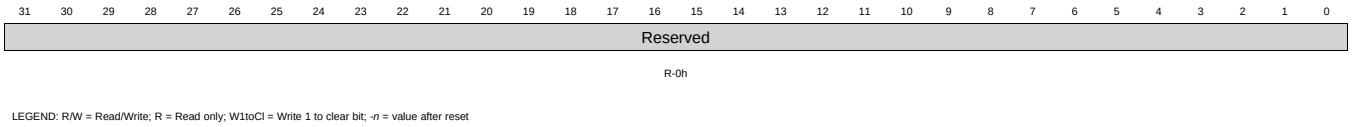
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.6 VPDMA_vpi_ctl_data Register (offset = 14h) [reset = 0h]

VPDMA_vpi_ctl_data is shown in Figure 12-301 and described in Table 12-214.

**Figure 12-301. VPDMA_vpi_ctl_data Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-214. VPDMA_vpi_ctl_data Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.7 VPDMA_bg_rgb Register (offset = 18h) [reset = 0h]

VPDMA_bg_rgb is shown in Figure 12-302 and described in Table 12-215.

**Figure 12-302. VPDMA_bg_rgb Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GREEN | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BLUE | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BLEND | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-215. VPDMA_bg_rgb Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RED | R/W | 0h | The red value to give on an RGB data port for a blank pixel when using virtual video buffering |
| 23-16 | GREEN | R/W | 0h | The green value to give on an RGB data port for a blank pixel when using virtual video buffering |
| 15-8 | BLUE | R/W | 0h | The blue value to give on an RGB data port for a blank pixel when using virtual video buffering |
| 7-0 | BLEND | R/W | 0h | The blend value to give on an RGB data port for a blank pixel when using virtual video buffering |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.8 VPDMA_bg_yuv Register (offset = 1Ch) [reset = 0h]

VPDMA_bg_yuv is shown in Figure 12-303 and described in Table 12-216.

**Figure 12-303. VPDMA_bg_yuv Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Y | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CR | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CB | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-216. VPDMA_bg_yuv Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-16 | Y | R/W | 0h | The Y value to give on a YUV data port for a blank pixel when using virtual video buffering |
| 15-8 | CR | R/W | 0h | The Cr value to give on a YUV data port for a blank pixel when using virtual video buffering |
| 7-0 | CB | R/W | 0h | The Cb value to give on a YUV data port for a blank pixel when using virtual video buffering |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.9 VPDMA_descriptor_top Register (offset = 20h) [reset = 0h]

VPDMA_descriptor_top is shown in Figure 12-304 and described in Table 12-217.

**Figure 12-304. VPDMA_descriptor_top Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-217. VPDMA_descriptor_top Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.10 VPDMA_descriptor_bottom Register (offset = 24h) [reset = 0h]

VPDMA_descriptor_bottom is shown in Figure 12-305 and described in Table 12-218.

**Figure 12-305. VPDMA_descriptor_bottom Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-218. VPDMA_descriptor_bottom Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.11 VPDMA_current_descriptor Register (offset = 28h) [reset = 0h]

VPDMA_current_descriptor is shown in Figure 12-306 and described in Table 12-219.

**Figure 12-306. VPDMA_current_descriptor Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-219. VPDMA_current_descriptor Register Field Descriptions**

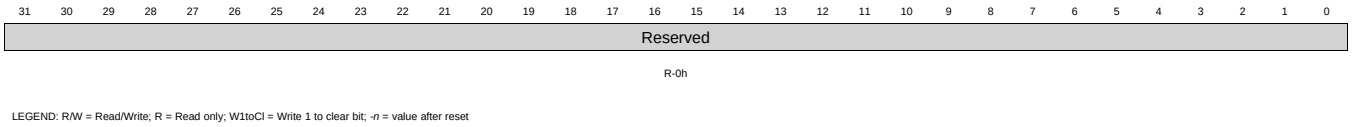
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.12 VPDMA_descriptor_status_control Register (offset = 2Ch) [reset = 0h]

VPDMA_descriptor_status_control is shown in Figure 12-307 and described in Table 12-220.

**Figure 12-307. VPDMA_descriptor_status_control Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-220. VPDMA_descriptor_status_control Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.13 VPDMA_int0_channel0_int_stat Register (offset = 40h) [reset = 0h]

VPDMA_int0_channel0_int_stat is shown in Figure 12-308 and described in Table 12-221.

**Figure 12-308. VPDMA_int0_channel0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_GRPX3 | INT_STAT_GRPX2 | INT_STAT_GRPX1 | INT_STAT_SCALER_OUT | Reserved | | | |
| W-0h | W-0h | W-0h | W-0h | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | INT_STAT_SCALER_CHROMA | INT_STAT_SCALER_LUMA | INT_STAT_HQ_SCALER | Reserved |
| R-0h | | | | W-0h | W-0h | W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_HQ_MV_OUT | Reserved | | INT_STAT_HQ_MV | Reserved | | | |
| W-0h | R-0h | | W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_STAT_HQ_VID3_CHROMA | INT_STAT_HQ_VID3_LUMA | INT_STAT_HQ_VID2_CHROMA | INT_STAT_HQ_VID2_LUMA | INT_STAT_HQ_VID1_CHROMA | INT_STAT_HQ_VID1_LUMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-221. VPDMA_int0_channel0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_GRPX3 | W | 0h | The last read DMA transaction has occurred for channel grpx3 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_GRPX2 | W | 0h | The last read DMA transaction has occurred for channel grpx2 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_GRPX1 | W | 0h | The last read DMA transaction has occurred for channel grpx1 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_SCALER_OUT | W | 0h | The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27-20 | Reserved | R | 0h | |
| 19 | INT_STAT_SCALER_CHROMA | W | 0h | The last write DMA transaction has completed for channel scaler_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-221. VPDMA_int0_channel0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | INT_STAT_SCALER_LUMA | W | 0h | The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_HQ_SCALER | W | 0h | The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | Reserved | R | 0h | |
| 15 | INT_STAT_HQ_MV_OUT | W | 0h | The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_STAT_HQ_MV | W | 0h | The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11-6 | Reserved | R | 0h | |
| 5 | INT_STAT_HQ_VID3_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid3_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_3_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_HQ_VID3_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid3_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_3_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_HQ_VID2_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid2_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_2_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_HQ_VID2_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid2_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_2_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-221. VPDMA_int0_channel0_int_stat Register Field Descriptions (continued)**

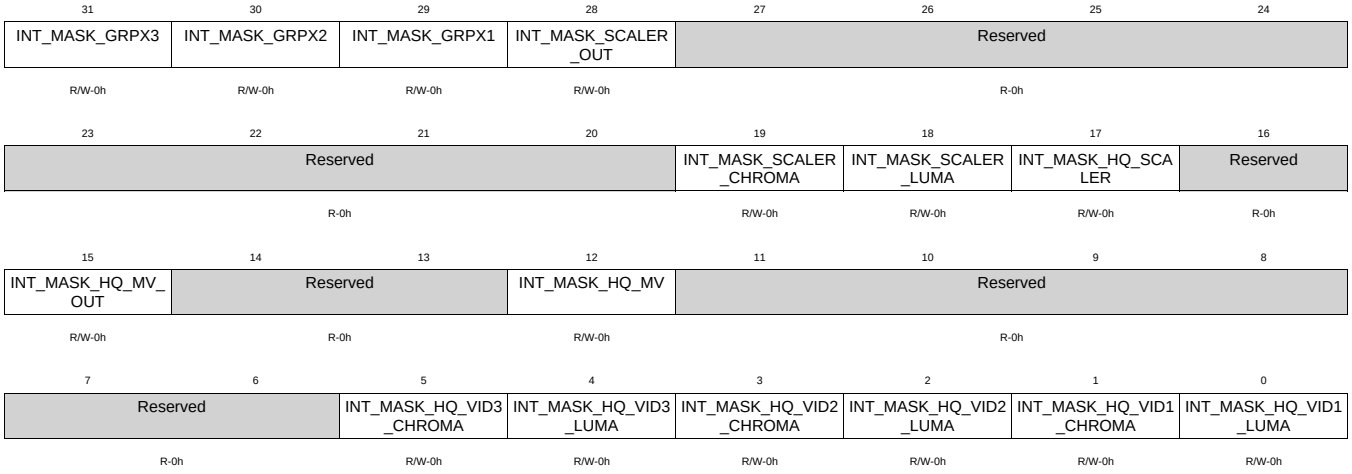
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | INT_STAT_HQ_VID1_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid1_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_1_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_HQ_VID1_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid1_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_1_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.14 VPDMA_int0_channel0_int_mask Register (offset = 44h) [reset = 0h]

VPDMA_int0_channel0_int_mask is shown in Figure 12-309 and described in Table 12-222.

**Figure 12-309. VPDMA_int0_channel0_int_mask Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-222. VPDMA_int0_channel0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_GRPX3 | R/W | 0h | The interrupt for Graphcis 2 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_GRPX2 | R/W | 0h | The interrupt for Graphics 1 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_GRPX1 | R/W | 0h | The interrupt for Graphics 0 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_SCALER_OUT | R/W | 0h | The interrupt for Low Cost DEI Scalar Write to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27-20 | Reserved | R | 0h | |
| 19 | INT_MASK_SCALER_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_SCALER_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_HQ_SCALER | R/W | 0h | The interrupt for High Quality DEI Scaler Write to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | Reserved | R | 0h | |
| 15 | INT_MASK_HQ_MV_OUT | R/W | 0h | The interrupt for Low Cost DEI Motion Vector Write should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_MASK_HQ_MV | R/W | 0h | The interrupt for Low Cost DEI Motion Vector should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11-6 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-222. VPDMA_int0_channel0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_MASK_HQ_VID3_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_HQ_VID3_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_HQ_VID2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_HQ_VID2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_HQ_VID1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_HQ_VID1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.15 VPDMA_int0_channel1_int_stat Register (offset = 48h) [reset = 0h]

VPDMA_int0_channel1_int_stat is shown in Figure 12-310 and described in Table 12-223.

**Figure 12-310. VPDMA_int0_channel1_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP1_MULT_PORTB_SRC9 | INT_STAT_VIP1_MULT_PORTB_SRC8 | INT_STAT_VIP1_MULT_PORTB_SRC7 | INT_STAT_VIP1_MULT_PORTB_SRC6 | INT_STAT_VIP1_MULT_PORTB_SRC5 | INT_STAT_VIP1_MULT_PORTB_SRC4 | INT_STAT_VIP1_MULT_PORTB_SRC3 | INT_STAT_VIP1_MULT_PORTB_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP1_MULT_PORTB_SRC1 | INT_STAT_VIP1_MULT_PORTB_SRC0 | INT_STAT_VIP1_MULT_PORTA_SRC15 | INT_STAT_VIP1_MULT_PORTA_SRC14 | INT_STAT_VIP1_MULT_PORTA_SRC13 | INT_STAT_VIP1_MULT_PORTA_SRC12 | INT_STAT_VIP1_MULT_PORTA_SRC11 | INT_STAT_VIP1_MULT_PORTA_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP1_MULT_PORTA_SRC9 | INT_STAT_VIP1_MULT_PORTA_SRC8 | INT_STAT_VIP1_MULT_PORTA_SRC7 | INT_STAT_VIP1_MULT_PORTA_SRC6 | INT_STAT_VIP1_MULT_PORTA_SRC5 | INT_STAT_VIP1_MULT_PORTA_SRC4 | INT_STAT_VIP1_MULT_PORTA_SRC3 | INT_STAT_VIP1_MULT_PORTA_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_MULT_PORTA_SRC1 | INT_STAT_VIP1_MULT_PORTA_SRC0 | INT_STAT_GRPX3_CLUT | INT_STAT_GRPX2_CLUT | INT_STAT_GRPX1_CLUT | INT_STAT_GRPX3_STENCIL | INT_STAT_GRPX2_STENCIL | INT_STAT_GRPX1_STENCIL |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-223. VPDMA_int0_channel1_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP1_MULT_PORTB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP1_MULT_PORTB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP1_MULT_PORTB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP1_MULT_PORTB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_MULT_PORTB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_MULT_PORTB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-223. VPDMA_int0_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP1_MULT_PORTB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP1_MULT_PORTB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP1_MULT_PORTB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP1_MULT_PORTB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP1_MULT_PORTA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP1_MULT_PORTA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP1_MULT_PORTA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP1_MULT_PORTA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP1_MULT_PORTA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP1_MULT_PORTA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-223. VPDMA_int0_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP1_MULT_PORTA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP1_MULT_PORTA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP1_MULT_PORTA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP1_MULT_PORTA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_MULT_PORTA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_MULT_PORTA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_MULT_PORTA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_MULT_PORTA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_MULT_PORTA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_MULT_PORTA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-223. VPDMA_int0_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_GRPX3_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx3_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_GRPX2_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx2_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_GRPX1_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx1_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_GRPX3_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx3_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_GRPX2_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx2_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_GRPX1_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx1_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.16 VPDMA_int0_channel1_int_mask Register (offset = 4Ch) [reset = 0h]

VPDMA_int0_channel1_int_mask is shown in Figure 12-311 and described in Table 12-224.

**Figure 12-311. VPDMA_int0_channel1_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_PORTB_SRC9 | INT_MASK_VIP1_MULT_PORTB_SRC8 | INT_MASK_VIP1_MULT_PORTB_SRC7 | INT_MASK_VIP1_MULT_PORTB_SRC6 | INT_MASK_VIP1_MULT_PORTB_SRC5 | INT_MASK_VIP1_MULT_PORTB_SRC4 | INT_MASK_VIP1_MULT_PORTB_SRC3 | INT_MASK_VIP1_MULT_PORTB_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP1_MULT_PORTB_SRC1 | INT_MASK_VIP1_MULT_PORTB_SRC0 | INT_MASK_VIP1_MULT_PORTA_SRC15 | INT_MASK_VIP1_MULT_PORTA_SRC14 | INT_MASK_VIP1_MULT_PORTA_SRC13 | INT_MASK_VIP1_MULT_PORTA_SRC12 | INT_MASK_VIP1_MULT_PORTA_SRC11 | INT_MASK_VIP1_MULT_PORTA_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP1_MULT_PORTA_SRC9 | INT_MASK_VIP1_MULT_PORTA_SRC8 | INT_MASK_VIP1_MULT_PORTA_SRC7 | INT_MASK_VIP1_MULT_PORTA_SRC6 | INT_MASK_VIP1_MULT_PORTA_SRC5 | INT_MASK_VIP1_MULT_PORTA_SRC4 | INT_MASK_VIP1_MULT_PORTA_SRC3 | INT_MASK_VIP1_MULT_PORTA_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP1_MULT_PORTA_SRC1 | INT_MASK_VIP1_MULT_PORTA_SRC0 | INT_MASK_GRPX3_CLUT | INT_MASK_GRPX2_CLUT | INT_MASK_GRPX1_CLUT | INT_MASK_GRPX3_STENCIL | INT_MASK_GRPX2_STENCIL | INT_MASK_GRPX1_STENCIL |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-224. VPDMA_int0_channel1_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP1_MULT_PORTB_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP1_MULT_PORTB_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP1_MULT_PORTB_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP1_MULT_PORTB_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_MULT_PORTB_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_MULT_PORTB_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP1_MULT_PORTB_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP1_MULT_PORTB_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP1_MULT_PORTB_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP1_MULT_PORTB_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP1_MULT_PORTA_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-224. VPDMA_int0_channel1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP1_MULT_PORTA_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP1_MULT_PORTA_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP1_MULT_PORTA_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP1_MULT_PORTA_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP1_MULT_PORTA_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP1_MULT_PORTA_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP1_MULT_PORTA_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP1_MULT_PORTA_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP1_MULT_PORTA_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_MULT_PORTA_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_MULT_PORTA_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_MULT_PORTA_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_MULT_PORTA_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_MULT_PORTA_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_MULT_PORTA_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_GRPX3_CLUT | R/W | 0h | The interrupt for Graphics 2 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_GRPX2_CLUT | R/W | 0h | The interrupt for Graphics 1 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_GRPX1_CLUT | R/W | 0h | The interrupt for Graphics 0 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-224. VPDMA_int0_channel1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_GRPX3_STENCIL | R/W | 0h | The interrupt for Graphics 2 Stencil should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_GRPX2_STENCIL | R/W | 0h | The interrupt for Graphics 1 Stencil should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_GRPX1_STENCIL | R/W | 0h | The interrupt for Graphics 0 Stencil should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.17 VPDMA_int0_channel2_int_stat Register (offset = 50h) [reset = 0h]

VPDMA_int0_channel2_int_stat is shown in Figure 12-312 and described in Table 12-225.

**Figure 12-312. VPDMA_int0_channel2_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP1_MULT_ANCB_SRC9 | INT_STAT_VIP1_MULT_ANCB_SRC8 | INT_STAT_VIP1_MULT_ANCB_SRC7 | INT_STAT_VIP1_MULT_ANCB_SRC6 | INT_STAT_VIP1_MULT_ANCB_SRC5 | INT_STAT_VIP1_MULT_ANCB_SRC4 | INT_STAT_VIP1_MULT_ANCB_SRC3 | INT_STAT_VIP1_MULT_ANCB_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP1_MULT_ANCB_SRC1 | INT_STAT_VIP1_MULT_ANCB_SRC0 | INT_STAT_VIP1_MULT_ANCA_SRC15 | INT_STAT_VIP1_MULT_ANCA_SRC14 | INT_STAT_VIP1_MULT_ANCA_SRC13 | INT_STAT_VIP1_MULT_ANCA_SRC12 | INT_STAT_VIP1_MULT_ANCA_SRC11 | INT_STAT_VIP1_MULT_ANCA_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP1_MULT_ANCA_SRC9 | INT_STAT_VIP1_MULT_ANCA_SRC8 | INT_STAT_VIP1_MULT_ANCA_SRC7 | INT_STAT_VIP1_MULT_ANCA_SRC6 | INT_STAT_VIP1_MULT_ANCA_SRC5 | INT_STAT_VIP1_MULT_ANCA_SRC4 | INT_STAT_VIP1_MULT_ANCA_SRC3 | INT_STAT_VIP1_MULT_ANCA_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_MULT_ANCA_SRC1 | INT_STAT_VIP1_MULT_ANCA_SRC0 | INT_STAT_VIP1_MULT_PORTB_SRC15 | INT_STAT_VIP1_MULT_PORTB_SRC14 | INT_STAT_VIP1_MULT_PORTB_SRC13 | INT_STAT_VIP1_MULT_PORTB_SRC12 | INT_STAT_VIP1_MULT_PORTB_SRC11 | INT_STAT_VIP1_MULT_PORTB_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-225. VPDMA_int0_channel2_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP1_MULT_ANCB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP1_MULT_ANCB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP1_MULT_ANCB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP1_MULT_ANCB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_MULT_ANCB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_MULT_ANCB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-225. VPDMA_int0_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP1_MULT_ANCB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP1_MULT_ANCB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP1_MULT_ANCB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP1_MULT_ANCB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP1_MULT_ANCA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP1_MULT_ANCA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP1_MULT_ANCA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP1_MULT_ANCA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP1_MULT_ANCA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP1_MULT_ANCA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-225. VPDMA_int0_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP1_MULT_ANCA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP1_MULT_ANCA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP1_MULT_ANCA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP1_MULT_ANCA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_MULT_ANCA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_MULT_ANCA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_MULT_ANCA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_MULT_ANCA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_MULT_ANCA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_MULT_ANCA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-225. VPDMA_int0_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_MULT_PORTB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_MULT_PORTB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_MULT_PORTB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_MULT_PORTB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP1_MULT_PORTB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP1_MULT_PORTB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.18 VPDMA_int0_channel2_int_mask Register (offset = 54h) [reset = 0h]

VPDMA_int0_channel2_int_mask is shown in Figure 12-313 and described in Table 12-226.

**Figure 12-313. VPDMA_int0_channel2_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_ANCB_SRC9 | INT_MASK_VIP1_MULT_ANCB_SRC8 | INT_MASK_VIP1_MULT_ANCB_SRC7 | INT_MASK_VIP1_MULT_ANCB_SRC6 | INT_MASK_VIP1_MULT_ANCB_SRC5 | INT_MASK_VIP1_MULT_ANCB_SRC4 | INT_MASK_VIP1_MULT_ANCB_SRC3 | INT_MASK_VIP1_MULT_ANCB_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP1_MULT_ANCB_SRC1 | INT_MASK_VIP1_MULT_ANCB_SRC0 | INT_MASK_VIP1_MULT_ANCA_SRC15 | INT_MASK_VIP1_MULT_ANCA_SRC14 | INT_MASK_VIP1_MULT_ANCA_SRC13 | INT_MASK_VIP1_MULT_ANCA_SRC12 | INT_MASK_VIP1_MULT_ANCA_SRC11 | INT_MASK_VIP1_MULT_ANCA_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP1_MULT_ANCA_SRC9 | INT_MASK_VIP1_MULT_ANCA_SRC8 | INT_MASK_VIP1_MULT_ANCA_SRC7 | INT_MASK_VIP1_MULT_ANCA_SRC6 | INT_MASK_VIP1_MULT_ANCA_SRC5 | INT_MASK_VIP1_MULT_ANCA_SRC4 | INT_MASK_VIP1_MULT_ANCA_SRC3 | INT_MASK_VIP1_MULT_ANCA_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP1_MULT_ANCA_SRC1 | INT_MASK_VIP1_MULT_ANCA_SRC0 | INT_MASK_VIP1_MULT_PORTB_SRC15 | INT_MASK_VIP1_MULT_PORTB_SRC14 | INT_MASK_VIP1_MULT_PORTB_SRC13 | INT_MASK_VIP1_MULT_PORTB_SRC12 | INT_MASK_VIP1_MULT_PORTB_SRC11 | INT_MASK_VIP1_MULT_PORTB_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-226. VPDMA_int0_channel2_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP1_MULT_ANCB_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP1_MULT_ANCB_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP1_MULT_ANCB_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP1_MULT_ANCB_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_MULT_ANCB_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_MULT_ANCB_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP1_MULT_ANCB_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP1_MULT_ANCB_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP1_MULT_ANCB_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP1_MULT_ANCB_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP1_MULT_ANCA_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-226. VPDMA_int0_channel2_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP1_MULT_ANCA_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP1_MULT_ANCA_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP1_MULT_ANCA_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP1_MULT_ANCA_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP1_MULT_ANCA_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP1_MULT_ANCA_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP1_MULT_ANCA_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP1_MULT_ANCA_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP1_MULT_ANCA_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_MULT_ANCA_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_MULT_ANCA_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_MULT_ANCA_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_MULT_ANCA_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_MULT_ANCA_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_MULT_ANCA_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_MULT_PORTB_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_MULT_PORTB_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_MULT_PORTB_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-226. VPDMA_int0_channel2_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP1_MULT_PORTB_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP1_MULT_PORTB_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP1_MULT_PORTB_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.19 VPDMA_int0_channel3_int_stat Register (offset = 58h) [reset = 0h]

VPDMA_int0_channel3_int_stat is shown in Figure 12-314 and described in Table 12-227.

**Figure 12-314. VPDMA_int0_channel3_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP2_MULT_PORTB_SRC3 | INT_STAT_VIP2_MULT_PORTB_SRC2 | INT_STAT_VIP2_MULT_PORTB_SRC1 | INT_STAT_VIP2_MULT_PORTB_SRC0 | INT_STAT_VIP2_MULT_PORTA_SRC15 | INT_STAT_VIP2_MULT_PORTA_SRC14 | INT_STAT_VIP2_MULT_PORTA_SRC13 | INT_STAT_VIP2_MULT_PORTA_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP2_MULT_PORTA_SRC11 | INT_STAT_VIP2_MULT_PORTA_SRC10 | INT_STAT_VIP2_MULT_PORTA_SRC9 | INT_STAT_VIP2_MULT_PORTA_SRC8 | INT_STAT_VIP2_MULT_PORTA_SRC7 | INT_STAT_VIP2_MULT_PORTA_SRC6 | INT_STAT_VIP2_MULT_PORTA_SRC5 | INT_STAT_VIP2_MULT_PORTA_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_MULT_PORTA_SRC3 | INT_STAT_VIP2_MULT_PORTA_SRC2 | INT_STAT_VIP2_MULT_PORTA_SRC1 | INT_STAT_VIP2_MULT_PORTA_SRC0 | INT_STAT_VIP1_PORTB_RGB | INT_STAT_VIP1_PORTA_RGB | INT_STAT_VIP1_PORTB_CHROMA | INT_STAT_VIP1_PORTB_LUMA |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_PORTA_CHROMA | INT_STAT_VIP1_PORTA_LUMA | INT_STAT_VIP1_MULT_ANCB_SRC15 | INT_STAT_VIP1_MULT_ANCB_SRC14 | INT_STAT_VIP1_MULT_ANCB_SRC13 | INT_STAT_VIP1_MULT_ANCB_SRC12 | INT_STAT_VIP1_MULT_ANCB_SRC11 | INT_STAT_VIP1_MULT_ANCB_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-227. VPDMA_int0_channel3_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP2_MULT_PORTB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP2_MULT_PORTB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP2_MULT_PORTB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_MULT_PORTB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP2_MULT_PORTA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP2_MULT_PORTA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-227. VPDMA_int0_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP2_MULT_PORTA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP2_MULT_PORTA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP2_MULT_PORTA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP2_MULT_PORTA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP2_MULT_PORTA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP2_MULT_PORTA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP2_MULT_PORTA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP2_MULT_PORTA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_MULT_PORTA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP2_MULT_PORTA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-227. VPDMA_int0_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP2_MULT_PORTA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_MULT_PORTA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_MULT_PORTA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_MULT_PORTA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_PORTB_RGB | W | 0h | The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_PORTA_RGB | W | 0h | The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_PORTB_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_PORTB_LUMA | W | 0h | The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_PORTA_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_PORTA_LUMA | W | 0h | The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-227. VPDMA_int0_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_MULT_ANCB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_MULT_ANCB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_MULT_ANCB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_MULT_ANCB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP1_MULT_ANCB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP1_MULT_ANCB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.20 VPDMA_int0_channel3_int_mask Register (offset = 5Ch) [reset = 0h]

VPDMA_int0_channel3_int_mask is shown in Figure 12-315 and described in Table 12-228.

**Figure 12-315. VPDMA_int0_channel3_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP2_MULT_PORTB_SRC3 | INT_MASK_VIP2_MULT_PORTB_SRC2 | INT_MASK_VIP2_MULT_PORTB_SRC1 | INT_MASK_VIP2_MULT_PORTB_SRC0 | INT_MASK_VIP2_MULT_PORTA_SRC15 | INT_MASK_VIP2_MULT_PORTA_SRC14 | INT_MASK_VIP2_MULT_PORTA_SRC13 | INT_MASK_VIP2_MULT_PORTA_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP2_MULT_PORTA_SRC11 | INT_MASK_VIP2_MULT_PORTA_SRC10 | INT_MASK_VIP2_MULT_PORTA_SRC9 | INT_MASK_VIP2_MULT_PORTA_SRC8 | INT_MASK_VIP2_MULT_PORTA_SRC7 | INT_MASK_VIP2_MULT_PORTA_SRC6 | INT_MASK_VIP2_MULT_PORTA_SRC5 | INT_MASK_VIP2_MULT_PORTA_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_MULT_PORTA_SRC3 | INT_MASK_VIP2_MULT_PORTA_SRC2 | INT_MASK_VIP2_MULT_PORTA_SRC1 | INT_MASK_VIP2_MULT_PORTA_SRC0 | INT_MASK_VIP1_PORTB_RGB | INT_MASK_VIP1_PORTA_RGB | INT_MASK_VIP1_PORTB_CHROMA | INT_MASK_VIP1_PORTB_LUMA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP1_PORTA_CHROMA | INT_MASK_VIP1_PORTA_LUMA | INT_MASK_VIP1_MULT_ANCB_SRC15 | INT_MASK_VIP1_MULT_ANCB_SRC14 | INT_MASK_VIP1_MULT_ANCB_SRC13 | INT_MASK_VIP1_MULT_ANCB_SRC12 | INT_MASK_VIP1_MULT_ANCB_SRC11 | INT_MASK_VIP1_MULT_ANCB_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-228. VPDMA_int0_channel3_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP2_MULT_PORTB_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP2_MULT_PORTB_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP2_MULT_PORTB_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_MULT_PORTB_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP2_MULT_PORTA_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP2_MULT_PORTA_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP2_MULT_PORTA_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP2_MULT_PORTA_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP2_MULT_PORTA_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP2_MULT_PORTA_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP2_MULT_PORTA_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-228. VPDMA_int0_channel3_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP2_MULT_PORTA_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP2_MULT_PORTA_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP2_MULT_PORTA_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_MULT_PORTA_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_MULT_PORTA_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_MULT_PORTA_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_MULT_PORTA_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_MULT_PORTA_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_MULT_PORTA_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_PORTB_RGB | R/W | 0h | The interrupt for Video Input 1 Port B RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_PORTA_RGB | R/W | 0h | The interrupt for Video Input 1 Port A RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_PORTB_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_PORTB_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_PORTA_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_PORTA_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_MULT_ANCB_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_MULT_ANCB_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_MULT_ANCB_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-228. VPDMA_int0_channel3_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP1_MULT_ANCB_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP1_MULT_ANCB_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP1_MULT_ANCB_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.21 VPDMA_int0_channel4_int_stat Register (offset = 60h) [reset = 0h]

VPDMA_int0_channel4_int_stat is shown in Figure 12-316 and described in Table 12-229.

**Figure 12-316. VPDMA_int0_channel4_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP2_MULT_ANCB_SRC3 | INT_STAT_VIP2_MULT_ANCB_SRC2 | INT_STAT_VIP2_MULT_ANCB_SRC1 | INT_STAT_VIP2_MULT_ANCB_SRC0 | INT_STAT_VIP2_MULT_ANCA_SRC15 | INT_STAT_VIP2_MULT_ANCA_SRC14 | INT_STAT_VIP2_MULT_ANCA_SRC13 | INT_STAT_VIP2_MULT_ANCA_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP2_MULT_ANCA_SRC11 | INT_STAT_VIP2_MULT_ANCA_SRC10 | INT_STAT_VIP2_MULT_ANCA_SRC9 | INT_STAT_VIP2_MULT_ANCA_SRC8 | INT_STAT_VIP2_MULT_ANCA_SRC7 | INT_STAT_VIP2_MULT_ANCA_SRC6 | INT_STAT_VIP2_MULT_ANCA_SRC5 | INT_STAT_VIP2_MULT_ANCA_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_MULT_ANCA_SRC3 | INT_STAT_VIP2_MULT_ANCA_SRC2 | INT_STAT_VIP2_MULT_ANCA_SRC1 | INT_STAT_VIP2_MULT_ANCA_SRC0 | INT_STAT_VIP2_MULT_PORTB_SRC15 | INT_STAT_VIP2_MULT_PORTB_SRC14 | INT_STAT_VIP2_MULT_PORTB_SRC13 | INT_STAT_VIP2_MULT_PORTB_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_MULT_PORTB_SRC11 | INT_STAT_VIP2_MULT_PORTB_SRC10 | INT_STAT_VIP2_MULT_PORTB_SRC9 | INT_STAT_VIP2_MULT_PORTB_SRC8 | INT_STAT_VIP2_MULT_PORTB_SRC7 | INT_STAT_VIP2_MULT_PORTB_SRC6 | INT_STAT_VIP2_MULT_PORTB_SRC5 | INT_STAT_VIP2_MULT_PORTB_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-229. VPDMA_int0_channel4_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP2_MULT_ANCB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP2_MULT_ANCB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP2_MULT_ANCB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_MULT_ANCB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP2_MULT_ANCA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP2_MULT_ANCA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-229. VPDMA_int0_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP2_MULT_ANCA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP2_MULT_ANCA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP2_MULT_ANCA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP2_MULT_ANCA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP2_MULT_ANCA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP2_MULT_ANCA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP2_MULT_ANCA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP2_MULT_ANCA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_MULT_ANCA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP2_MULT_ANCA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-229. VPDMA_int0_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP2_MULT_ANCA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_MULT_ANCA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_MULT_ANCA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_MULT_ANCA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP2_MULT_PORTB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP2_MULT_PORTB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_MULT_PORTB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_MULT_PORTB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_MULT_PORTB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP2_MULT_PORTB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-229. VPDMA_int0_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP2_MULT_PORTB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP2_MULT_PORTB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP2_MULT_PORTB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP2_MULT_PORTB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP2_MULT_PORTB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP2_MULT_PORTB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.22 VPDMA_int0_channel4_int_mask Register (offset = 64h) [reset = 0h]

VPDMA_int0_channel4_int_mask is shown in Figure 12-317 and described in Table 12-230.

**Figure 12-317. VPDMA_int0_channel4_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP2_MULT_ANCB_SRC3 | INT_MASK_VIP2_MULT_ANCB_SRC2 | INT_MASK_VIP2_MULT_ANCB_SRC1 | INT_MASK_VIP2_MULT_ANCB_SRC0 | INT_MASK_VIP2_MULT_ANCA_SRC15 | INT_MASK_VIP2_MULT_ANCA_SRC14 | INT_MASK_VIP2_MULT_ANCA_SRC13 | INT_MASK_VIP2_MULT_ANCA_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP2_MULT_ANCA_SRC11 | INT_MASK_VIP2_MULT_ANCA_SRC10 | INT_MASK_VIP2_MULT_ANCA_SRC9 | INT_MASK_VIP2_MULT_ANCA_SRC8 | INT_MASK_VIP2_MULT_ANCA_SRC7 | INT_MASK_VIP2_MULT_ANCA_SRC6 | INT_MASK_VIP2_MULT_ANCA_SRC5 | INT_MASK_VIP2_MULT_ANCA_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_MULT_ANCA_SRC3 | INT_MASK_VIP2_MULT_ANCA_SRC2 | INT_MASK_VIP2_MULT_ANCA_SRC1 | INT_MASK_VIP2_MULT_ANCA_SRC0 | INT_MASK_VIP2_MULT_PORTB_SRC15 | INT_MASK_VIP2_MULT_PORTB_SRC14 | INT_MASK_VIP2_MULT_PORTB_SRC13 | INT_MASK_VIP2_MULT_PORTB_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_MULT_PORTB_SRC11 | INT_MASK_VIP2_MULT_PORTB_SRC10 | INT_MASK_VIP2_MULT_PORTB_SRC9 | INT_MASK_VIP2_MULT_PORTB_SRC8 | INT_MASK_VIP2_MULT_PORTB_SRC7 | INT_MASK_VIP2_MULT_PORTB_SRC6 | INT_MASK_VIP2_MULT_PORTB_SRC5 | INT_MASK_VIP2_MULT_PORTB_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-230. VPDMA_int0_channel4_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP2_MULT_ANCB_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP2_MULT_ANCB_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP2_MULT_ANCB_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_MULT_ANCB_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP2_MULT_ANCA_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP2_MULT_ANCA_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP2_MULT_ANCA_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP2_MULT_ANCA_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP2_MULT_ANCA_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP2_MULT_ANCA_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP2_MULT_ANCA_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-230. VPDMA_int0_channel4_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP2_MULT_ANCA_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP2_MULT_ANCA_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP2_MULT_ANCA_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_MULT_ANCA_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_MULT_ANCA_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_MULT_ANCA_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_MULT_ANCA_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_MULT_ANCA_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_MULT_ANCA_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP2_MULT_PORTB_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP2_MULT_PORTB_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_MULT_PORTB_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_MULT_PORTB_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_MULT_PORTB_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_MULT_PORTB_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP2_MULT_PORTB_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP2_MULT_PORTB_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP2_MULT_PORTB_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-230. VPDMA_int0_channel4_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP2_MULT_PORTB_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP2_MULT_PORTB_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP2_MULT_PORTB_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.23 VPDMA_int0_channel5_int_stat Register (offset = 68h) [reset = 0h]

VPDMA_int0_channel5_int_stat is shown in Figure 12-318 and described in Table 12-231.

**Figure 12-318. VPDMA_int0_channel5_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_TRANSCODE2_CHROMA | INT_STAT_TRANSCODE2_LUMA | INT_STAT_TRANSCODE1_CHROMA | INT_STAT_TRANSCODE1_LUMA | INT_STAT_AUX_IN | INT_STAT_PIP_FRAME | INT_STAT_POST_COMP_WR | INT_STAT_VBI_SD_VENC |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | INT_STAT_NF_LAST_CHROMA | INT_STAT_NF_LAST_LUMA | INT_STAT_NF_WRITE_CHROMA | INT_STAT_NF_WRITE_LUMA | INT_STAT_NF_READ | INT_STAT_VIP2_PORTB_RGB | INT_STAT_VIP2_PORTA_RGB |
| R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_PORTB_CHROMA | INT_STAT_VIP2_PORTB_LUMA | INT_STAT_VIP2_PORTA_CHROMA | INT_STAT_VIP2_PORTA_LUMA | INT_STAT_VIP2_MULT_ANCB_SRC15 | INT_STAT_VIP2_MULT_ANCB_SRC14 | INT_STAT_VIP2_MULT_ANCB_SRC13 | INT_STAT_VIP2_MULT_ANCB_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_MULT_ANCB_SRC11 | INT_STAT_VIP2_MULT_ANCB_SRC10 | INT_STAT_VIP2_MULT_ANCB_SRC9 | INT_STAT_VIP2_MULT_ANCB_SRC8 | INT_STAT_VIP2_MULT_ANCB_SRC7 | INT_STAT_VIP2_MULT_ANCB_SRC6 | INT_STAT_VIP2_MULT_ANCB_SRC5 | INT_STAT_VIP2_MULT_ANCB_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-231. VPDMA_int0_channel5_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_TRANSCODE2_CHROMA | W | 0h | The last read DMA transaction has occurred for channel transcode2_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans2_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_TRANSCODE2_LUMA | W | 0h | The last read DMA transaction has occurred for channel transcode2_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans2_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_TRANSCODE1_CHROMA | W | 0h | The last read DMA transaction has occurred for channel transcode1_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans1_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_TRANSCODE1_LUMA | W | 0h | The last read DMA transaction has occurred for channel transcode1_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans1_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_AUX_IN | W | 0h | The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-231. VPDMA_int0_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 26 | INT_STAT_PIP_FRAME | W | 0h | The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_POST_COMP_WR | W | 0h | The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VBI_SD_VENC | W | 0h | The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | Reserved | R | 0h | |
| 22 | INT_STAT_NF_LAST_CHROMA | W | 0h | The last read DMA transaction has occurred for channel nf_last_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_420_uv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_NF_LAST_LUMA | W | 0h | The last read DMA transaction has occurred for channel nf_last_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_420_y_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_NF_WRITE_CHROMA | W | 0h | The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_NF_WRITE_LUMA | W | 0h | The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_NF_READ | W | 0h | The last read DMA transaction has occurred for channel nf_read and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_422_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_PORTB_RGB | W | 0h | The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-231. VPDMA_int0_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | INT_STAT_VIP2_PORTA_RGB | W | 0h | The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_VIP2_PORTB_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_PORTB_LUMA | W | 0h | The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_PORTA_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_PORTA_LUMA | W | 0h | The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP2_MULT_ANCB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP2_MULT_ANCB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_MULT_ANCB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_MULT_ANCB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_MULT_ANCB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-231. VPDMA_int0_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | INT_STAT_VIP2_MULT_ANCB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 5 | INT_STAT_VIP2_MULT_ANCB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP2_MULT_ANCB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP2_MULT_ANCB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP2_MULT_ANCB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP2_MULT_ANCB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP2_MULT_ANCB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.24 VPDMA_int0_channel5_int_mask Register (offset = 6Ch) [reset = 0h]

VPDMA_int0_channel5_int_mask is shown in Figure 12-319 and described in Table 12-232.

**Figure 12-319. VPDMA_int0_channel5_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_TRANSCODE2_CHROMA | INT_MASK_TRANSCODE2_LUMA | INT_MASK_TRANSCODE1_CHROMA | INT_MASK_TRANSCODE1_LUMA | INT_MASK_AUX_IN | INT_MASK_PIP_FRAME | INT_MASK_POST_COMP_WR | INT_MASK_VBI_SD_VENC |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | INT_MASK_NF_LAST_CHROMA | INT_MASK_NF_LAST_LUMA | INT_MASK_NF_WRITE_CHROMA | INT_MASK_NF_WRITE_LUMA | INT_MASK_NF_READ | INT_MASK_VIP2_PORTB_RGB | INT_MASK_VIP2_PORTA_RGB |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_PORTB_CHROMA | INT_MASK_VIP2_PORTB_LUMA | INT_MASK_VIP2_PORTA_CHROMA | INT_MASK_VIP2_PORTA_LUMA | INT_MASK_VIP2_MULT_ANCB_SRC15 | INT_MASK_VIP2_MULT_ANCB_SRC14 | INT_MASK_VIP2_MULT_ANCB_SRC13 | INT_MASK_VIP2_MULT_ANCB_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_MULT_ANCB_SRC11 | INT_MASK_VIP2_MULT_ANCB_SRC10 | INT_MASK_VIP2_MULT_ANCB_SRC9 | INT_MASK_VIP2_MULT_ANCB_SRC8 | INT_MASK_VIP2_MULT_ANCB_SRC7 | INT_MASK_VIP2_MULT_ANCB_SRC6 | INT_MASK_VIP2_MULT_ANCB_SRC5 | INT_MASK_VIP2_MULT_ANCB_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-232. VPDMA_int0_channel5_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_TRANSCODE2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_TRANSCODE2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_TRANSCODE1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_TRANSCODE1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_AUX_IN | R/W | 0h | The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_PIP_FRAME | R/W | 0h | The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_POST_COMP_WR | R/W | 0h | The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VBI_SD_VENC | R/W | 0h | The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | Reserved | R | 0h | |
| 22 | INT_MASK_NF_LAST_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_NF_LAST_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-232. VPDMA_int0_channel5_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_NF_WRITE_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_NF_WRITE_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_NF_READ | R/W | 0h | The interrupt for Noise Filter Input Data 422 Interleaved should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_PORTB_RGB | R/W | 0h | The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_PORTA_RGB | R/W | 0h | The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_PORTB_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_PORTB_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_PORTA_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_PORTA_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP2_MULT_ANCB_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP2_MULT_ANCB_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_MULT_ANCB_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_MULT_ANCB_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_MULT_ANCB_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_MULT_ANCB_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP2_MULT_ANCB_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP2_MULT_ANCB_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP2_MULT_ANCB_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-232. VPDMA_int0_channel5_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP2_MULT_ANCB_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP2_MULT_ANCB_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP2_MULT_ANCB_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.25 VPDMA_int0_channel6_int_stat Register (offset = 70h) [reset = 0h]

VPDMA_int0_channel6_int_stat is shown in Figure 12-320 and described in Table 12-233.

**Figure 12-320. VPDMA_int0_channel6_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | INT_STAT_OTHER |
| R-0h | | | | | | | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-233. VPDMA_int0_channel6_int_stat Register Field Descriptions**

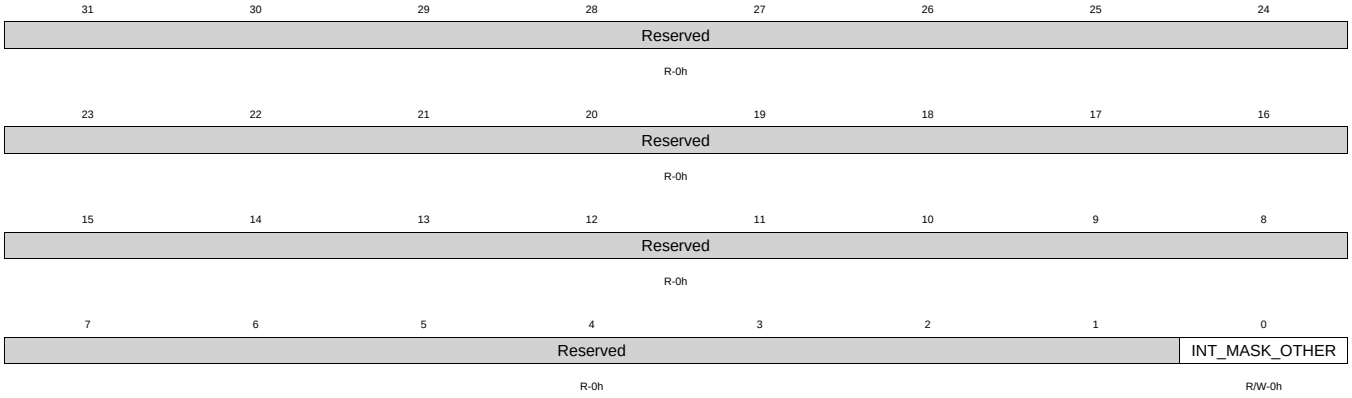
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | INT_STAT_OTHER | W | 0h | This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.26 VPDMA_int0_channel6_int_mask Register (offset = 74h) [reset = 0h]

VPDMA_int0_channel6_int_mask is shown in Figure 12-321 and described in Table 12-234.

**Figure 12-321. VPDMA_int0_channel6_int_mask Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-234. VPDMA_int0_channel6_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | INT_MASK_OTHER | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.27 VPDMA_int0_client0_int_stat Register (offset = 78h) [reset = 0h]

VPDMA_int0_client0_int_stat is shown in Figure 12-322 and described in Table 12-235.

**Figure 12-322. VPDMA_int0_client0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_GRPX1_DATA | INT_STAT_COMP_WRBK | INT_STAT_SC_OUT | Reserved | | | | |
| W-0h | W-0h | W-0h | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | INT_STAT_SC_IN_LUMA | INT_STAT_SC_IN_CHROMA | INT_STAT_PIP_WRBK | INT_STAT_DEI_SC_OUT | Reserved |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_DEI_HQ_MV_OUT | Reserved | | INT_STAT_DEI_HQ_MV_IN | Reserved | | | |
| W-0h | R-0h | | W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_STAT_DEI_HQ_3_CHROMA | INT_STAT_DEI_HQ_3_LUMA | INT_STAT_DEI_HQ_2_CHROMA | INT_STAT_DEI_HQ_2_LUMA | INT_STAT_DEI_HQ_1_LUMA | INT_STAT_DEI_HQ_1_CHROMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-235. VPDMA_int0_client0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_GRPX1_DATA | W | 0h | The client interface grpx1_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_COMP_WRBK | W | 0h | The client interface comp_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_SC_OUT | W | 0h | The client interface sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28-21 | Reserved | R | 0h | |
| 20 | INT_STAT_SC_IN_LUMA | W | 0h | The client interface sc_in_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_SC_IN_CHROMA | W | 0h | The client interface sc_in_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-235. VPDMA_int0_client0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | INT_STAT_PIP_WRBK | W | 0h | The client interface pip_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_DEI_SC_OUT | W | 0h | The client interface dei_sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | Reserved | R | 0h | |
| 15 | INT_STAT_DEI_HQ_MV_OUT | W | 0h | The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_STAT_DEI_HQ_MV_IN | W | 0h | The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11-6 | Reserved | R | 0h | |
| 5 | INT_STAT_DEI_HQ_3_CHROMA | W | 0h | The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_DEI_HQ_3_LUMA | W | 0h | The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_DEI_HQ_2_CHROMA | W | 0h | The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_DEI_HQ_2_LUMA | W | 0h | The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-235. VPDMA_int0_client0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | INT_STAT_DEI_HQ_1_LUMA | W | 0h | The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_DEI_HQ_1_CHROMA | W | 0h | The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.28 VPDMA_int0_client0_int_mask Register (offset = 7Ch) [reset = 0h]

VPDMA_int0_client0_int_mask is shown in Figure 12-323 and described in Table 12-236.

**Figure 12-323. VPDMA_int0_client0_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_GRPX1_DATA | INT_MASK_COMP_WRBK | INT_MASK_SC_OUT | Reserved | | | | |
| R/W-0h | R/W-0h | R/W-0h | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | INT_MASK_SC_IN_LUMA | INT_MASK_SC_IN_CHROMA | INT_MASK_PIP_WRBK | INT_MASK_DEI_SC_OUT | Reserved |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_DEI_HQ_MV_OUT | Reserved | | INT_MASK_DEI_HQ_MV_IN | Reserved | | | |
| R/W-0h | R-0h | | R/W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_MASK_DEI_HQ_3_CHROMA | INT_MASK_DEI_HQ_3_LUMA | INT_MASK_DEI_HQ_2_CHROMA | INT_MASK_DEI_HQ_2_LUMA | INT_MASK_DEI_HQ_1_LUMA | INT_MASK_DEI_HQ_1_CHROMA |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-236. VPDMA_int0_client0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_GRPX1_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_COMP_WRBK | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_SC_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28-21 | Reserved | R | 0h | |
| 20 | INT_MASK_SC_IN_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_SC_IN_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_PIP_WRBK | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_DEI_SC_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | Reserved | R | 0h | |
| 15 | INT_MASK_DEI_HQ_MV_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_MASK_DEI_HQ_MV_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11-6 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-236. VPDMA_int0_client0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_MASK_DEI_HQ_3_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_DEI_HQ_3_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_DEI_HQ_2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_DEI_HQ_2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_DEI_HQ_1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_DEI_HQ_1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.29 VPDMA_int0_client1_int_stat Register (offset = 80h) [reset = 0h]

VPDMA_int0_client1_int_stat is shown in Figure 12-324 and described in Table 12-237.

**Figure 12-324. VPDMA_int0_client1_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | INT_STAT_VIP2_ANC_B | INT_STAT_VIP2_ANC_A | INT_STAT_VIP1_ANC_B | INT_STAT_VIP1_ANC_A | INT_STAT_TRANS2_LUMA | INT_STAT_TRANS2_CHROMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_TRANS1_LUMA | INT_STAT_TRANS1_CHROMA | INT_STAT_HDMI_WRBK_OUT | INT_STAT_VPI_CTL | INT_STAT_VBI_SDVENC | Reserved | INT_STAT_NF_420_UV_OUT | INT_STAT_NF_420_Y_OUT |
| W-0h | W-0h | W-0h | W-0h | W-0h | R-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_NF_420_UV_IN | INT_STAT_NF_420_Y_IN | INT_STAT_NF_422_IN | INT_STAT_GRPX3_ST | INT_STAT_GRPX2_ST | INT_STAT_GRPX1_ST | INT_STAT_VIP2_UP_UV | INT_STAT_VIP2_UP_Y |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_LO_UV | INT_STAT_VIP2_LO_Y | INT_STAT_VIP1_UP_UV | INT_STAT_VIP1_UP_Y | INT_STAT_VIP1_LO_UV | INT_STAT_VIP1_LO_Y | INT_STAT_GRPX3_DATA | INT_STAT_GRPX2_DATA |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-237. VPDMA_int0_client1_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29 | INT_STAT_VIP2_ANC_B | W | 0h | The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_ANC_A | W | 0h | The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_ANC_B | W | 0h | The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_ANC_A | W | 0h | The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_TRANS2_LUMA | W | 0h | The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-237. VPDMA_int0_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 24 | INT_STAT_TRANS2_CHROMA | W | 0h | The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_TRANS1_LUMA | W | 0h | The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_TRANS1_CHROMA | W | 0h | The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_HDMI_WRBK_OUT | W | 0h | The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VPI_CTL | W | 0h | The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VBI_SDVENC | W | 0h | The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | Reserved | R | 0h | |
| 17 | INT_STAT_NF_420_UV_OUT | W | 0h | The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_NF_420_Y_OUT | W | 0h | The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_NF_420_UV_IN | W | 0h | The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-237. VPDMA_int0_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | INT_STAT_NF_420_Y_IN | W | 0h | The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_NF_422_IN | W | 0h | The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_GRPX3_ST | W | 0h | The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_GRPX2_ST | W | 0h | The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_GRPX1_ST | W | 0h | The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_UP_UV | W | 0h | The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_UP_Y | W | 0h | The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_LO_UV | W | 0h | The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP2_LO_Y | W | 0h | The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-237. VPDMA_int0_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_UP_UV | W | 0h | The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_UP_Y | W | 0h | The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_LO_UV | W | 0h | The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_LO_Y | W | 0h | The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_GRPX3_DATA | W | 0h | The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_GRPX2_DATA | W | 0h | The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.30 VPDMA_int0_client1_int_mask Register (offset = 84h) [reset = 0h]

VPDMA_int0_client1_int_mask is shown in Figure 12-325 and described in Table 12-238.

**Figure 12-325. VPDMA_int0_client1_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | INT_MASK_VIP2_ANC_B | INT_MASK_VIP2_ANC_A | INT_MASK_VIP1_ANC_B | INT_MASK_VIP1_ANC_A | INT_MASK_TRANS2_LUMA | INT_MASK_TRANS2_CHROMA |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_TRANS1_LUMA | INT_MASK_TRANS1_CHROMA | INT_MASK_HDMI_WRBK_OUT | INT_MASK_VPI_CTL | INT_MASK_VBI_SDVENC | Reserved | INT_MASK_NF_420_UV_OUT | INT_MASK_NF_420_Y_OUT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_NF_420_UV_IN | INT_MASK_NF_420_Y_IN | INT_MASK_NF_422_IN | INT_MASK_GRPX3_ST | INT_MASK_GRPX2_ST | INT_MASK_GRPX1_ST | INT_MASK_VIP2_UP_UV | INT_MASK_VIP2_UP_Y |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_LO_UV | INT_MASK_VIP2_LO_Y | INT_MASK_VIP1_UP_UV | INT_MASK_VIP1_UP_Y | INT_MASK_VIP1_LO_UV | INT_MASK_VIP1_LO_Y | INT_MASK_GRPX3_DATA | INT_MASK_GRPX2_DATA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-238. VPDMA_int0_client1_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29 | INT_MASK_VIP2_ANC_B | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_ANC_A | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_ANC_B | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_ANC_A | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_TRANS2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_TRANS2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_TRANS1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_TRANS1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_HDMI_WRBK_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 20 | INT_MASK_VPI_CTL | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-238. VPDMA_int0_client1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | INT_MASK_VBI_SDVENC | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | Reserved | R | 0h | |
| 17 | INT_MASK_NF_420_UV_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_NF_420_Y_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_NF_420_UV_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_NF_420_Y_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_NF_422_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_GRPX3_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_GRPX2_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_GRPX1_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_UP_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_UP_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_LO_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_LO_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_UP_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_UP_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_LO_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_VIP1_LO_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_GRPX3_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-238. VPDMA_int0_client1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | INT_MASK_GRPX2_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.31 VPDMA_int0_list0_int_stat Register (offset = 88h) [reset = 0h]

VPDMA_int0_list0_int_stat is shown in Figure 12-326 and described in Table 12-239.

**Figure 12-326. VPDMA_int0_list0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_CONTROL_DESCRIPTOR_INT15 | INT_STAT_CONTROL_DESCRIPTOR_INT14 | INT_STAT_CONTROL_DESCRIPTOR_INT13 | INT_STAT_CONTROL_DESCRIPTOR_INT12 | INT_STAT_CONTROL_DESCRIPTOR_INT11 | INT_STAT_CONTROL_DESCRIPTOR_INT10 | INT_STAT_CONTROL_DESCRIPTOR_INT9 | INT_STAT_CONTROL_DESCRIPTOR_INT8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_CONTROL_DESCRIPTOR_INT7 | INT_STAT_CONTROL_DESCRIPTOR_INT6 | INT_STAT_CONTROL_DESCRIPTOR_INT5 | INT_STAT_CONTROL_DESCRIPTOR_INT4 | INT_STAT_CONTROL_DESCRIPTOR_INT3 | INT_STAT_CONTROL_DESCRIPTOR_INT2 | INT_STAT_CONTROL_DESCRIPTOR_INT1 | INT_STAT_CONTROL_DESCRIPTOR_INT0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_LIST7_NOTIFY | INT_STAT_LIST7_COMPLETE | INT_STAT_LIST6_NOTIFY | INT_STAT_LIST6_COMPLETE | INT_STAT_LIST5_NOTIFY | INT_STAT_LIST5_COMPLETE | INT_STAT_LIST4_NOTIFY | INT_STAT_LIST4_COMPLETE |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_LIST3_NOTIFY | INT_STAT_LIST3_COMPLETE | INT_STAT_LIST2_NOTIFY | INT_STAT_LIST2_COMPLETE | INT_STAT_LIST1_NOTIFY | INT_STAT_LIST1_COMPLETE | INT_STAT_LIST0_NOTIFY | INT_STAT_LIST0_COMPLETE |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-239. VPDMA_int0_list0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_CONTROL_DESCRIPTOR_INT15 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_CONTROL_DESCRIPTOR_INT14 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_CONTROL_DESCRIPTOR_INT13 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_CONTROL_DESCRIPTOR_INT12 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_CONTROL_DESCRIPTOR_INT11 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_CONTROL_DESCRIPTOR_INT10 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_CONTROL_DESCRIPTOR_INT9 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_CONTROL_DESCRIPTOR_INT8 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-239. VPDMA_int0_list0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | INT_STAT_CONTROL_DESCRIPTOR_INT7 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_CONTROL_DESCRIPTOR_INT6 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_CONTROL_DESCRIPTOR_INT5 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_CONTROL_DESCRIPTOR_INT4 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_CONTROL_DESCRIPTOR_INT3 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_CONTROL_DESCRIPTOR_INT2 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_CONTROL_DESCRIPTOR_INT1 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_CONTROL_DESCRIPTOR_INT0 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_LIST7_NOTIFY | W | 0h | A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_LIST7_COMPLETE | W | 0h | List 7 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_LIST6_NOTIFY | W | 0h | A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_LIST6_COMPLETE | W | 0h | List 6 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_LIST5_NOTIFY | W | 0h | A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_LIST5_COMPLETE | W | 0h | List 5 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_LIST4_NOTIFY | W | 0h | A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-239. VPDMA_int0_list0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | INT_STAT_LIST4_COMPLETE | W | 0h | List 4 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_LIST3_NOTIFY | W | 0h | A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_LIST3_COMPLETE | W | 0h | List 3 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 5 | INT_STAT_LIST2_NOTIFY | W | 0h | A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_LIST2_COMPLETE | W | 0h | List 2 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_LIST1_NOTIFY | W | 0h | A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_LIST1_COMPLETE | W | 0h | List 1 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_LIST0_NOTIFY | W | 0h | A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_LIST0_COMPLETE | W | 0h | List 0 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.32 VPDMA_int0_list0_int_mask Register (offset = 8Ch) [reset = 0h]

VPDMA_int0_list0_int_mask is shown in Figure 12-327 and described in Table 12-240.

**Figure 12-327. VPDMA_int0_list0_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_CONTROL_DESCRIPTOR_INT15 | INT_MASK_CONTROL_DESCRIPTOR_INT14 | INT_MASK_CONTROL_DESCRIPTOR_INT13 | INT_MASK_CONTROL_DESCRIPTOR_INT12 | INT_MASK_CONTROL_DESCRIPTOR_INT11 | INT_MASK_CONTROL_DESCRIPTOR_INT10 | INT_MASK_CONTROL_DESCRIPTOR_INT9 | INT_MASK_CONTROL_DESCRIPTOR_INT8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_CONTROL_DESCRIPTOR_INT7 | INT_MASK_CONTROL_DESCRIPTOR_INT6 | INT_MASK_CONTROL_DESCRIPTOR_INT5 | INT_MASK_CONTROL_DESCRIPTOR_INT4 | INT_MASK_CONTROL_DESCRIPTOR_INT3 | INT_MASK_CONTROL_DESCRIPTOR_INT2 | INT_MASK_CONTROL_DESCRIPTOR_INT1 | INT_MASK_CONTROL_DESCRIPTOR_INT0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_LIST7_NOTIFY | INT_MASK_LIST7_COMPLETE | INT_MASK_LIST6_NOTIFY | INT_MASK_LIST6_COMPLETE | INT_MASK_LIST5_NOTIFY | INT_MASK_LIST5_COMPLETE | INT_MASK_LIST4_NOTIFY | INT_MASK_LIST4_COMPLETE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_LIST3_NOTIFY | INT_MASK_LIST3_COMPLETE | INT_MASK_LIST2_NOTIFY | INT_MASK_LIST2_COMPLETE | INT_MASK_LIST1_NOTIFY | INT_MASK_LIST1_COMPLETE | INT_MASK_LIST0_NOTIFY | INT_MASK_LIST0_COMPLETE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-240. VPDMA_int0_list0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_CONTROL_DESCRIPTOR_INT15 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_CONTROL_DESCRIPTOR_INT14 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_CONTROL_DESCRIPTOR_INT13 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_CONTROL_DESCRIPTOR_INT12 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_CONTROL_DESCRIPTOR_INT11 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_CONTROL_DESCRIPTOR_INT10 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_CONTROL_DESCRIPTOR_INT9 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_CONTROL_DESCRIPTOR_INT8 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_CONTROL_DESCRIPTOR_INT7 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_CONTROL_DESCRIPTOR_INT6 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-240. VPDMA_int0_list0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | INT_MASK_CONTROL_DESCRIPTOR_INT5 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 20 | INT_MASK_CONTROL_DESCRIPTOR_INT4 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_CONTROL_DESCRIPTOR_INT3 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_CONTROL_DESCRIPTOR_INT2 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_CONTROL_DESCRIPTOR_INT1 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_CONTROL_DESCRIPTOR_INT0 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_LIST7_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_LIST7_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_LIST6_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_LIST6_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_LIST5_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_LIST5_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_LIST4_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_LIST4_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_LIST3_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_LIST3_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_LIST2_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_LIST2_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-240. VPDMA_int0_list0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | INT_MASK_LIST1_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_LIST1_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_LIST0_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_LIST0_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.33 VPDMA_int1_channel0_int_stat Register (offset = 90h) [reset = 0h]

VPDMA_int1_channel0_int_stat is shown in Figure 12-328 and described in Table 12-241.

**Figure 12-328. VPDMA_int1_channel0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_GRPX3 | INT_STAT_GRPX2 | INT_STAT_GRPX1 | INT_STAT_SCALER_OUT | Reserved | | | |
| W-0h | W-0h | W-0h | W-0h | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | INT_STAT_SCALER_CHROMA | INT_STAT_SCALER_LUMA | INT_STAT_HQ_SCALER | Reserved |
| R-0h | | | | W-0h | W-0h | W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_HQ_MV_OUT | Reserved | | INT_STAT_HQ_MV | Reserved | | | |
| W-0h | R-0h | | W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_STAT_HQ_VID3_CHROMA | INT_STAT_HQ_VID3_LUMA | INT_STAT_HQ_VID2_CHROMA | INT_STAT_HQ_VID2_LUMA | INT_STAT_HQ_VID1_CHROMA | INT_STAT_HQ_VID1_LUMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-241. VPDMA_int1_channel0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_GRPX3 | W | 0h | The last read DMA transaction has occurred for channel grpx3 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_GRPX2 | W | 0h | The last read DMA transaction has occurred for channel grpx2 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_GRPX1 | W | 0h | The last read DMA transaction has occurred for channel grpx1 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_SCALER_OUT | W | 0h | The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27-20 | Reserved | R | 0h | |
| 19 | INT_STAT_SCALER_CHROMA | W | 0h | The last write DMA transaction has completed for channel scaler_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-241. VPDMA_int1_channel0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | INT_STAT_SCALER_LUMA | W | 0h | The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_HQ_SCALER | W | 0h | The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | Reserved | R | 0h | |
| 15 | INT_STAT_HQ_MV_OUT | W | 0h | The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_STAT_HQ_MV | W | 0h | The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11-6 | Reserved | R | 0h | |
| 5 | INT_STAT_HQ_VID3_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid3_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_3_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_HQ_VID3_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid3_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_3_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_HQ_VID2_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid2_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_2_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_HQ_VID2_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid2_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_2_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-241. VPDMA_int1_channel0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | INT_STAT_HQ_VID1_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid1_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_1_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_HQ_VID1_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid1_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_1_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.34 VPDMA_int1_channel0_int_mask Register (offset = 94h) [reset = 0h]

VPDMA_int1_channel0_int_mask is shown in Figure 12-329 and described in Table 12-242.

**Figure 12-329. VPDMA_int1_channel0_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_GRPX3 | INT_MASK_GRPX2 | INT_MASK_GRPX1 | INT_MASK_SCALER_OUT | Reserved | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | INT_MASK_SCALER_CHROMA | INT_MASK_SCALER_LUMA | INT_MASK_HQ_SCALER | Reserved |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_HQ_MV_OUT | Reserved | | INT_MASK_HQ_MV | Reserved | | | |
| R/W-0h | R-0h | | R/W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_MASK_HQ_VID3_CHROMA | INT_MASK_HQ_VID3_LUMA | INT_MASK_HQ_VID2_CHROMA | INT_MASK_HQ_VID2_LUMA | INT_MASK_HQ_VID1_CHROMA | INT_MASK_HQ_VID1_LUMA |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-242. VPDMA_int1_channel0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_GRPX3 | R/W | 0h | The interrupt for Graphcis 2 Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_GRPX2 | R/W | 0h | The interrupt for Graphics 1 Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_GRPX1 | R/W | 0h | The interrupt for Graphics 0 Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_SCALER_OUT | R/W | 0h | The interrupt for Low Cost DEI Scalar Write to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27-20 | Reserved | R | 0h | |
| 19 | INT_MASK_SCALER_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_SCALER_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_HQ_SCALER | R/W | 0h | The interrupt for High Quality DEI Scaler Write to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | Reserved | R | 0h | |
| 15 | INT_MASK_HQ_MV_OUT | R/W | 0h | The interrupt for Low Cost DEI Motion Vector Write should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_MASK_HQ_MV | R/W | 0h | The interrupt for Low Cost DEI Motion Vector should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11-6 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-242. VPDMA_int1_channel0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_MASK_HQ_VID3_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_HQ_VID3_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_HQ_VID2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_HQ_VID2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_HQ_VID1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_HQ_VID1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.35 VPDMA_int1_channel1_int_stat Register (offset = 98h) [reset = 0h]

VPDMA_int1_channel1_int_stat is shown in Figure 12-330 and described in Table 12-243.

**Figure 12-330. VPDMA_int1_channel1_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP1_MULT_PORTB_SRC9 | INT_STAT_VIP1_MULT_PORTB_SRC8 | INT_STAT_VIP1_MULT_PORTB_SRC7 | INT_STAT_VIP1_MULT_PORTB_SRC6 | INT_STAT_VIP1_MULT_PORTB_SRC5 | INT_STAT_VIP1_MULT_PORTB_SRC4 | INT_STAT_VIP1_MULT_PORTB_SRC3 | INT_STAT_VIP1_MULT_PORTB_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP1_MULT_PORTB_SRC1 | INT_STAT_VIP1_MULT_PORTB_SRC0 | INT_STAT_VIP1_MULT_PORTA_SRC15 | INT_STAT_VIP1_MULT_PORTA_SRC14 | INT_STAT_VIP1_MULT_PORTA_SRC13 | INT_STAT_VIP1_MULT_PORTA_SRC12 | INT_STAT_VIP1_MULT_PORTA_SRC11 | INT_STAT_VIP1_MULT_PORTA_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP1_MULT_PORTA_SRC9 | INT_STAT_VIP1_MULT_PORTA_SRC8 | INT_STAT_VIP1_MULT_PORTA_SRC7 | INT_STAT_VIP1_MULT_PORTA_SRC6 | INT_STAT_VIP1_MULT_PORTA_SRC5 | INT_STAT_VIP1_MULT_PORTA_SRC4 | INT_STAT_VIP1_MULT_PORTA_SRC3 | INT_STAT_VIP1_MULT_PORTA_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_MULT_PORTA_SRC1 | INT_STAT_VIP1_MULT_PORTA_SRC0 | INT_STAT_GRPX3_CLUT | INT_STAT_GRPX2_CLUT | INT_STAT_GRPX1_CLUT | INT_STAT_GRPX3_STENCIL | INT_STAT_GRPX2_STENCIL | INT_STAT_GRPX1_STENCIL |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-243. VPDMA_int1_channel1_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP1_MULT_PORTB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP1_MULT_PORTB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP1_MULT_PORTB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP1_MULT_PORTB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_MULT_PORTB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_MULT_PORTB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-243. VPDMA_int1_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP1_MULT_PORTB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP1_MULT_PORTB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP1_MULT_PORTB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP1_MULT_PORTB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP1_MULT_PORTA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP1_MULT_PORTA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP1_MULT_PORTA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP1_MULT_PORTA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP1_MULT_PORTA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP1_MULT_PORTA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-243. VPDMA_int1_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP1_MULT_PORTA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP1_MULT_PORTA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP1_MULT_PORTA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP1_MULT_PORTA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_MULT_PORTA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_MULT_PORTA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_MULT_PORTA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_MULT_PORTA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_MULT_PORTA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_MULT_PORTA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-243. VPDMA_int1_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_GRPX3_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx3_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_GRPX2_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx2_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_GRPX1_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx1_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_GRPX3_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx3_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_GRPX2_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx2_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_GRPX1_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx1_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.36 VPDMA_int1_channel1_int_mask Register (offset = 9Ch) [reset = 0h]

VPDMA_int1_channel1_int_mask is shown in Figure 12-331 and described in Table 12-244.

**Figure 12-331. VPDMA_int1_channel1_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_PORTB_SRC9 | INT_MASK_VIP1_MULT_PORTB_SRC8 | INT_MASK_VIP1_MULT_PORTB_SRC7 | INT_MASK_VIP1_MULT_PORTB_SRC6 | INT_MASK_VIP1_MULT_PORTB_SRC5 | INT_MASK_VIP1_MULT_PORTB_SRC4 | INT_MASK_VIP1_MULT_PORTB_SRC3 | INT_MASK_VIP1_MULT_PORTB_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP1_MULT_PORTB_SRC1 | INT_MASK_VIP1_MULT_PORTB_SRC0 | INT_MASK_VIP1_MULT_PORTA_SRC15 | INT_MASK_VIP1_MULT_PORTA_SRC14 | INT_MASK_VIP1_MULT_PORTA_SRC13 | INT_MASK_VIP1_MULT_PORTA_SRC12 | INT_MASK_VIP1_MULT_PORTA_SRC11 | INT_MASK_VIP1_MULT_PORTA_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP1_MULT_PORTA_SRC9 | INT_MASK_VIP1_MULT_PORTA_SRC8 | INT_MASK_VIP1_MULT_PORTA_SRC7 | INT_MASK_VIP1_MULT_PORTA_SRC6 | INT_MASK_VIP1_MULT_PORTA_SRC5 | INT_MASK_VIP1_MULT_PORTA_SRC4 | INT_MASK_VIP1_MULT_PORTA_SRC3 | INT_MASK_VIP1_MULT_PORTA_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP1_MULT_PORTA_SRC1 | INT_MASK_VIP1_MULT_PORTA_SRC0 | INT_MASK_GRPX3_CLUT | INT_MASK_GRPX2_CLUT | INT_MASK_GRPX1_CLUT | INT_MASK_GRPX3_STENCIL | INT_MASK_GRPX2_STENCIL | INT_MASK_GRPX1_STENCIL |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-244. VPDMA_int1_channel1_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP1_MULT_PORTB_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP1_MULT_PORTB_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP1_MULT_PORTB_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP1_MULT_PORTB_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_MULT_PORTB_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_MULT_PORTB_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP1_MULT_PORTB_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP1_MULT_PORTB_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP1_MULT_PORTB_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP1_MULT_PORTB_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP1_MULT_PORTA_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-244. VPDMA_int1_channel1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP1_MULT_PORTA_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP1_MULT_PORTA_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP1_MULT_PORTA_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP1_MULT_PORTA_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP1_MULT_PORTA_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP1_MULT_PORTA_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP1_MULT_PORTA_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP1_MULT_PORTA_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP1_MULT_PORTA_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_MULT_PORTA_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_MULT_PORTA_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_MULT_PORTA_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_MULT_PORTA_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_MULT_PORTA_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_MULT_PORTA_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_GRPX3_CLUT | R/W | 0h | The interrupt for Graphics 2 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_GRPX2_CLUT | R/W | 0h | The interrupt for Graphics 1 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_GRPX1_CLUT | R/W | 0h | The interrupt for Graphics 0 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-244. VPDMA_int1_channel1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_GRPX3_STENCIL | R/W | 0h | The interrupt for Graphics 2 Stencil should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_GRPX2_STENCIL | R/W | 0h | The interrupt for Graphics 1 Stencil should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_GRPX1_STENCIL | R/W | 0h | The interrupt for Graphics 0 Stencil should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.37 VPDMA_int1_channel2_int_stat Register (offset = A0h) [reset = 0h]

VPDMA_int1_channel2_int_stat is shown in Figure 12-332 and described in Table 12-245.

**Figure 12-332. VPDMA_int1_channel2_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP1_MULT_ANCB_SRC9 | INT_STAT_VIP1_MULT_ANCB_SRC8 | INT_STAT_VIP1_MULT_ANCB_SRC7 | INT_STAT_VIP1_MULT_ANCB_SRC6 | INT_STAT_VIP1_MULT_ANCB_SRC5 | INT_STAT_VIP1_MULT_ANCB_SRC4 | INT_STAT_VIP1_MULT_ANCB_SRC3 | INT_STAT_VIP1_MULT_ANCB_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP1_MULT_ANCB_SRC1 | INT_STAT_VIP1_MULT_ANCB_SRC0 | INT_STAT_VIP1_MULT_ANCA_SRC15 | INT_STAT_VIP1_MULT_ANCA_SRC14 | INT_STAT_VIP1_MULT_ANCA_SRC13 | INT_STAT_VIP1_MULT_ANCA_SRC12 | INT_STAT_VIP1_MULT_ANCA_SRC11 | INT_STAT_VIP1_MULT_ANCA_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP1_MULT_ANCA_SRC9 | INT_STAT_VIP1_MULT_ANCA_SRC8 | INT_STAT_VIP1_MULT_ANCA_SRC7 | INT_STAT_VIP1_MULT_ANCA_SRC6 | INT_STAT_VIP1_MULT_ANCA_SRC5 | INT_STAT_VIP1_MULT_ANCA_SRC4 | INT_STAT_VIP1_MULT_ANCA_SRC3 | INT_STAT_VIP1_MULT_ANCA_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_MULT_ANCA_SRC1 | INT_STAT_VIP1_MULT_ANCA_SRC0 | INT_STAT_VIP1_MULT_PORTB_SRC15 | INT_STAT_VIP1_MULT_PORTB_SRC14 | INT_STAT_VIP1_MULT_PORTB_SRC13 | INT_STAT_VIP1_MULT_PORTB_SRC12 | INT_STAT_VIP1_MULT_PORTB_SRC11 | INT_STAT_VIP1_MULT_PORTB_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-245. VPDMA_int1_channel2_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP1_MULT_ANCB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP1_MULT_ANCB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP1_MULT_ANCB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP1_MULT_ANCB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_MULT_ANCB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_MULT_ANCB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-245. VPDMA_int1_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP1_MULT_ANCB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP1_MULT_ANCB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP1_MULT_ANCB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP1_MULT_ANCB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP1_MULT_ANCA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP1_MULT_ANCA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP1_MULT_ANCA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP1_MULT_ANCA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP1_MULT_ANCA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP1_MULT_ANCA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-245. VPDMA_int1_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP1_MULT_ANCA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP1_MULT_ANCA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP1_MULT_ANCA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP1_MULT_ANCA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_MULT_ANCA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_MULT_ANCA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_MULT_ANCA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_MULT_ANCA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_MULT_ANCA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_MULT_ANCA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-245. VPDMA_int1_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_MULT_PORTB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_MULT_PORTB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_MULT_PORTB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_MULT_PORTB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP1_MULT_PORTB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP1_MULT_PORTB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.38 VPDMA_int1_channel2_int_mask Register (offset = A4h) [reset = 0h]

VPDMA_int1_channel2_int_mask is shown in Figure 12-333 and described in Table 12-246.

**Figure 12-333. VPDMA_int1_channel2_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_ANCB_SRC9 | INT_MASK_VIP1_MULT_ANCB_SRC8 | INT_MASK_VIP1_MULT_ANCB_SRC7 | INT_MASK_VIP1_MULT_ANCB_SRC6 | INT_MASK_VIP1_MULT_ANCB_SRC5 | INT_MASK_VIP1_MULT_ANCB_SRC4 | INT_MASK_VIP1_MULT_ANCB_SRC3 | INT_MASK_VIP1_MULT_ANCB_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP1_MULT_ANCB_SRC1 | INT_MASK_VIP1_MULT_ANCB_SRC0 | INT_MASK_VIP1_MULT_ANCA_SRC15 | INT_MASK_VIP1_MULT_ANCA_SRC14 | INT_MASK_VIP1_MULT_ANCA_SRC13 | INT_MASK_VIP1_MULT_ANCA_SRC12 | INT_MASK_VIP1_MULT_ANCA_SRC11 | INT_MASK_VIP1_MULT_ANCA_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP1_MULT_ANCA_SRC9 | INT_MASK_VIP1_MULT_ANCA_SRC8 | INT_MASK_VIP1_MULT_ANCA_SRC7 | INT_MASK_VIP1_MULT_ANCA_SRC6 | INT_MASK_VIP1_MULT_ANCA_SRC5 | INT_MASK_VIP1_MULT_ANCA_SRC4 | INT_MASK_VIP1_MULT_ANCA_SRC3 | INT_MASK_VIP1_MULT_ANCA_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP1_MULT_ANCA_SRC1 | INT_MASK_VIP1_MULT_ANCA_SRC0 | INT_MASK_VIP1_MULT_PORTB_SRC15 | INT_MASK_VIP1_MULT_PORTB_SRC14 | INT_MASK_VIP1_MULT_PORTB_SRC13 | INT_MASK_VIP1_MULT_PORTB_SRC12 | INT_MASK_VIP1_MULT_PORTB_SRC11 | INT_MASK_VIP1_MULT_PORTB_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-246. VPDMA_int1_channel2_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP1_MULT_ANCB_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP1_MULT_ANCB_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP1_MULT_ANCB_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP1_MULT_ANCB_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_MULT_ANCB_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_MULT_ANCB_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP1_MULT_ANCB_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP1_MULT_ANCB_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP1_MULT_ANCB_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP1_MULT_ANCB_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP1_MULT_ANCA_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-246. VPDMA_int1_channel2_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP1_MULT_ANCA_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP1_MULT_ANCA_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP1_MULT_ANCA_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP1_MULT_ANCA_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP1_MULT_ANCA_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP1_MULT_ANCA_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP1_MULT_ANCA_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP1_MULT_ANCA_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP1_MULT_ANCA_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_MULT_ANCA_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_MULT_ANCA_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_MULT_ANCA_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_MULT_ANCA_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_MULT_ANCA_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_MULT_ANCA_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_MULT_PORTB_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_MULT_PORTB_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_MULT_PORTB_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-246. VPDMA_int1_channel2_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP1_MULT_PORTB_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP1_MULT_PORTB_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP1_MULT_PORTB_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.39 VPDMA_int1_channel3_int_stat Register (offset = A8h) [reset = 0h]

VPDMA_int1_channel3_int_stat is shown in Figure 12-334 and described in Table 12-247.

**Figure 12-334. VPDMA_int1_channel3_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP2_MULT_PORTB_SRC3 | INT_STAT_VIP2_MULT_PORTB_SRC2 | INT_STAT_VIP2_MULT_PORTB_SRC1 | INT_STAT_VIP2_MULT_PORTB_SRC0 | INT_STAT_VIP2_MULT_PORTA_SRC15 | INT_STAT_VIP2_MULT_PORTA_SRC14 | INT_STAT_VIP2_MULT_PORTA_SRC13 | INT_STAT_VIP2_MULT_PORTA_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP2_MULT_PORTA_SRC11 | INT_STAT_VIP2_MULT_PORTA_SRC10 | INT_STAT_VIP2_MULT_PORTA_SRC9 | INT_STAT_VIP2_MULT_PORTA_SRC8 | INT_STAT_VIP2_MULT_PORTA_SRC7 | INT_STAT_VIP2_MULT_PORTA_SRC6 | INT_STAT_VIP2_MULT_PORTA_SRC5 | INT_STAT_VIP2_MULT_PORTA_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_MULT_PORTA_SRC3 | INT_STAT_VIP2_MULT_PORTA_SRC2 | INT_STAT_VIP2_MULT_PORTA_SRC1 | INT_STAT_VIP2_MULT_PORTA_SRC0 | INT_STAT_VIP1_PORTB_RGB | INT_STAT_VIP1_PORTA_RGB | INT_STAT_VIP1_PORTB_CHROMA | INT_STAT_VIP1_PORTB_LUMA |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_PORTA_CHROMA | INT_STAT_VIP1_PORTA_LUMA | INT_STAT_VIP1_MULT_ANCB_SRC15 | INT_STAT_VIP1_MULT_ANCB_SRC14 | INT_STAT_VIP1_MULT_ANCB_SRC13 | INT_STAT_VIP1_MULT_ANCB_SRC12 | INT_STAT_VIP1_MULT_ANCB_SRC11 | INT_STAT_VIP1_MULT_ANCB_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-247. VPDMA_int1_channel3_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP2_MULT_PORTB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP2_MULT_PORTB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP2_MULT_PORTB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_MULT_PORTB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP2_MULT_PORTA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP2_MULT_PORTA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-247. VPDMA_int1_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP2_MULT_PORTA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP2_MULT_PORTA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP2_MULT_PORTA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP2_MULT_PORTA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP2_MULT_PORTA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP2_MULT_PORTA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP2_MULT_PORTA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP2_MULT_PORTA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_MULT_PORTA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP2_MULT_PORTA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-247. VPDMA_int1_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP2_MULT_PORTA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_MULT_PORTA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_MULT_PORTA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_MULT_PORTA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_PORTB_RGB | W | 0h | The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_PORTA_RGB | W | 0h | The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_PORTB_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_PORTB_LUMA | W | 0h | The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_PORTA_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_PORTA_LUMA | W | 0h | The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-247. VPDMA_int1_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_MULT_ANCB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_MULT_ANCB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_MULT_ANCB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_MULT_ANCB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP1_MULT_ANCB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP1_MULT_ANCB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.40 VPDMA_int1_channel3_int_mask Register (offset = ACh) [reset = 0h]

VPDMA_int1_channel3_int_mask is shown in Figure 12-335 and described in Table 12-248.

**Figure 12-335. VPDMA_int1_channel3_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP2_MULT_PORTB_SRC3 | INT_MASK_VIP2_MULT_PORTB_SRC2 | INT_MASK_VIP2_MULT_PORTB_SRC1 | INT_MASK_VIP2_MULT_PORTB_SRC0 | INT_MASK_VIP2_MULT_PORTA_SRC15 | INT_MASK_VIP2_MULT_PORTA_SRC14 | INT_MASK_VIP2_MULT_PORTA_SRC13 | INT_MASK_VIP2_MULT_PORTA_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP2_MULT_PORTA_SRC11 | INT_MASK_VIP2_MULT_PORTA_SRC10 | INT_MASK_VIP2_MULT_PORTA_SRC9 | INT_MASK_VIP2_MULT_PORTA_SRC8 | INT_MASK_VIP2_MULT_PORTA_SRC7 | INT_MASK_VIP2_MULT_PORTA_SRC6 | INT_MASK_VIP2_MULT_PORTA_SRC5 | INT_MASK_VIP2_MULT_PORTA_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_MULT_PORTA_SRC3 | INT_MASK_VIP2_MULT_PORTA_SRC2 | INT_MASK_VIP2_MULT_PORTA_SRC1 | INT_MASK_VIP2_MULT_PORTA_SRC0 | INT_MASK_VIP1_PORTB_RGB | INT_MASK_VIP1_PORTA_RGB | INT_MASK_VIP1_PORTB_CHROMA | INT_MASK_VIP1_PORTB_LUMA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP1_PORTA_CHROMA | INT_MASK_VIP1_PORTA_LUMA | INT_MASK_VIP1_MULT_ANCB_SRC15 | INT_MASK_VIP1_MULT_ANCB_SRC14 | INT_MASK_VIP1_MULT_ANCB_SRC13 | INT_MASK_VIP1_MULT_ANCB_SRC12 | INT_MASK_VIP1_MULT_ANCB_SRC11 | INT_MASK_VIP1_MULT_ANCB_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-248. VPDMA_int1_channel3_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP2_MULT_PORTB_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP2_MULT_PORTB_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP2_MULT_PORTB_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_MULT_PORTB_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP2_MULT_PORTA_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP2_MULT_PORTA_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP2_MULT_PORTA_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP2_MULT_PORTA_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP2_MULT_PORTA_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP2_MULT_PORTA_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP2_MULT_PORTA_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-248. VPDMA_int1_channel3_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP2_MULT_PORTA_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP2_MULT_PORTA_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP2_MULT_PORTA_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_MULT_PORTA_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_MULT_PORTA_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_MULT_PORTA_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_MULT_PORTA_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_MULT_PORTA_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_MULT_PORTA_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_PORTB_RGB | R/W | 0h | The interrupt for Video Input 1 Port B RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_PORTA_RGB | R/W | 0h | The interrupt for Video Input 1 Port A RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_PORTB_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_PORTB_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_PORTA_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_PORTA_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_MULT_ANCB_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_MULT_ANCB_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_MULT_ANCB_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-248. VPDMA_int1_channel3_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP1_MULT_ANCB_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP1_MULT_ANCB_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP1_MULT_ANCB_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.41 VPDMA_int1_channel4_int_stat Register (offset = B0h) [reset = 0h]

VPDMA_int1_channel4_int_stat is shown in Figure 12-336 and described in Table 12-249.

**Figure 12-336. VPDMA_int1_channel4_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP2_MULT_ANCB_SRC3 | INT_STAT_VIP2_MULT_ANCB_SRC2 | INT_STAT_VIP2_MULT_ANCB_SRC1 | INT_STAT_VIP2_MULT_ANCB_SRC0 | INT_STAT_VIP2_MULT_ANCA_SRC15 | INT_STAT_VIP2_MULT_ANCA_SRC14 | INT_STAT_VIP2_MULT_ANCA_SRC13 | INT_STAT_VIP2_MULT_ANCA_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP2_MULT_ANCA_SRC11 | INT_STAT_VIP2_MULT_ANCA_SRC10 | INT_STAT_VIP2_MULT_ANCA_SRC9 | INT_STAT_VIP2_MULT_ANCA_SRC8 | INT_STAT_VIP2_MULT_ANCA_SRC7 | INT_STAT_VIP2_MULT_ANCA_SRC6 | INT_STAT_VIP2_MULT_ANCA_SRC5 | INT_STAT_VIP2_MULT_ANCA_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_MULT_ANCA_SRC3 | INT_STAT_VIP2_MULT_ANCA_SRC2 | INT_STAT_VIP2_MULT_ANCA_SRC1 | INT_STAT_VIP2_MULT_ANCA_SRC0 | INT_STAT_VIP2_MULT_PORTB_SRC15 | INT_STAT_VIP2_MULT_PORTB_SRC14 | INT_STAT_VIP2_MULT_PORTB_SRC13 | INT_STAT_VIP2_MULT_PORTB_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_MULT_PORTB_SRC11 | INT_STAT_VIP2_MULT_PORTB_SRC10 | INT_STAT_VIP2_MULT_PORTB_SRC9 | INT_STAT_VIP2_MULT_PORTB_SRC8 | INT_STAT_VIP2_MULT_PORTB_SRC7 | INT_STAT_VIP2_MULT_PORTB_SRC6 | INT_STAT_VIP2_MULT_PORTB_SRC5 | INT_STAT_VIP2_MULT_PORTB_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-249. VPDMA_int1_channel4_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP2_MULT_ANCB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP2_MULT_ANCB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP2_MULT_ANCB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_MULT_ANCB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP2_MULT_ANCA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP2_MULT_ANCA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-249. VPDMA_int1_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP2_MULT_ANCA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP2_MULT_ANCA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP2_MULT_ANCA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP2_MULT_ANCA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP2_MULT_ANCA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP2_MULT_ANCA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP2_MULT_ANCA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP2_MULT_ANCA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_MULT_ANCA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP2_MULT_ANCA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-249. VPDMA_int1_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP2_MULT_ANCA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_MULT_ANCA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_MULT_ANCA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_MULT_ANCA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP2_MULT_PORTB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP2_MULT_PORTB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_MULT_PORTB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_MULT_PORTB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_MULT_PORTB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP2_MULT_PORTB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-249. VPDMA_int1_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP2_MULT_PORTB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP2_MULT_PORTB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP2_MULT_PORTB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP2_MULT_PORTB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP2_MULT_PORTB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP2_MULT_PORTB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.42 VPDMA_int1_channel4_int_mask Register (offset = B4h) [reset = 0h]

VPDMA_int1_channel4_int_mask is shown in Figure 12-337 and described in Table 12-250.

**Figure 12-337. VPDMA_int1_channel4_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP2_MULT_ANCB_SRC3 | INT_MASK_VIP2_MULT_ANCB_SRC2 | INT_MASK_VIP2_MULT_ANCB_SRC1 | INT_MASK_VIP2_MULT_ANCB_SRC0 | INT_MASK_VIP2_MULT_ANCA_SRC15 | INT_MASK_VIP2_MULT_ANCA_SRC14 | INT_MASK_VIP2_MULT_ANCA_SRC13 | INT_MASK_VIP2_MULT_ANCA_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP2_MULT_ANCA_SRC11 | INT_MASK_VIP2_MULT_ANCA_SRC10 | INT_MASK_VIP2_MULT_ANCA_SRC9 | INT_MASK_VIP2_MULT_ANCA_SRC8 | INT_MASK_VIP2_MULT_ANCA_SRC7 | INT_MASK_VIP2_MULT_ANCA_SRC6 | INT_MASK_VIP2_MULT_ANCA_SRC5 | INT_MASK_VIP2_MULT_ANCA_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_MULT_ANCA_SRC3 | INT_MASK_VIP2_MULT_ANCA_SRC2 | INT_MASK_VIP2_MULT_ANCA_SRC1 | INT_MASK_VIP2_MULT_ANCA_SRC0 | INT_MASK_VIP2_MULT_PORTB_SRC15 | INT_MASK_VIP2_MULT_PORTB_SRC14 | INT_MASK_VIP2_MULT_PORTB_SRC13 | INT_MASK_VIP2_MULT_PORTB_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_MULT_PORTB_SRC11 | INT_MASK_VIP2_MULT_PORTB_SRC10 | INT_MASK_VIP2_MULT_PORTB_SRC9 | INT_MASK_VIP2_MULT_PORTB_SRC8 | INT_MASK_VIP2_MULT_PORTB_SRC7 | INT_MASK_VIP2_MULT_PORTB_SRC6 | INT_MASK_VIP2_MULT_PORTB_SRC5 | INT_MASK_VIP2_MULT_PORTB_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-250. VPDMA_int1_channel4_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP2_MULT_ANCB_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP2_MULT_ANCB_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP2_MULT_ANCB_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_MULT_ANCB_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP2_MULT_ANCA_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP2_MULT_ANCA_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP2_MULT_ANCA_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP2_MULT_ANCA_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP2_MULT_ANCA_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP2_MULT_ANCA_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP2_MULT_ANCA_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-250. VPDMA_int1_channel4_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP2_MULT_ANCA_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP2_MULT_ANCA_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP2_MULT_ANCA_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_MULT_ANCA_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_MULT_ANCA_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_MULT_ANCA_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_MULT_ANCA_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_MULT_ANCA_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_MULT_ANCA_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP2_MULT_PORTB_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP2_MULT_PORTB_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_MULT_PORTB_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_MULT_PORTB_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_MULT_PORTB_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_MULT_PORTB_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP2_MULT_PORTB_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP2_MULT_PORTB_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP2_MULT_PORTB_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-250. VPDMA_int1_channel4_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP2_MULT_PORTB_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP2_MULT_PORTB_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP2_MULT_PORTB_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.43 VPDMA_int1_channel5_int_stat Register (offset = B8h) [reset = 0h]

VPDMA_int1_channel5_int_stat is shown in Figure 12-338 and described in Table 12-251.

**Figure 12-338. VPDMA_int1_channel5_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_TRANSCODE2_CHROMA | INT_STAT_TRANSCODE2_LUMA | INT_STAT_TRANSCODE1_CHROMA | INT_STAT_TRANSCODE1_LUMA | INT_STAT_AUX_IN | INT_STAT_PIP_FRAME | INT_STAT_POST_COMP_WR | INT_STAT_VBI_SD_VENC |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | INT_STAT_NF_LAST_CHROMA | INT_STAT_NF_LAST_LUMA | INT_STAT_NF_WRITE_CHROMA | INT_STAT_NF_WRITE_LUMA | INT_STAT_NF_READ | INT_STAT_VIP2_PORTB_RGB | INT_STAT_VIP2_PORTA_RGB |
| R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_PORTB_CHROMA | INT_STAT_VIP2_PORTB_LUMA | INT_STAT_VIP2_PORTA_CHROMA | INT_STAT_VIP2_PORTA_LUMA | INT_STAT_VIP2_MULT_ANCB_SRC15 | INT_STAT_VIP2_MULT_ANCB_SRC14 | INT_STAT_VIP2_MULT_ANCB_SRC13 | INT_STAT_VIP2_MULT_ANCB_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_MULT_ANCB_SRC11 | INT_STAT_VIP2_MULT_ANCB_SRC10 | INT_STAT_VIP2_MULT_ANCB_SRC9 | INT_STAT_VIP2_MULT_ANCB_SRC8 | INT_STAT_VIP2_MULT_ANCB_SRC7 | INT_STAT_VIP2_MULT_ANCB_SRC6 | INT_STAT_VIP2_MULT_ANCB_SRC5 | INT_STAT_VIP2_MULT_ANCB_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-251. VPDMA_int1_channel5_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_TRANSCODE2_CHROMA | W | 0h | The last read DMA transaction has occurred for channel transcode2_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans2_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_TRANSCODE2_LUMA | W | 0h | The last read DMA transaction has occurred for channel transcode2_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans2_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_TRANSCODE1_CHROMA | W | 0h | The last read DMA transaction has occurred for channel transcode1_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans1_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_TRANSCODE1_LUMA | W | 0h | The last read DMA transaction has occurred for channel transcode1_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans1_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_AUX_IN | W | 0h | The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-251. VPDMA_int1_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 26 | INT_STAT_PIP_FRAME | W | 0h | The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_POST_COMP_WR | W | 0h | The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VBI_SD_VENC | W | 0h | The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | Reserved | R | 0h | |
| 22 | INT_STAT_NF_LAST_CHROMA | W | 0h | The last read DMA transaction has occurred for channel nf_last_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_420_uv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_NF_LAST_LUMA | W | 0h | The last read DMA transaction has occurred for channel nf_last_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_420_y_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_NF_WRITE_CHROMA | W | 0h | The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_NF_WRITE_LUMA | W | 0h | The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_NF_READ | W | 0h | The last read DMA transaction has occurred for channel nf_read and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_422_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_PORTB_RGB | W | 0h | The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-251. VPDMA_int1_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | INT_STAT_VIP2_PORTA_RGB | W | 0h | The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_VIP2_PORTB_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_PORTB_LUMA | W | 0h | The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_PORTA_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_PORTA_LUMA | W | 0h | The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP2_MULT_ANCB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP2_MULT_ANCB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_MULT_ANCB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_MULT_ANCB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_MULT_ANCB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-251. VPDMA_int1_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | INT_STAT_VIP2_MULT_ANCB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 5 | INT_STAT_VIP2_MULT_ANCB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP2_MULT_ANCB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP2_MULT_ANCB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP2_MULT_ANCB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP2_MULT_ANCB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP2_MULT_ANCB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.44 VPDMA_int1_channel5_int_mask Register (offset = BCh) [reset = 0h]

VPDMA_int1_channel5_int_mask is shown in Figure 12-339 and described in Table 12-252.

**Figure 12-339. VPDMA_int1_channel5_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_TRANSCODE2_CHROMA | INT_MASK_TRANSCODE2_LUMA | INT_MASK_TRANSCODE1_CHROMA | INT_MASK_TRANSCODE1_LUMA | INT_MASK_AUX_IN | INT_MASK_PIP_FRAME | INT_MASK_POST_COMP_WR | INT_MASK_VBI_SD_VENC |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | INT_MASK_NF_LAST_CHROMA | INT_MASK_NF_LAST_LUMA | INT_MASK_NF_WRITE_CHROMA | INT_MASK_NF_WRITE_LUMA | INT_MASK_NF_READ | INT_MASK_VIP2_PORTB_RGB | INT_MASK_VIP2_PORTA_RGB |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_PORTB_CHROMA | INT_MASK_VIP2_PORTB_LUMA | INT_MASK_VIP2_PORTA_CHROMA | INT_MASK_VIP2_PORTA_LUMA | INT_MASK_VIP2_MULT_ANCB_SRC15 | INT_MASK_VIP2_MULT_ANCB_SRC14 | INT_MASK_VIP2_MULT_ANCB_SRC13 | INT_MASK_VIP2_MULT_ANCB_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_MULT_ANCB_SRC11 | INT_MASK_VIP2_MULT_ANCB_SRC10 | INT_MASK_VIP2_MULT_ANCB_SRC9 | INT_MASK_VIP2_MULT_ANCB_SRC8 | INT_MASK_VIP2_MULT_ANCB_SRC7 | INT_MASK_VIP2_MULT_ANCB_SRC6 | INT_MASK_VIP2_MULT_ANCB_SRC5 | INT_MASK_VIP2_MULT_ANCB_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-252. VPDMA_int1_channel5_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_TRANSCODE2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_TRANSCODE2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_TRANSCODE1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_TRANSCODE1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_AUX_IN | R/W | 0h | The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_PIP_FRAME | R/W | 0h | The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_POST_COMP_WR | R/W | 0h | The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VBI_SD_VENC | R/W | 0h | The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | Reserved | R | 0h | |
| 22 | INT_MASK_NF_LAST_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_NF_LAST_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-252. VPDMA_int1_channel5_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_NF_WRITE_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_NF_WRITE_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_NF_READ | R/W | 0h | The interrupt for Noise Filter Input Data 422 Interleaved should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_PORTB_RGB | R/W | 0h | The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_PORTA_RGB | R/W | 0h | The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_PORTB_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_PORTB_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_PORTA_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_PORTA_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP2_MULT_ANCB_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP2_MULT_ANCB_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_MULT_ANCB_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_MULT_ANCB_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_MULT_ANCB_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_MULT_ANCB_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP2_MULT_ANCB_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP2_MULT_ANCB_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP2_MULT_ANCB_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-252. VPDMA_int1_channel5_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP2_MULT_ANCB_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP2_MULT_ANCB_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP2_MULT_ANCB_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.45 VPDMA_int1_channel6_int_stat Register (offset = C0h) [reset = 0h]

VPDMA_int1_channel6_int_stat is shown in Figure 12-340 and described in Table 12-253.

**Figure 12-340. VPDMA_int1_channel6_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | INT_STAT_OTHER |
| R-0h | | | | | | | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-253. VPDMA_int1_channel6_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | INT_STAT_OTHER | W | 0h | This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.46 VPDMA_int1_channel6_int_mask Register (offset = C4h) [reset = 0h]

VPDMA_int1_channel6_int_mask is shown in Figure 12-341 and described in Table 12-254.

**Figure 12-341. VPDMA_int1_channel6_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | INT_MASK_OTHER |
| R-0h | | | | | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-254. VPDMA_int1_channel6_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | INT_MASK_OTHER | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.47 VPDMA_int1_client0_int_stat Register (offset = C8h) [reset = 0h]

VPDMA_int1_client0_int_stat is shown in Figure 12-342 and described in Table 12-255.

**Figure 12-342. VPDMA_int1_client0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_GRPX1_DATA | INT_STAT_COMP_WRBK | INT_STAT_SC_OUT | Reserved | | | | |
| W-0h | W-0h | W-0h | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | INT_STAT_SC_IN_LUMA | INT_STAT_SC_IN_CHROMA | INT_STAT_PIP_WRBK | INT_STAT_DEI_SC_OUT | Reserved |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_DEI_HQ_MV_OUT | Reserved | | INT_STAT_DEI_HQ_MV_IN | Reserved | | | |
| W-0h | R-0h | | W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_STAT_DEI_HQ_3_CHROMA | INT_STAT_DEI_HQ_3_LUMA | INT_STAT_DEI_HQ_2_CHROMA | INT_STAT_DEI_HQ_2_LUMA | INT_STAT_DEI_HQ_1_LUMA | INT_STAT_DEI_HQ_1_CHROMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-255. VPDMA_int1_client0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_GRPX1_DATA | W | 0h | The client interface grpx1_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_COMP_WRBK | W | 0h | The client interface comp_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_SC_OUT | W | 0h | The client interface sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28-21 | Reserved | R | 0h | |
| 20 | INT_STAT_SC_IN_LUMA | W | 0h | The client interface sc_in_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_SC_IN_CHROMA | W | 0h | The client interface sc_in_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-255. VPDMA_int1_client0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | INT_STAT_PIP_WRBK | W | 0h | The client interface pip_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_DEI_SC_OUT | W | 0h | The client interface dei_sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | Reserved | R | 0h | |
| 15 | INT_STAT_DEI_HQ_MV_OUT | W | 0h | The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_STAT_DEI_HQ_MV_IN | W | 0h | The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11-6 | Reserved | R | 0h | |
| 5 | INT_STAT_DEI_HQ_3_CHROMA | W | 0h | The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_DEI_HQ_3_LUMA | W | 0h | The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_DEI_HQ_2_CHROMA | W | 0h | The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_DEI_HQ_2_LUMA | W | 0h | The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-255. VPDMA_int1_client0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | INT_STAT_DEI_HQ_1_LUMA | W | 0h | The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_DEI_HQ_1_CHROMA | W | 0h | The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.48 VPDMA_int1_client0_int_mask Register (offset = CCh) [reset = 0h]

VPDMA_int1_client0_int_mask is shown in Figure 12-343 and described in Table 12-256.

**Figure 12-343. VPDMA_int1_client0_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_GRPX1_DATA | INT_MASK_COMP_WRBK | INT_MASK_SC_OUT | Reserved | | | | |
| R/W-0h | R/W-0h | R/W-0h | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | INT_MASK_SC_IN_LUMA | INT_MASK_SC_IN_CHROMA | INT_MASK_PIP_WRBK | INT_MASK_DEI_SC_OUT | Reserved |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_DEI_HQ_MV_OUT | Reserved | | INT_MASK_DEI_HQ_MV_IN | Reserved | | | |
| R/W-0h | R-0h | | R/W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_MASK_DEI_HQ_3_CHROMA | INT_MASK_DEI_HQ_3_LUMA | INT_MASK_DEI_HQ_2_CHROMA | INT_MASK_DEI_HQ_2_LUMA | INT_MASK_DEI_HQ_1_LUMA | INT_MASK_DEI_HQ_1_CHROMA |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-256. VPDMA_int1_client0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_GRPX1_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_COMP_WRBK | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_SC_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28-21 | Reserved | R | 0h | |
| 20 | INT_MASK_SC_IN_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_SC_IN_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_PIP_WRBK | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_DEI_SC_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | Reserved | R | 0h | |
| 15 | INT_MASK_DEI_HQ_MV_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_MASK_DEI_HQ_MV_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11-6 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-256. VPDMA_int1_client0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_MASK_DEI_HQ_3_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_DEI_HQ_3_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_DEI_HQ_2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_DEI_HQ_2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_DEI_HQ_1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_DEI_HQ_1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.49 VPDMA_int1_client1_int_stat Register (offset = D0h) [reset = 0h]

VPDMA_int1_client1_int_stat is shown in Figure 12-344 and described in Table 12-257.

**Figure 12-344. VPDMA_int1_client1_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | INT_STAT_VIP2_ANC_B | INT_STAT_VIP2_ANC_A | INT_STAT_VIP1_ANC_B | INT_STAT_VIP1_ANC_A | INT_STAT_TRANS2_LUMA | INT_STAT_TRANS2_CHROMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_TRANS1_LUMA | INT_STAT_TRANS1_CHROMA | INT_STAT_HDMI_WRBK_OUT | INT_STAT_VPI_CTL | INT_STAT_VBI_SDVENC | Reserved | INT_STAT_NF_420_UV_OUT | INT_STAT_NF_420_Y_OUT |
| W-0h | W-0h | W-0h | W-0h | W-0h | R-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_NF_420_UV_IN | INT_STAT_NF_420_Y_IN | INT_STAT_NF_422_IN | INT_STAT_GRPX3_ST | INT_STAT_GRPX2_ST | INT_STAT_GRPX1_ST | INT_STAT_VIP2_UP_UV | INT_STAT_VIP2_UP_Y |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_LO_UV | INT_STAT_VIP2_LO_Y | INT_STAT_VIP1_UP_UV | INT_STAT_VIP1_UP_Y | INT_STAT_VIP1_LO_UV | INT_STAT_VIP1_LO_Y | INT_STAT_GRPX3_DATA | INT_STAT_GRPX2_DATA |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-257. VPDMA_int1_client1_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29 | INT_STAT_VIP2_ANC_B | W | 0h | The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_ANC_A | W | 0h | The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_ANC_B | W | 0h | The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_ANC_A | W | 0h | The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_TRANS2_LUMA | W | 0h | The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-257. VPDMA_int1_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 24 | INT_STAT_TRANS2_CHROMA | W | 0h | The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_TRANS1_LUMA | W | 0h | The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_TRANS1_CHROMA | W | 0h | The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_HDMI_WRBK_OUT | W | 0h | The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VPI_CTL | W | 0h | The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VBI_SDVENC | W | 0h | The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | Reserved | R | 0h | |
| 17 | INT_STAT_NF_420_UV_OUT | W | 0h | The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_NF_420_Y_OUT | W | 0h | The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_NF_420_UV_IN | W | 0h | The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-257. VPDMA_int1_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | INT_STAT_NF_420_Y_IN | W | 0h | The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_NF_422_IN | W | 0h | The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_GRPX3_ST | W | 0h | The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_GRPX2_ST | W | 0h | The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_GRPX1_ST | W | 0h | The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_UP_UV | W | 0h | The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_UP_Y | W | 0h | The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_LO_UV | W | 0h | The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP2_LO_Y | W | 0h | The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-257. VPDMA_int1_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_UP_UV | W | 0h | The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_UP_Y | W | 0h | The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_LO_UV | W | 0h | The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_LO_Y | W | 0h | The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_GRPX3_DATA | W | 0h | The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_GRPX2_DATA | W | 0h | The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.50 VPDMA_int1_client1_int_mask Register (offset = D4h) [reset = 0h]

VPDMA_int1_client1_int_mask is shown in Figure 12-345 and described in Table 12-258.

**Figure 12-345. VPDMA_int1_client1_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | INT_MASK_VIP2_ANC_B | INT_MASK_VIP2_ANC_A | INT_MASK_VIP1_ANC_B | INT_MASK_VIP1_ANC_A | INT_MASK_TRANS2_LUMA | INT_MASK_TRANS2_CHROMA |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_TRANS1_LUMA | INT_MASK_TRANS1_CHROMA | INT_MASK_HDMI_WRBK_OUT | INT_MASK_VPI_CTL | INT_MASK_VBI_SDVENC | Reserved | INT_MASK_NF_420_UV_OUT | INT_MASK_NF_420_Y_OUT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_NF_420_UV_IN | INT_MASK_NF_420_Y_IN | INT_MASK_NF_422_IN | INT_MASK_GRPX3_ST | INT_MASK_GRPX2_ST | INT_MASK_GRPX1_ST | INT_MASK_VIP2_UP_UV | INT_MASK_VIP2_UP_Y |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_LO_UV | INT_MASK_VIP2_LO_Y | INT_MASK_VIP1_UP_UV | INT_MASK_VIP1_UP_Y | INT_MASK_VIP1_LO_UV | INT_MASK_VIP1_LO_Y | INT_MASK_GRPX3_DATA | INT_MASK_GRPX2_DATA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-258. VPDMA_int1_client1_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29 | INT_MASK_VIP2_ANC_B | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_ANC_A | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_ANC_B | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_ANC_A | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_TRANS2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_TRANS2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_TRANS1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_TRANS1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_HDMI_WRBK_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 20 | INT_MASK_VPI_CTL | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-258. VPDMA_int1_client1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | INT_MASK_VBI_SDVENC | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | Reserved | R | 0h | |
| 17 | INT_MASK_NF_420_UV_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_NF_420_Y_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_NF_420_UV_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_NF_420_Y_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_NF_422_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_GRPX3_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_GRPX2_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_GRPX1_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_UP_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_UP_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_LO_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_LO_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_UP_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_UP_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_LO_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_VIP1_LO_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_GRPX3_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-258. VPDMA_int1_client1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | INT_MASK_GRPX2_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.51 VPDMA_int1_list0_int_stat Register (offset = D8h) [reset = 0h]

VPDMA_int1_list0_int_stat is shown in Figure 12-346 and described in Table 12-259.

**Figure 12-346. VPDMA_int1_list0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_CONTROL_DESCRIPTOR_INT15 | INT_STAT_CONTROL_DESCRIPTOR_INT14 | INT_STAT_CONTROL_DESCRIPTOR_INT13 | INT_STAT_CONTROL_DESCRIPTOR_INT12 | INT_STAT_CONTROL_DESCRIPTOR_INT11 | INT_STAT_CONTROL_DESCRIPTOR_INT10 | INT_STAT_CONTROL_DESCRIPTOR_INT9 | INT_STAT_CONTROL_DESCRIPTOR_INT8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_CONTROL_DESCRIPTOR_INT7 | INT_STAT_CONTROL_DESCRIPTOR_INT6 | INT_STAT_CONTROL_DESCRIPTOR_INT5 | INT_STAT_CONTROL_DESCRIPTOR_INT4 | INT_STAT_CONTROL_DESCRIPTOR_INT3 | INT_STAT_CONTROL_DESCRIPTOR_INT2 | INT_STAT_CONTROL_DESCRIPTOR_INT1 | INT_STAT_CONTROL_DESCRIPTOR_INT0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_LIST7_NOTIFY | INT_STAT_LIST7_COMPLETE | INT_STAT_LIST6_NOTIFY | INT_STAT_LIST6_COMPLETE | INT_STAT_LIST5_NOTIFY | INT_STAT_LIST5_COMPLETE | INT_STAT_LIST4_NOTIFY | INT_STAT_LIST4_COMPLETE |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_LIST3_NOTIFY | INT_STAT_LIST3_COMPLETE | INT_STAT_LIST2_NOTIFY | INT_STAT_LIST2_COMPLETE | INT_STAT_LIST1_NOTIFY | INT_STAT_LIST1_COMPLETE | INT_STAT_LIST0_NOTIFY | INT_STAT_LIST0_COMPLETE |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-259. VPDMA_int1_list0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_CONTROL_DESCRIPTOR_INT15 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_CONTROL_DESCRIPTOR_INT14 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_CONTROL_DESCRIPTOR_INT13 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_CONTROL_DESCRIPTOR_INT12 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_CONTROL_DESCRIPTOR_INT11 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_CONTROL_DESCRIPTOR_INT10 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_CONTROL_DESCRIPTOR_INT9 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_CONTROL_DESCRIPTOR_INT8 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-259. VPDMA_int1_list0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | INT_STAT_CONTROL_DESCRIPTOR_INT7 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_CONTROL_DESCRIPTOR_INT6 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_CONTROL_DESCRIPTOR_INT5 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_CONTROL_DESCRIPTOR_INT4 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_CONTROL_DESCRIPTOR_INT3 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_CONTROL_DESCRIPTOR_INT2 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_CONTROL_DESCRIPTOR_INT1 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_CONTROL_DESCRIPTOR_INT0 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_LIST7_NOTIFY | W | 0h | A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_LIST7_COMPLETE | W | 0h | List 7 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_LIST6_NOTIFY | W | 0h | A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_LIST6_COMPLETE | W | 0h | List 6 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_LIST5_NOTIFY | W | 0h | A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_LIST5_COMPLETE | W | 0h | List 5 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_LIST4_NOTIFY | W | 0h | A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-259. VPDMA_int1_list0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | INT_STAT_LIST4_COMPLETE | W | 0h | List 4 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_LIST3_NOTIFY | W | 0h | A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_LIST3_COMPLETE | W | 0h | List 3 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 5 | INT_STAT_LIST2_NOTIFY | W | 0h | A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_LIST2_COMPLETE | W | 0h | List 2 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_LIST1_NOTIFY | W | 0h | A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_LIST1_COMPLETE | W | 0h | List 1 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_LIST0_NOTIFY | W | 0h | A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_LIST0_COMPLETE | W | 0h | List 0 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.52 VPDMA_int1_list0_int_mask Register (offset = DCh) [reset = 0h]

VPDMA_int1_list0_int_mask is shown in Figure 12-347 and described in Table 12-260.

**Figure 12-347. VPDMA_int1_list0_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_CONTROL_DESCRIPTOR_INT15 | INT_MASK_CONTROL_DESCRIPTOR_INT14 | INT_MASK_CONTROL_DESCRIPTOR_INT13 | INT_MASK_CONTROL_DESCRIPTOR_INT12 | INT_MASK_CONTROL_DESCRIPTOR_INT11 | INT_MASK_CONTROL_DESCRIPTOR_INT10 | INT_MASK_CONTROL_DESCRIPTOR_INT9 | INT_MASK_CONTROL_DESCRIPTOR_INT8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_CONTROL_DESCRIPTOR_INT7 | INT_MASK_CONTROL_DESCRIPTOR_INT6 | INT_MASK_CONTROL_DESCRIPTOR_INT5 | INT_MASK_CONTROL_DESCRIPTOR_INT4 | INT_MASK_CONTROL_DESCRIPTOR_INT3 | INT_MASK_CONTROL_DESCRIPTOR_INT2 | INT_MASK_CONTROL_DESCRIPTOR_INT1 | INT_MASK_CONTROL_DESCRIPTOR_INT0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_LIST7_NOTIFY | INT_MASK_LIST7_COMPLETE | INT_MASK_LIST6_NOTIFY | INT_MASK_LIST6_COMPLETE | INT_MASK_LIST5_NOTIFY | INT_MASK_LIST5_COMPLETE | INT_MASK_LIST4_NOTIFY | INT_MASK_LIST4_COMPLETE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_LIST3_NOTIFY | INT_MASK_LIST3_COMPLETE | INT_MASK_LIST2_NOTIFY | INT_MASK_LIST2_COMPLETE | INT_MASK_LIST1_NOTIFY | INT_MASK_LIST1_COMPLETE | INT_MASK_LIST0_NOTIFY | INT_MASK_LIST0_COMPLETE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-260. VPDMA_int1_list0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_CONTROL_DESCRIPTOR_INT15 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_CONTROL_DESCRIPTOR_INT14 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_CONTROL_DESCRIPTOR_INT13 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_CONTROL_DESCRIPTOR_INT12 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_CONTROL_DESCRIPTOR_INT11 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_CONTROL_DESCRIPTOR_INT10 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_CONTROL_DESCRIPTOR_INT9 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_CONTROL_DESCRIPTOR_INT8 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_CONTROL_DESCRIPTOR_INT7 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_CONTROL_DESCRIPTOR_INT6 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-260. VPDMA_int1_list0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | INT_MASK_CONTROL_DESCRIPTOR_INT5 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 20 | INT_MASK_CONTROL_DESCRIPTOR_INT4 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_CONTROL_DESCRIPTOR_INT3 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_CONTROL_DESCRIPTOR_INT2 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_CONTROL_DESCRIPTOR_INT1 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_CONTROL_DESCRIPTOR_INT0 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_LIST7_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_LIST7_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_LIST6_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_LIST6_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_LIST5_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_LIST5_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_LIST4_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_LIST4_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_LIST3_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_LIST3_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_LIST2_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_LIST2_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-260. VPDMA_int1_list0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | INT_MASK_LIST1_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_LIST1_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_LIST0_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_LIST0_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.53 VPDMA_int2_channel0_int_stat Register (offset = E0h) [reset = 0h]

VPDMA_int2_channel0_int_stat is shown in Figure 12-348 and described in Table 12-261.

**Figure 12-348. VPDMA_int2_channel0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_GRPX3 | INT_STAT_GRPX2 | INT_STAT_GRPX1 | INT_STAT_SCALER_OUT | Reserved | | | |
| W-0h | W-0h | W-0h | W-0h | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | INT_STAT_SCALER_CHROMA | INT_STAT_SCALER_LUMA | INT_STAT_HQ_SCALER | Reserved |
| R-0h | | | | W-0h | W-0h | W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_HQ_MV_OUT | Reserved | | INT_STAT_HQ_MV | Reserved | | | |
| W-0h | R-0h | | W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_STAT_HQ_VID3_CHROMA | INT_STAT_HQ_VID3_LUMA | INT_STAT_HQ_VID2_CHROMA | INT_STAT_HQ_VID2_LUMA | INT_STAT_HQ_VID1_CHROMA | INT_STAT_HQ_VID1_LUMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-261. VPDMA_int2_channel0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_GRPX3 | W | 0h | The last read DMA transaction has occurred for channel grpx3 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_GRPX2 | W | 0h | The last read DMA transaction has occurred for channel grpx2 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_GRPX1 | W | 0h | The last read DMA transaction has occurred for channel grpx1 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_SCALER_OUT | W | 0h | The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27-20 | Reserved | R | 0h | |
| 19 | INT_STAT_SCALER_CHROMA | W | 0h | The last write DMA transaction has completed for channel scaler_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-261. VPDMA_int2_channel0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | INT_STAT_SCALER_LUMA | W | 0h | The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_HQ_SCALER | W | 0h | The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | Reserved | R | 0h | |
| 15 | INT_STAT_HQ_MV_OUT | W | 0h | The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_STAT_HQ_MV | W | 0h | The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11-6 | Reserved | R | 0h | |
| 5 | INT_STAT_HQ_VID3_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid3_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_3_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_HQ_VID3_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid3_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_3_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_HQ_VID2_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid2_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_2_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_HQ_VID2_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid2_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_2_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-261. VPDMA_int2_channel0_int_stat Register Field Descriptions (continued)**

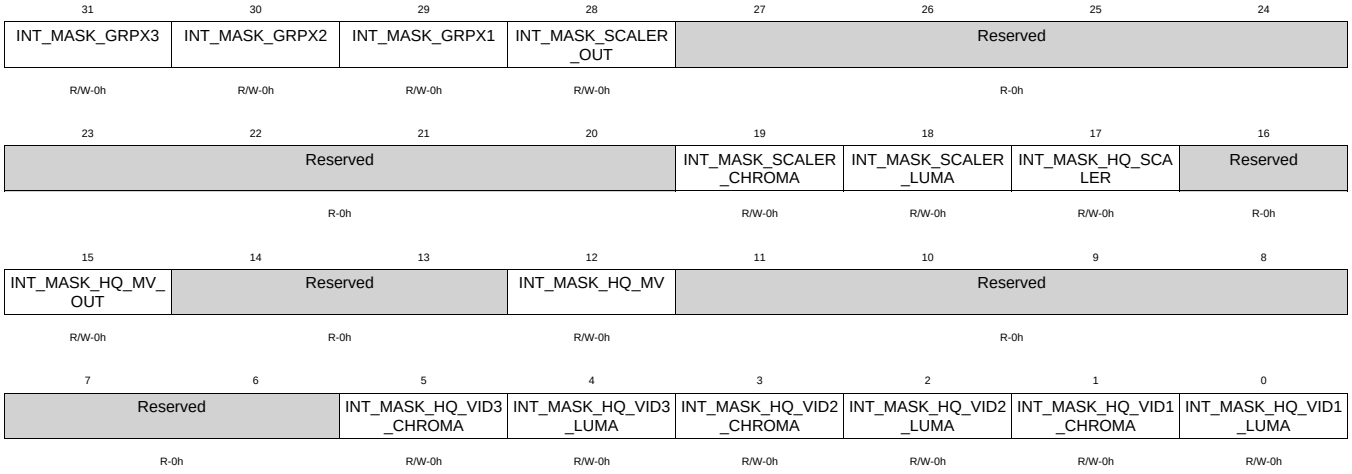
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | INT_STAT_HQ_VID1_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid1_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_1_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_HQ_VID1_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid1_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_1_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.54 VPDMA_int2_channel0_int_mask Register (offset = E4h) [reset = 0h]

VPDMA_int2_channel0_int_mask is shown in Figure 12-349 and described in Table 12-262.

**Figure 12-349. VPDMA_int2_channel0_int_mask Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-262. VPDMA_int2_channel0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_GRPX3 | R/W | 0h | The interrupt for Graphcis 2 Data should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_GRPX2 | R/W | 0h | The interrupt for Graphics 1 Data should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_GRPX1 | R/W | 0h | The interrupt for Graphics 0 Data should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_SCALER_OUT | R/W | 0h | The interrupt for Low Cost DEI Scalar Write to Memory should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27-20 | Reserved | R | 0h | |
| 19 | INT_MASK_SCALER_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_SCALER_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_HQ_SCALER | R/W | 0h | The interrupt for High Quality DEI Scaler Write to Memory should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | Reserved | R | 0h | |
| 15 | INT_MASK_HQ_MV_OUT | R/W | 0h | The interrupt for Low Cost DEI Motion Vector Write should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_MASK_HQ_MV | R/W | 0h | The interrupt for Low Cost DEI Motion Vector should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11-6 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-262. VPDMA_int2_channel0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_MASK_HQ_VID3_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_HQ_VID3_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_HQ_VID2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_HQ_VID2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_HQ_VID1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_HQ_VID1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.55 VPDMA_int2_channel1_int_stat Register (offset = E8h) [reset = 0h]

VPDMA_int2_channel1_int_stat is shown in Figure 12-350 and described in Table 12-263.

**Figure 12-350. VPDMA_int2_channel1_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP1_MULT_PORTB_SRC9 | INT_STAT_VIP1_MULT_PORTB_SRC8 | INT_STAT_VIP1_MULT_PORTB_SRC7 | INT_STAT_VIP1_MULT_PORTB_SRC6 | INT_STAT_VIP1_MULT_PORTB_SRC5 | INT_STAT_VIP1_MULT_PORTB_SRC4 | INT_STAT_VIP1_MULT_PORTB_SRC3 | INT_STAT_VIP1_MULT_PORTB_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP1_MULT_PORTB_SRC1 | INT_STAT_VIP1_MULT_PORTB_SRC0 | INT_STAT_VIP1_MULT_PORTA_SRC15 | INT_STAT_VIP1_MULT_PORTA_SRC14 | INT_STAT_VIP1_MULT_PORTA_SRC13 | INT_STAT_VIP1_MULT_PORTA_SRC12 | INT_STAT_VIP1_MULT_PORTA_SRC11 | INT_STAT_VIP1_MULT_PORTA_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP1_MULT_PORTA_SRC9 | INT_STAT_VIP1_MULT_PORTA_SRC8 | INT_STAT_VIP1_MULT_PORTA_SRC7 | INT_STAT_VIP1_MULT_PORTA_SRC6 | INT_STAT_VIP1_MULT_PORTA_SRC5 | INT_STAT_VIP1_MULT_PORTA_SRC4 | INT_STAT_VIP1_MULT_PORTA_SRC3 | INT_STAT_VIP1_MULT_PORTA_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_MULT_PORTA_SRC1 | INT_STAT_VIP1_MULT_PORTA_SRC0 | INT_STAT_GRPX3_CLUT | INT_STAT_GRPX2_CLUT | INT_STAT_GRPX1_CLUT | INT_STAT_GRPX3_STENCIL | INT_STAT_GRPX2_STENCIL | INT_STAT_GRPX1_STENCIL |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-263. VPDMA_int2_channel1_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP1_MULT_PORTB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP1_MULT_PORTB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP1_MULT_PORTB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP1_MULT_PORTB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_MULT_PORTB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_MULT_PORTB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-263. VPDMA_int2_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP1_MULT_PORTB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP1_MULT_PORTB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP1_MULT_PORTB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP1_MULT_PORTB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP1_MULT_PORTA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP1_MULT_PORTA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP1_MULT_PORTA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP1_MULT_PORTA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP1_MULT_PORTA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP1_MULT_PORTA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-263. VPDMA_int2_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP1_MULT_PORTA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP1_MULT_PORTA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP1_MULT_PORTA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP1_MULT_PORTA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_MULT_PORTA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_MULT_PORTA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_MULT_PORTA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_MULT_PORTA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_MULT_PORTA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_MULT_PORTA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-263. VPDMA_int2_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_GRPX3_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx3_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_GRPX2_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx2_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_GRPX1_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx1_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_GRPX3_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx3_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_GRPX2_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx2_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_GRPX1_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx1_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.56 VPDMA_int2_channel1_int_mask Register (offset = ECh) [reset = 0h]

VPDMA_int2_channel1_int_mask is shown in Figure 12-351 and described in Table 12-264.

**Figure 12-351. VPDMA_int2_channel1_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_PORTB_SRC9 | INT_MASK_VIP1_MULT_PORTB_SRC8 | INT_MASK_VIP1_MULT_PORTB_SRC7 | INT_MASK_VIP1_MULT_PORTB_SRC6 | INT_MASK_VIP1_MULT_PORTB_SRC5 | INT_MASK_VIP1_MULT_PORTB_SRC4 | INT_MASK_VIP1_MULT_PORTB_SRC3 | INT_MASK_VIP1_MULT_PORTB_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_PORTB_SRC1 | INT_MASK_VIP1_MULT_PORTB_SRC0 | INT_MASK_VIP1_MULT_PORTA_SRC15 | INT_MASK_VIP1_MULT_PORTA_SRC14 | INT_MASK_VIP1_MULT_PORTA_SRC13 | INT_MASK_VIP1_MULT_PORTA_SRC12 | INT_MASK_VIP1_MULT_PORTA_SRC11 | INT_MASK_VIP1_MULT_PORTA_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_PORTA_SRC9 | INT_MASK_VIP1_MULT_PORTA_SRC8 | INT_MASK_VIP1_MULT_PORTA_SRC7 | INT_MASK_VIP1_MULT_PORTA_SRC6 | INT_MASK_VIP1_MULT_PORTA_SRC5 | INT_MASK_VIP1_MULT_PORTA_SRC4 | INT_MASK_VIP1_MULT_PORTA_SRC3 | INT_MASK_VIP1_MULT_PORTA_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_PORTA_SRC1 | INT_MASK_VIP1_MULT_PORTA_SRC0 | INT_MASK_GRPX3_CLUT | INT_MASK_GRPX2_CLUT | INT_MASK_GRPX1_CLUT | INT_MASK_GRPX3_STENCIL | INT_MASK_GRPX2_STENCIL | INT_MASK_GRPX1_STENCIL |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-264. VPDMA_int2_channel1_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP1_MULT_PORTB_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 9 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP1_MULT_PORTB_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 8 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP1_MULT_PORTB_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 7 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP1_MULT_PORTB_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 6 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_MULT_PORTB_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 5 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_MULT_PORTB_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 4 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP1_MULT_PORTB_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 3 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP1_MULT_PORTB_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 2 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP1_MULT_PORTB_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 1 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP1_MULT_PORTB_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 0 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP1_MULT_PORTA_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 15 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-264. VPDMA_int2_channel1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP1_MULT_PORTA_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 14 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP1_MULT_PORTA_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 13 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP1_MULT_PORTA_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 12 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP1_MULT_PORTA_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 11 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP1_MULT_PORTA_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 10 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP1_MULT_PORTA_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 9 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP1_MULT_PORTA_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 8 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP1_MULT_PORTA_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 7 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP1_MULT_PORTA_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 6 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_MULT_PORTA_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 5 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_MULT_PORTA_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 4 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_MULT_PORTA_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 3 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_MULT_PORTA_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 2 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_MULT_PORTA_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 1 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_MULT_PORTA_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 0 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_GRPX3_CLUT | R/W | 0h | The interrupt for Graphics 2 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_GRPX2_CLUT | R/W | 0h | The interrupt for Graphics 1 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_GRPX1_CLUT | R/W | 0h | The interrupt for Graphics 0 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-264. VPDMA_int2_channel1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_GRPX3_STENCIL | R/W | 0h | The interrupt for Graphics 2 Stencil should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_GRPX2_STENCIL | R/W | 0h | The interrupt for Graphics 1 Stencil should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_GRPX1_STENCIL | R/W | 0h | The interrupt for Graphics 0 Stencil should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.57 VPDMA_int2_channel2_int_stat Register (offset = F0h) [reset = 0h]

VPDMA_int2_channel2_int_stat is shown in Figure 12-352 and described in Table 12-265.

**Figure 12-352. VPDMA_int2_channel2_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP1_MULT_ANCB_SRC9 | INT_STAT_VIP1_MULT_ANCB_SRC8 | INT_STAT_VIP1_MULT_ANCB_SRC7 | INT_STAT_VIP1_MULT_ANCB_SRC6 | INT_STAT_VIP1_MULT_ANCB_SRC5 | INT_STAT_VIP1_MULT_ANCB_SRC4 | INT_STAT_VIP1_MULT_ANCB_SRC3 | INT_STAT_VIP1_MULT_ANCB_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP1_MULT_ANCB_SRC1 | INT_STAT_VIP1_MULT_ANCB_SRC0 | INT_STAT_VIP1_MULT_ANCA_SRC15 | INT_STAT_VIP1_MULT_ANCA_SRC14 | INT_STAT_VIP1_MULT_ANCA_SRC13 | INT_STAT_VIP1_MULT_ANCA_SRC12 | INT_STAT_VIP1_MULT_ANCA_SRC11 | INT_STAT_VIP1_MULT_ANCA_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP1_MULT_ANCA_SRC9 | INT_STAT_VIP1_MULT_ANCA_SRC8 | INT_STAT_VIP1_MULT_ANCA_SRC7 | INT_STAT_VIP1_MULT_ANCA_SRC6 | INT_STAT_VIP1_MULT_ANCA_SRC5 | INT_STAT_VIP1_MULT_ANCA_SRC4 | INT_STAT_VIP1_MULT_ANCA_SRC3 | INT_STAT_VIP1_MULT_ANCA_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_MULT_ANCA_SRC1 | INT_STAT_VIP1_MULT_ANCA_SRC0 | INT_STAT_VIP1_MULT_PORTB_SRC15 | INT_STAT_VIP1_MULT_PORTB_SRC14 | INT_STAT_VIP1_MULT_PORTB_SRC13 | INT_STAT_VIP1_MULT_PORTB_SRC12 | INT_STAT_VIP1_MULT_PORTB_SRC11 | INT_STAT_VIP1_MULT_PORTB_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-265. VPDMA_int2_channel2_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP1_MULT_ANCB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP1_MULT_ANCB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP1_MULT_ANCB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP1_MULT_ANCB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_MULT_ANCB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_MULT_ANCB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-265. VPDMA_int2_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP1_MULT_ANCB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP1_MULT_ANCB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP1_MULT_ANCB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP1_MULT_ANCB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP1_MULT_ANCA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP1_MULT_ANCA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP1_MULT_ANCA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP1_MULT_ANCA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP1_MULT_ANCA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP1_MULT_ANCA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-265. VPDMA_int2_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP1_MULT_ANCA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP1_MULT_ANCA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP1_MULT_ANCA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP1_MULT_ANCA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_MULT_ANCA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_MULT_ANCA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_MULT_ANCA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_MULT_ANCA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_MULT_ANCA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_MULT_ANCA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-265. VPDMA_int2_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_MULT_PORTB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_MULT_PORTB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_MULT_PORTB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_MULT_PORTB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP1_MULT_PORTB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP1_MULT_PORTB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.58 VPDMA_int2_channel2_int_mask Register (offset = F4h) [reset = 0h]

VPDMA_int2_channel2_int_mask is shown in Figure 12-353 and described in Table 12-266.

**Figure 12-353. VPDMA_int2_channel2_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_ANCB_SRC9 | INT_MASK_VIP1_MULT_ANCB_SRC8 | INT_MASK_VIP1_MULT_ANCB_SRC7 | INT_MASK_VIP1_MULT_ANCB_SRC6 | INT_MASK_VIP1_MULT_ANCB_SRC5 | INT_MASK_VIP1_MULT_ANCB_SRC4 | INT_MASK_VIP1_MULT_ANCB_SRC3 | INT_MASK_VIP1_MULT_ANCB_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP1_MULT_ANCB_SRC1 | INT_MASK_VIP1_MULT_ANCB_SRC0 | INT_MASK_VIP1_MULT_ANCA_SRC15 | INT_MASK_VIP1_MULT_ANCA_SRC14 | INT_MASK_VIP1_MULT_ANCA_SRC13 | INT_MASK_VIP1_MULT_ANCA_SRC12 | INT_MASK_VIP1_MULT_ANCA_SRC11 | INT_MASK_VIP1_MULT_ANCA_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP1_MULT_ANCA_SRC9 | INT_MASK_VIP1_MULT_ANCA_SRC8 | INT_MASK_VIP1_MULT_ANCA_SRC7 | INT_MASK_VIP1_MULT_ANCA_SRC6 | INT_MASK_VIP1_MULT_ANCA_SRC5 | INT_MASK_VIP1_MULT_ANCA_SRC4 | INT_MASK_VIP1_MULT_ANCA_SRC3 | INT_MASK_VIP1_MULT_ANCA_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP1_MULT_ANCA_SRC1 | INT_MASK_VIP1_MULT_ANCA_SRC0 | INT_MASK_VIP1_MULT_PORTB_SRC15 | INT_MASK_VIP1_MULT_PORTB_SRC14 | INT_MASK_VIP1_MULT_PORTB_SRC13 | INT_MASK_VIP1_MULT_PORTB_SRC12 | INT_MASK_VIP1_MULT_PORTB_SRC11 | INT_MASK_VIP1_MULT_PORTB_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-266. VPDMA_int2_channel2_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP1_MULT_ANCB_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP1_MULT_ANCB_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP1_MULT_ANCB_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP1_MULT_ANCB_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_MULT_ANCB_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_MULT_ANCB_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP1_MULT_ANCB_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP1_MULT_ANCB_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP1_MULT_ANCB_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP1_MULT_ANCB_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP1_MULT_ANCA_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-266. VPDMA_int2_channel2_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP1_MULT_ANCA_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP1_MULT_ANCA_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP1_MULT_ANCA_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP1_MULT_ANCA_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP1_MULT_ANCA_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP1_MULT_ANCA_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP1_MULT_ANCA_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP1_MULT_ANCA_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP1_MULT_ANCA_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_MULT_ANCA_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_MULT_ANCA_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_MULT_ANCA_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_MULT_ANCA_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_MULT_ANCA_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_MULT_ANCA_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_MULT_PORTB_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 15 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_MULT_PORTB_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 14 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_MULT_PORTB_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 13 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-266. VPDMA_int2_channel2_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP1_MULT_PORTB_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 12 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP1_MULT_PORTB_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 11 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP1_MULT_PORTB_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 10 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.59 VPDMA_int2_channel3_int_stat Register (offset = F8h) [reset = 0h]

VPDMA_int2_channel3_int_stat is shown in Figure 12-354 and described in Table 12-267.

**Figure 12-354. VPDMA_int2_channel3_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP2_MULT_PORTB_SRC3 | INT_STAT_VIP2_MULT_PORTB_SRC2 | INT_STAT_VIP2_MULT_PORTB_SRC1 | INT_STAT_VIP2_MULT_PORTB_SRC0 | INT_STAT_VIP2_MULT_PORTA_SRC15 | INT_STAT_VIP2_MULT_PORTA_SRC14 | INT_STAT_VIP2_MULT_PORTA_SRC13 | INT_STAT_VIP2_MULT_PORTA_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP2_MULT_PORTA_SRC11 | INT_STAT_VIP2_MULT_PORTA_SRC10 | INT_STAT_VIP2_MULT_PORTA_SRC9 | INT_STAT_VIP2_MULT_PORTA_SRC8 | INT_STAT_VIP2_MULT_PORTA_SRC7 | INT_STAT_VIP2_MULT_PORTA_SRC6 | INT_STAT_VIP2_MULT_PORTA_SRC5 | INT_STAT_VIP2_MULT_PORTA_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_MULT_PORTA_SRC3 | INT_STAT_VIP2_MULT_PORTA_SRC2 | INT_STAT_VIP2_MULT_PORTA_SRC1 | INT_STAT_VIP2_MULT_PORTA_SRC0 | INT_STAT_VIP1_PORTB_RGB | INT_STAT_VIP1_PORTA_RGB | INT_STAT_VIP1_PORTB_CHROMA | INT_STAT_VIP1_PORTB_LUMA |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_PORTA_CHROMA | INT_STAT_VIP1_PORTA_LUMA | INT_STAT_VIP1_MULT_ANCB_SRC15 | INT_STAT_VIP1_MULT_ANCB_SRC14 | INT_STAT_VIP1_MULT_ANCB_SRC13 | INT_STAT_VIP1_MULT_ANCB_SRC12 | INT_STAT_VIP1_MULT_ANCB_SRC11 | INT_STAT_VIP1_MULT_ANCB_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-267. VPDMA_int2_channel3_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP2_MULT_PORTB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP2_MULT_PORTB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP2_MULT_PORTB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_MULT_PORTB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP2_MULT_PORTA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP2_MULT_PORTA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-267. VPDMA_int2_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP2_MULT_PORTA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP2_MULT_PORTA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP2_MULT_PORTA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP2_MULT_PORTA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP2_MULT_PORTA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP2_MULT_PORTA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP2_MULT_PORTA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP2_MULT_PORTA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_MULT_PORTA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP2_MULT_PORTA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-267. VPDMA_int2_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP2_MULT_PORTA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_MULT_PORTA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_MULT_PORTA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_MULT_PORTA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_PORTB_RGB | W | 0h | The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_PORTA_RGB | W | 0h | The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_PORTB_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_PORTB_LUMA | W | 0h | The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_PORTA_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_PORTA_LUMA | W | 0h | The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-267. VPDMA_int2_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_MULT_ANCB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_MULT_ANCB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_MULT_ANCB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_MULT_ANCB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP1_MULT_ANCB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP1_MULT_ANCB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.60 VPDMA_int2_channel3_int_mask Register (offset = FCh) [reset = 0h]

VPDMA_int2_channel3_int_mask is shown in Figure 12-355 and described in Table 12-268.

**Figure 12-355. VPDMA_int2_channel3_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP2_MULT_PORTB_SRC3 | INT_MASK_VIP2_MULT_PORTB_SRC2 | INT_MASK_VIP2_MULT_PORTB_SRC1 | INT_MASK_VIP2_MULT_PORTB_SRC0 | INT_MASK_VIP2_MULT_PORTA_SRC15 | INT_MASK_VIP2_MULT_PORTA_SRC14 | INT_MASK_VIP2_MULT_PORTA_SRC13 | INT_MASK_VIP2_MULT_PORTA_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP2_MULT_PORTA_SRC11 | INT_MASK_VIP2_MULT_PORTA_SRC10 | INT_MASK_VIP2_MULT_PORTA_SRC9 | INT_MASK_VIP2_MULT_PORTA_SRC8 | INT_MASK_VIP2_MULT_PORTA_SRC7 | INT_MASK_VIP2_MULT_PORTA_SRC6 | INT_MASK_VIP2_MULT_PORTA_SRC5 | INT_MASK_VIP2_MULT_PORTA_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_MULT_PORTA_SRC3 | INT_MASK_VIP2_MULT_PORTA_SRC2 | INT_MASK_VIP2_MULT_PORTA_SRC1 | INT_MASK_VIP2_MULT_PORTA_SRC0 | INT_MASK_VIP1_PORTB_RGB | INT_MASK_VIP1_PORTA_RGB | INT_MASK_VIP1_PORTB_CHROMA | INT_MASK_VIP1_PORTB_LUMA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP1_PORTA_CHROMA | INT_MASK_VIP1_PORTA_LUMA | INT_MASK_VIP1_MULT_ANCB_SRC15 | INT_MASK_VIP1_MULT_ANCB_SRC14 | INT_MASK_VIP1_MULT_ANCB_SRC13 | INT_MASK_VIP1_MULT_ANCB_SRC12 | INT_MASK_VIP1_MULT_ANCB_SRC11 | INT_MASK_VIP1_MULT_ANCB_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-268. VPDMA_int2_channel3_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP2_MULT_PORTB_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 3 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP2_MULT_PORTB_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 2 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP2_MULT_PORTB_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 1 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_MULT_PORTB_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 0 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP2_MULT_PORTA_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 15 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP2_MULT_PORTA_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 14 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP2_MULT_PORTA_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 13 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP2_MULT_PORTA_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 12 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP2_MULT_PORTA_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 11 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP2_MULT_PORTA_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 10 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP2_MULT_PORTA_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 9 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-268. VPDMA_int2_channel3_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP2_MULT_PORTA_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 8 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP2_MULT_PORTA_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 7 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP2_MULT_PORTA_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 6 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_MULT_PORTA_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 5 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_MULT_PORTA_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 4 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_MULT_PORTA_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 3 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_MULT_PORTA_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 2 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_MULT_PORTA_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 1 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_MULT_PORTA_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 0 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_PORTB_RGB | R/W | 0h | The interrupt for Video Input 1 Port B RGB Data should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_PORTA_RGB | R/W | 0h | The interrupt for Video Input 1 Port A RGB Data should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_PORTB_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_PORTB_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_PORTA_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_PORTA_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_MULT_ANCB_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_MULT_ANCB_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_MULT_ANCB_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-268. VPDMA_int2_channel3_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP1_MULT_ANCB_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP1_MULT_ANCB_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP1_MULT_ANCB_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.61 VPDMA_int2_channel4_int_stat Register (offset = 100h) [reset = 0h]

VPDMA_int2_channel4_int_stat is shown in Figure 12-356 and described in Table 12-269.

**Figure 12-356. VPDMA_int2_channel4_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP2_MULT_ANCB_SRC3 | INT_STAT_VIP2_MULT_ANCB_SRC2 | INT_STAT_VIP2_MULT_ANCB_SRC1 | INT_STAT_VIP2_MULT_ANCB_SRC0 | INT_STAT_VIP2_MULT_ANCA_SRC15 | INT_STAT_VIP2_MULT_ANCA_SRC14 | INT_STAT_VIP2_MULT_ANCA_SRC13 | INT_STAT_VIP2_MULT_ANCA_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP2_MULT_ANCA_SRC11 | INT_STAT_VIP2_MULT_ANCA_SRC10 | INT_STAT_VIP2_MULT_ANCA_SRC9 | INT_STAT_VIP2_MULT_ANCA_SRC8 | INT_STAT_VIP2_MULT_ANCA_SRC7 | INT_STAT_VIP2_MULT_ANCA_SRC6 | INT_STAT_VIP2_MULT_ANCA_SRC5 | INT_STAT_VIP2_MULT_ANCA_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_MULT_ANCA_SRC3 | INT_STAT_VIP2_MULT_ANCA_SRC2 | INT_STAT_VIP2_MULT_ANCA_SRC1 | INT_STAT_VIP2_MULT_ANCA_SRC0 | INT_STAT_VIP2_MULT_PORTB_SRC15 | INT_STAT_VIP2_MULT_PORTB_SRC14 | INT_STAT_VIP2_MULT_PORTB_SRC13 | INT_STAT_VIP2_MULT_PORTB_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_MULT_PORTB_SRC11 | INT_STAT_VIP2_MULT_PORTB_SRC10 | INT_STAT_VIP2_MULT_PORTB_SRC9 | INT_STAT_VIP2_MULT_PORTB_SRC8 | INT_STAT_VIP2_MULT_PORTB_SRC7 | INT_STAT_VIP2_MULT_PORTB_SRC6 | INT_STAT_VIP2_MULT_PORTB_SRC5 | INT_STAT_VIP2_MULT_PORTB_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-269. VPDMA_int2_channel4_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP2_MULT_ANCB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP2_MULT_ANCB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP2_MULT_ANCB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_MULT_ANCB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP2_MULT_ANCA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP2_MULT_ANCA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-269. VPDMA_int2_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP2_MULT_ANCA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP2_MULT_ANCA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP2_MULT_ANCA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP2_MULT_ANCA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP2_MULT_ANCA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP2_MULT_ANCA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP2_MULT_ANCA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP2_MULT_ANCA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_MULT_ANCA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP2_MULT_ANCA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-269. VPDMA_int2_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP2_MULT_ANCA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_MULT_ANCA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_MULT_ANCA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_MULT_ANCA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP2_MULT_PORTB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP2_MULT_PORTB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_MULT_PORTB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_MULT_PORTB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_MULT_PORTB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP2_MULT_PORTB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-269. VPDMA_int2_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP2_MULT_PORTB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP2_MULT_PORTB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP2_MULT_PORTB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP2_MULT_PORTB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP2_MULT_PORTB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP2_MULT_PORTB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.62 VPDMA_int2_channel4_int_mask Register (offset = 104h) [reset = 0h]

VPDMA_int2_channel4_int_mask is shown in Figure 12-357 and described in Table 12-270.

**Figure 12-357. VPDMA_int2_channel4_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP2_MULT_ANCB_SRC3 | INT_MASK_VIP2_MULT_ANCB_SRC2 | INT_MASK_VIP2_MULT_ANCB_SRC1 | INT_MASK_VIP2_MULT_ANCB_SRC0 | INT_MASK_VIP2_MULT_ANCA_SRC15 | INT_MASK_VIP2_MULT_ANCA_SRC14 | INT_MASK_VIP2_MULT_ANCA_SRC13 | INT_MASK_VIP2_MULT_ANCA_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP2_MULT_ANCA_SRC11 | INT_MASK_VIP2_MULT_ANCA_SRC10 | INT_MASK_VIP2_MULT_ANCA_SRC9 | INT_MASK_VIP2_MULT_ANCA_SRC8 | INT_MASK_VIP2_MULT_ANCA_SRC7 | INT_MASK_VIP2_MULT_ANCA_SRC6 | INT_MASK_VIP2_MULT_ANCA_SRC5 | INT_MASK_VIP2_MULT_ANCA_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_MULT_ANCA_SRC3 | INT_MASK_VIP2_MULT_ANCA_SRC2 | INT_MASK_VIP2_MULT_ANCA_SRC1 | INT_MASK_VIP2_MULT_ANCA_SRC0 | INT_MASK_VIP2_MULT_PORTB_SRC15 | INT_MASK_VIP2_MULT_PORTB_SRC14 | INT_MASK_VIP2_MULT_PORTB_SRC13 | INT_MASK_VIP2_MULT_PORTB_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_MULT_PORTB_SRC11 | INT_MASK_VIP2_MULT_PORTB_SRC10 | INT_MASK_VIP2_MULT_PORTB_SRC9 | INT_MASK_VIP2_MULT_PORTB_SRC8 | INT_MASK_VIP2_MULT_PORTB_SRC7 | INT_MASK_VIP2_MULT_PORTB_SRC6 | INT_MASK_VIP2_MULT_PORTB_SRC5 | INT_MASK_VIP2_MULT_PORTB_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-270. VPDMA_int2_channel4_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP2_MULT_ANCB_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP2_MULT_ANCB_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP2_MULT_ANCB_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_MULT_ANCB_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP2_MULT_ANCA_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP2_MULT_ANCA_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP2_MULT_ANCA_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP2_MULT_ANCA_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP2_MULT_ANCA_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP2_MULT_ANCA_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP2_MULT_ANCA_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-270. VPDMA_int2_channel4_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP2_MULT_ANCA_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP2_MULT_ANCA_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP2_MULT_ANCA_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_MULT_ANCA_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_MULT_ANCA_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_MULT_ANCA_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_MULT_ANCA_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_MULT_ANCA_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_MULT_ANCA_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP2_MULT_PORTB_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 15 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP2_MULT_PORTB_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 14 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_MULT_PORTB_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 13 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_MULT_PORTB_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 12 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_MULT_PORTB_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 11 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_MULT_PORTB_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 10 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP2_MULT_PORTB_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 9 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP2_MULT_PORTB_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 8 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP2_MULT_PORTB_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 7 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-270. VPDMA_int2_channel4_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP2_MULT_PORTB_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 6 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP2_MULT_PORTB_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 5 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP2_MULT_PORTB_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 4 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.63 VPDMA_int2_channel5_int_stat Register (offset = 108h) [reset = 0h]

VPDMA_int2_channel5_int_stat is shown in Figure 12-358 and described in Table 12-271.

**Figure 12-358. VPDMA_int2_channel5_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_TRANSCODE2_CHROMA | INT_STAT_TRANSCODE2_LUMA | INT_STAT_TRANSCODE1_CHROMA | INT_STAT_TRANSCODE1_LUMA | INT_STAT_AUX_IN | INT_STAT_PIP_FRAME | INT_STAT_POST_COMP_WR | INT_STAT_VBI_SD_VENC |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | INT_STAT_NF_LAST_CHROMA | INT_STAT_NF_LAST_LUMA | INT_STAT_NF_WRITE_CHROMA | INT_STAT_NF_WRITE_LUMA | INT_STAT_NF_READ | INT_STAT_VIP2_PORTB_RGB | INT_STAT_VIP2_PORTA_RGB |
| R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_PORTB_CHROMA | INT_STAT_VIP2_PORTB_LUMA | INT_STAT_VIP2_PORTA_CHROMA | INT_STAT_VIP2_PORTA_LUMA | INT_STAT_VIP2_MULT_ANCB_SRC15 | INT_STAT_VIP2_MULT_ANCB_SRC14 | INT_STAT_VIP2_MULT_ANCB_SRC13 | INT_STAT_VIP2_MULT_ANCB_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_MULT_ANCB_SRC11 | INT_STAT_VIP2_MULT_ANCB_SRC10 | INT_STAT_VIP2_MULT_ANCB_SRC9 | INT_STAT_VIP2_MULT_ANCB_SRC8 | INT_STAT_VIP2_MULT_ANCB_SRC7 | INT_STAT_VIP2_MULT_ANCB_SRC6 | INT_STAT_VIP2_MULT_ANCB_SRC5 | INT_STAT_VIP2_MULT_ANCB_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-271. VPDMA_int2_channel5_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_TRANSCODE2_CHROMA | W | 0h | The last read DMA transaction has occurred for channel transcode2_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans2_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_TRANSCODE2_LUMA | W | 0h | The last read DMA transaction has occurred for channel transcode2_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans2_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_TRANSCODE1_CHROMA | W | 0h | The last read DMA transaction has occurred for channel transcode1_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans1_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_TRANSCODE1_LUMA | W | 0h | The last read DMA transaction has occurred for channel transcode1_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans1_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_AUX_IN | W | 0h | The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-271. VPDMA_int2_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 26 | INT_STAT_PIP_FRAME | W | 0h | The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_POST_COMP_WR | W | 0h | The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VBI_SD_VENC | W | 0h | The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | Reserved | R | 0h | |
| 22 | INT_STAT_NF_LAST_CHROMA | W | 0h | The last read DMA transaction has occurred for channel nf_last_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_420_uv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_NF_LAST_LUMA | W | 0h | The last read DMA transaction has occurred for channel nf_last_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_420_y_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_NF_WRITE_CHROMA | W | 0h | The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_NF_WRITE_LUMA | W | 0h | The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_NF_READ | W | 0h | The last read DMA transaction has occurred for channel nf_read and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_422_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_PORTB_RGB | W | 0h | The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-271. VPDMA_int2_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | INT_STAT_VIP2_PORTA_RGB | W | 0h | The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_VIP2_PORTB_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_PORTB_LUMA | W | 0h | The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_PORTA_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_PORTA_LUMA | W | 0h | The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP2_MULT_ANCB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP2_MULT_ANCB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_MULT_ANCB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_MULT_ANCB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_MULT_ANCB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-271. VPDMA_int2_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | INT_STAT_VIP2_MULT_ANCB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 5 | INT_STAT_VIP2_MULT_ANCB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP2_MULT_ANCB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP2_MULT_ANCB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP2_MULT_ANCB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP2_MULT_ANCB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP2_MULT_ANCB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.64 VPDMA_int2_channel5_int_mask Register (offset = 10Ch) [reset = 0h]

VPDMA_int2_channel5_int_mask is shown in Figure 12-359 and described in Table 12-272.

**Figure 12-359. VPDMA_int2_channel5_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_TRANSCODE2_CHROMA | INT_MASK_TRANSCODE2_LUMA | INT_MASK_TRANSCODE1_CHROMA | INT_MASK_TRANSCODE1_LUMA | INT_MASK_AUX_IN | INT_MASK_PIP_FRAME | INT_MASK_POST_COMP_WR | INT_MASK_VBI_SD_VENC |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | INT_MASK_NF_LAST_CHROMA | INT_MASK_NF_LAST_LUMA | INT_MASK_NF_WRITE_CHROMA | INT_MASK_NF_WRITE_LUMA | INT_MASK_NF_READ | INT_MASK_VIP2_PORTB_RGB | INT_MASK_VIP2_PORTA_RGB |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_PORTB_CHROMA | INT_MASK_VIP2_PORTB_LUMA | INT_MASK_VIP2_PORTA_CHROMA | INT_MASK_VIP2_PORTA_LUMA | INT_MASK_VIP2_MULT_ANCB_SRC15 | INT_MASK_VIP2_MULT_ANCB_SRC14 | INT_MASK_VIP2_MULT_ANCB_SRC13 | INT_MASK_VIP2_MULT_ANCB_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_MULT_ANCB_SRC11 | INT_MASK_VIP2_MULT_ANCB_SRC10 | INT_MASK_VIP2_MULT_ANCB_SRC9 | INT_MASK_VIP2_MULT_ANCB_SRC8 | INT_MASK_VIP2_MULT_ANCB_SRC7 | INT_MASK_VIP2_MULT_ANCB_SRC6 | INT_MASK_VIP2_MULT_ANCB_SRC5 | INT_MASK_VIP2_MULT_ANCB_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-272. VPDMA_int2_channel5_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_TRANSCODE2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_TRANSCODE2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_TRANSCODE1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_TRANSCODE1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_AUX_IN | R/W | 0h | The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_PIP_FRAME | R/W | 0h | The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_POST_COMP_WR | R/W | 0h | The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VBI_SD_VENC | R/W | 0h | The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | Reserved | R | 0h | |
| 22 | INT_MASK_NF_LAST_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_NF_LAST_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-272. VPDMA_int2_channel5_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_NF_WRITE_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_NF_WRITE_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_NF_READ | R/W | 0h | The interrupt for Noise Filter Input Data 422 Interleaved should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_PORTB_RGB | R/W | 0h | The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_PORTA_RGB | R/W | 0h | The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_PORTB_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_PORTB_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_PORTA_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_PORTA_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP2_MULT_ANCB_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP2_MULT_ANCB_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_MULT_ANCB_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_MULT_ANCB_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_MULT_ANCB_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_MULT_ANCB_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP2_MULT_ANCB_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP2_MULT_ANCB_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP2_MULT_ANCB_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-272. VPDMA_int2_channel5_int_mask Register Field Descriptions (continued)**

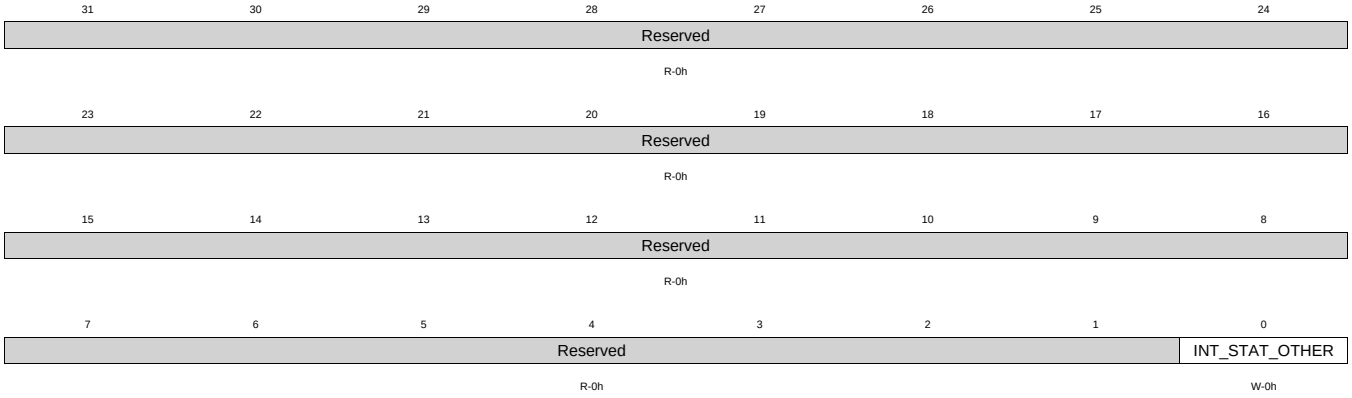
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP2_MULT_ANCB_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP2_MULT_ANCB_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP2_MULT_ANCB_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.65 VPDMA_int2_channel6_int_stat Register (offset = 110h) [reset = 0h]

VPDMA_int2_channel6_int_stat is shown in Figure 12-360 and described in Table 12-273.

**Figure 12-360. VPDMA_int2_channel6_int_stat Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-273. VPDMA_int2_channel6_int_stat Register Field Descriptions**

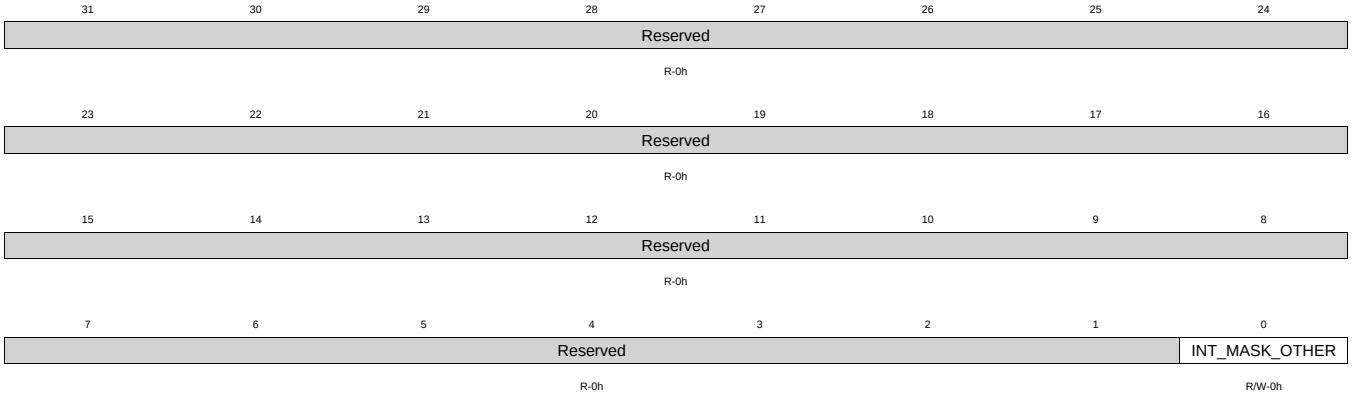
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | INT_STAT_OTHER | W | 0h | This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.66 VPDMA_int2_channel6_int_mask Register (offset = 114h) [reset = 0h]

VPDMA_int2_channel6_int_mask is shown in Figure 12-361 and described in Table 12-274.

**Figure 12-361. VPDMA_int2_channel6_int_mask Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-274. VPDMA_int2_channel6_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | INT_MASK_OTHER | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.67 VPDMA_int2_client0_int_stat Register (offset = 118h) [reset = 0h]

VPDMA_int2_client0_int_stat is shown in Figure 12-362 and described in Table 12-275.

**Figure 12-362. VPDMA_int2_client0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_GRPX1_DATA | INT_STAT_COMP_WRBK | INT_STAT_SC_OUT | Reserved | | | | |
| W-0h | W-0h | W-0h | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | INT_STAT_SC_IN_LUMA | INT_STAT_SC_IN_CHROMA | INT_STAT_PIP_WRBK | INT_STAT_DEI_SC_OUT | Reserved |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_DEI_HQ_MV_OUT | Reserved | | INT_STAT_DEI_HQ_MV_IN | Reserved | | | |
| W-0h | R-0h | | W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_STAT_DEI_HQ_3_CHROMA | INT_STAT_DEI_HQ_3_LUMA | INT_STAT_DEI_HQ_2_CHROMA | INT_STAT_DEI_HQ_2_LUMA | INT_STAT_DEI_HQ_1_LUMA | INT_STAT_DEI_HQ_1_CHROMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-275. VPDMA_int2_client0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_GRPX1_DATA | W | 0h | The client interface grpx1_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_COMP_WRBK | W | 0h | The client interface comp_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_SC_OUT | W | 0h | The client interface sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28-21 | Reserved | R | 0h | |
| 20 | INT_STAT_SC_IN_LUMA | W | 0h | The client interface sc_in_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_SC_IN_CHROMA | W | 0h | The client interface sc_in_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-275. VPDMA_int2_client0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | INT_STAT_PIP_WRBK | W | 0h | The client interface pip_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_DEI_SC_OUT | W | 0h | The client interface dei_sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | Reserved | R | 0h | |
| 15 | INT_STAT_DEI_HQ_MV_OUT | W | 0h | The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_STAT_DEI_HQ_MV_IN | W | 0h | The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11-6 | Reserved | R | 0h | |
| 5 | INT_STAT_DEI_HQ_3_CHROMA | W | 0h | The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_DEI_HQ_3_LUMA | W | 0h | The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_DEI_HQ_2_CHROMA | W | 0h | The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_DEI_HQ_2_LUMA | W | 0h | The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-275. VPDMA_int2_client0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | INT_STAT_DEI_HQ_1_LUMA | W | 0h | The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_DEI_HQ_1_CHROMA | W | 0h | The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.68 VPDMA_int2_client0_int_mask Register (offset = 11Ch) [reset = 0h]

VPDMA_int2_client0_int_mask is shown in Figure 12-363 and described in Table 12-276.

**Figure 12-363. VPDMA_int2_client0_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_GRPX1_DATA | INT_MASK_COMP_WRBK | INT_MASK_SC_OUT | Reserved | | | | |
| R/W-0h | R/W-0h | R/W-0h | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | INT_MASK_SC_IN_LUMA | INT_MASK_SC_IN_CHROMA | INT_MASK_PIP_WRBK | INT_MASK_DEI_SC_OUT | Reserved |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_DEI_HQ_MV_OUT | Reserved | | INT_MASK_DEI_HQ_MV_IN | Reserved | | | |
| R/W-0h | R-0h | | R/W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_MASK_DEI_HQ_3_CHROMA | INT_MASK_DEI_HQ_3_LUMA | INT_MASK_DEI_HQ_2_CHROMA | INT_MASK_DEI_HQ_2_LUMA | INT_MASK_DEI_HQ_1_LUMA | INT_MASK_DEI_HQ_1_CHROMA |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-276. VPDMA_int2_client0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_GRPX1_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_COMP_WRBK | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_SC_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28-21 | Reserved | R | 0h | |
| 20 | INT_MASK_SC_IN_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_SC_IN_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_PIP_WRBK | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_DEI_SC_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | Reserved | R | 0h | |
| 15 | INT_MASK_DEI_HQ_MV_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_MASK_DEI_HQ_MV_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11-6 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-276. VPDMA_int2_client0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_MASK_DEI_HQ_3_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_DEI_HQ_3_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_DEI_HQ_2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_DEI_HQ_2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_DEI_HQ_1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_DEI_HQ_1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.69 VPDMA_int2_client1_int_stat Register (offset = 120h) [reset = 0h]

VPDMA_int2_client1_int_stat is shown in Figure 12-364 and described in Table 12-277.

**Figure 12-364. VPDMA_int2_client1_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | INT_STAT_VIP2_ANC_B | INT_STAT_VIP2_ANC_A | INT_STAT_VIP1_ANC_B | INT_STAT_VIP1_ANC_A | INT_STAT_TRANS2_LUMA | INT_STAT_TRANS2_CHROMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_TRANS1_LUMA | INT_STAT_TRANS1_CHROMA | INT_STAT_HDMI_WRBK_OUT | INT_STAT_VPI_CTL | INT_STAT_VBI_SDVENC | Reserved | INT_STAT_NF_420_UV_OUT | INT_STAT_NF_420_Y_OUT |
| W-0h | W-0h | W-0h | W-0h | W-0h | R-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_NF_420_UV_IN | INT_STAT_NF_420_Y_IN | INT_STAT_NF_422_IN | INT_STAT_GRPX3_ST | INT_STAT_GRPX2_ST | INT_STAT_GRPX1_ST | INT_STAT_VIP2_UP_UV | INT_STAT_VIP2_UP_Y |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_LO_UV | INT_STAT_VIP2_LO_Y | INT_STAT_VIP1_UP_UV | INT_STAT_VIP1_UP_Y | INT_STAT_VIP1_LO_UV | INT_STAT_VIP1_LO_Y | INT_STAT_GRPX3_DATA | INT_STAT_GRPX2_DATA |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-277. VPDMA_int2_client1_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29 | INT_STAT_VIP2_ANC_B | W | 0h | The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_ANC_A | W | 0h | The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_ANC_B | W | 0h | The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_ANC_A | W | 0h | The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_TRANS2_LUMA | W | 0h | The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-277. VPDMA_int2_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 24 | INT_STAT_TRANS2_CHROMA | W | 0h | The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_TRANS1_LUMA | W | 0h | The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_TRANS1_CHROMA | W | 0h | The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_HDMI_WRBK_OUT | W | 0h | The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VPI_CTL | W | 0h | The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VBI_SDVENC | W | 0h | The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | Reserved | R | 0h | |
| 17 | INT_STAT_NF_420_UV_OUT | W | 0h | The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_NF_420_Y_OUT | W | 0h | The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_NF_420_UV_IN | W | 0h | The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-277. VPDMA_int2_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | INT_STAT_NF_420_Y_IN | W | 0h | The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_NF_422_IN | W | 0h | The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_GRPX3_ST | W | 0h | The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_GRPX2_ST | W | 0h | The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_GRPX1_ST | W | 0h | The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_UP_UV | W | 0h | The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_UP_Y | W | 0h | The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_LO_UV | W | 0h | The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP2_LO_Y | W | 0h | The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-277. VPDMA_int2_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_UP_UV | W | 0h | The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_UP_Y | W | 0h | The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_LO_UV | W | 0h | The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_LO_Y | W | 0h | The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_GRPX3_DATA | W | 0h | The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_GRPX2_DATA | W | 0h | The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.70 VPDMA_int2_client1_int_mask Register (offset = 124h) [reset = 0h]

VPDMA_int2_client1_int_mask is shown in Figure 12-365 and described in Table 12-278.

**Figure 12-365. VPDMA_int2_client1_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | INT_MASK_VIP2_ANC_B | INT_MASK_VIP2_ANC_A | INT_MASK_VIP1_ANC_B | INT_MASK_VIP1_ANC_A | INT_MASK_TRANS2_LUMA | INT_MASK_TRANS2_CHROMA |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_TRANS1_LUMA | INT_MASK_TRANS1_CHROMA | INT_MASK_HDMI_WRBK_OUT | INT_MASK_VPI_CTL | INT_MASK_VBI_SDVENC | Reserved | INT_MASK_NF_420_UV_OUT | INT_MASK_NF_420_Y_OUT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_NF_420_UV_IN | INT_MASK_NF_420_Y_IN | INT_MASK_NF_422_IN | INT_MASK_GRPX3_ST | INT_MASK_GRPX2_ST | INT_MASK_GRPX1_ST | INT_MASK_VIP2_UP_UV | INT_MASK_VIP2_UP_Y |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_LO_UV | INT_MASK_VIP2_LO_Y | INT_MASK_VIP1_UP_UV | INT_MASK_VIP1_UP_Y | INT_MASK_VIP1_LO_UV | INT_MASK_VIP1_LO_Y | INT_MASK_GRPX3_DATA | INT_MASK_GRPX2_DATA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-278. VPDMA_int2_client1_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29 | INT_MASK_VIP2_ANC_B | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_ANC_A | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_ANC_B | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_ANC_A | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_TRANS2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_TRANS2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_TRANS1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_TRANS1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_HDMI_WRBK_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 20 | INT_MASK_VPI_CTL | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-278. VPDMA_int2_client1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | INT_MASK_VBI_SDVENC | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | Reserved | R | 0h | |
| 17 | INT_MASK_NF_420_UV_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_NF_420_Y_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_NF_420_UV_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_NF_420_Y_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_NF_422_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_GRPX3_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_GRPX2_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_GRPX1_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_UP_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_UP_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_LO_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_LO_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_UP_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_UP_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_LO_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_VIP1_LO_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_GRPX3_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-278. VPDMA_int2_client1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | INT_MASK_GRPX2_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.71 VPDMA_int2_list0_int_stat Register (offset = 128h) [reset = 0h]

VPDMA_int2_list0_int_stat is shown in Figure 12-366 and described in Table 12-279.

**Figure 12-366. VPDMA_int2_list0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_CONTROL_DESCRIPTOR_INT15 | INT_STAT_CONTROL_DESCRIPTOR_INT14 | INT_STAT_CONTROL_DESCRIPTOR_INT13 | INT_STAT_CONTROL_DESCRIPTOR_INT12 | INT_STAT_CONTROL_DESCRIPTOR_INT11 | INT_STAT_CONTROL_DESCRIPTOR_INT10 | INT_STAT_CONTROL_DESCRIPTOR_INT9 | INT_STAT_CONTROL_DESCRIPTOR_INT8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_CONTROL_DESCRIPTOR_INT7 | INT_STAT_CONTROL_DESCRIPTOR_INT6 | INT_STAT_CONTROL_DESCRIPTOR_INT5 | INT_STAT_CONTROL_DESCRIPTOR_INT4 | INT_STAT_CONTROL_DESCRIPTOR_INT3 | INT_STAT_CONTROL_DESCRIPTOR_INT2 | INT_STAT_CONTROL_DESCRIPTOR_INT1 | INT_STAT_CONTROL_DESCRIPTOR_INT0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_LIST7_NOTIFY | INT_STAT_LIST7_COMPLETE | INT_STAT_LIST6_NOTIFY | INT_STAT_LIST6_COMPLETE | INT_STAT_LIST5_NOTIFY | INT_STAT_LIST5_COMPLETE | INT_STAT_LIST4_NOTIFY | INT_STAT_LIST4_COMPLETE |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_LIST3_NOTIFY | INT_STAT_LIST3_COMPLETE | INT_STAT_LIST2_NOTIFY | INT_STAT_LIST2_COMPLETE | INT_STAT_LIST1_NOTIFY | INT_STAT_LIST1_COMPLETE | INT_STAT_LIST0_NOTIFY | INT_STAT_LIST0_COMPLETE |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-279. VPDMA_int2_list0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_CONTROL_DESCRIPTOR_INT15 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_CONTROL_DESCRIPTOR_INT14 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_CONTROL_DESCRIPTOR_INT13 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_CONTROL_DESCRIPTOR_INT12 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_CONTROL_DESCRIPTOR_INT11 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_CONTROL_DESCRIPTOR_INT10 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_CONTROL_DESCRIPTOR_INT9 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_CONTROL_DESCRIPTOR_INT8 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-279. VPDMA_int2_list0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | INT_STAT_CONTROL_DESCRIPTOR_INT7 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_CONTROL_DESCRIPTOR_INT6 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_CONTROL_DESCRIPTOR_INT5 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_CONTROL_DESCRIPTOR_INT4 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_CONTROL_DESCRIPTOR_INT3 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_CONTROL_DESCRIPTOR_INT2 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_CONTROL_DESCRIPTOR_INT1 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_CONTROL_DESCRIPTOR_INT0 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_LIST7_NOTIFY | W | 0h | A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_LIST7_COMPLETE | W | 0h | List 7 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_LIST6_NOTIFY | W | 0h | A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_LIST6_COMPLETE | W | 0h | List 6 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_LIST5_NOTIFY | W | 0h | A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_LIST5_COMPLETE | W | 0h | List 5 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_LIST4_NOTIFY | W | 0h | A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-279. VPDMA_int2_list0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | INT_STAT_LIST4_COMPLETE | W | 0h | List 4 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_LIST3_NOTIFY | W | 0h | A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_LIST3_COMPLETE | W | 0h | List 3 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 5 | INT_STAT_LIST2_NOTIFY | W | 0h | A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_LIST2_COMPLETE | W | 0h | List 2 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_LIST1_NOTIFY | W | 0h | A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_LIST1_COMPLETE | W | 0h | List 1 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_LIST0_NOTIFY | W | 0h | A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_LIST0_COMPLETE | W | 0h | List 0 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.72 VPDMA_int2_list0_int_mask Register (offset = 12Ch) [reset = 0h]

VPDMA_int2_list0_int_mask is shown in Figure 12-367 and described in Table 12-280.

**Figure 12-367. VPDMA_int2_list0_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_CONTROL_DESCRIPTOR_INT15 | INT_MASK_CONTROL_DESCRIPTOR_INT14 | INT_MASK_CONTROL_DESCRIPTOR_INT13 | INT_MASK_CONTROL_DESCRIPTOR_INT12 | INT_MASK_CONTROL_DESCRIPTOR_INT11 | INT_MASK_CONTROL_DESCRIPTOR_INT10 | INT_MASK_CONTROL_DESCRIPTOR_INT9 | INT_MASK_CONTROL_DESCRIPTOR_INT8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_CONTROL_DESCRIPTOR_INT7 | INT_MASK_CONTROL_DESCRIPTOR_INT6 | INT_MASK_CONTROL_DESCRIPTOR_INT5 | INT_MASK_CONTROL_DESCRIPTOR_INT4 | INT_MASK_CONTROL_DESCRIPTOR_INT3 | INT_MASK_CONTROL_DESCRIPTOR_INT2 | INT_MASK_CONTROL_DESCRIPTOR_INT1 | INT_MASK_CONTROL_DESCRIPTOR_INT0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_LIST7_NOTIFY | INT_MASK_LIST7_COMPLETE | INT_MASK_LIST6_NOTIFY | INT_MASK_LIST6_COMPLETE | INT_MASK_LIST5_NOTIFY | INT_MASK_LIST5_COMPLETE | INT_MASK_LIST4_NOTIFY | INT_MASK_LIST4_COMPLETE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_LIST3_NOTIFY | INT_MASK_LIST3_COMPLETE | INT_MASK_LIST2_NOTIFY | INT_MASK_LIST2_COMPLETE | INT_MASK_LIST1_NOTIFY | INT_MASK_LIST1_COMPLETE | INT_MASK_LIST0_NOTIFY | INT_MASK_LIST0_COMPLETE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-280. VPDMA_int2_list0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_CONTROL_DESCRIPTOR_INT15 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_CONTROL_DESCRIPTOR_INT14 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_CONTROL_DESCRIPTOR_INT13 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_CONTROL_DESCRIPTOR_INT12 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_CONTROL_DESCRIPTOR_INT11 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_CONTROL_DESCRIPTOR_INT10 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_CONTROL_DESCRIPTOR_INT9 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_CONTROL_DESCRIPTOR_INT8 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_CONTROL_DESCRIPTOR_INT7 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_CONTROL_DESCRIPTOR_INT6 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-280. VPDMA_int2_list0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | INT_MASK_CONTROL_DESCRIPTOR_INT5 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 20 | INT_MASK_CONTROL_DESCRIPTOR_INT4 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_CONTROL_DESCRIPTOR_INT3 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_CONTROL_DESCRIPTOR_INT2 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_CONTROL_DESCRIPTOR_INT1 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_CONTROL_DESCRIPTOR_INT0 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_LIST7_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_LIST7_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_LIST6_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_LIST6_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_LIST5_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_LIST5_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_LIST4_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_LIST4_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_LIST3_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_LIST3_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_LIST2_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_LIST2_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-280. VPDMA_int2_list0_int_mask Register Field Descriptions (continued)**

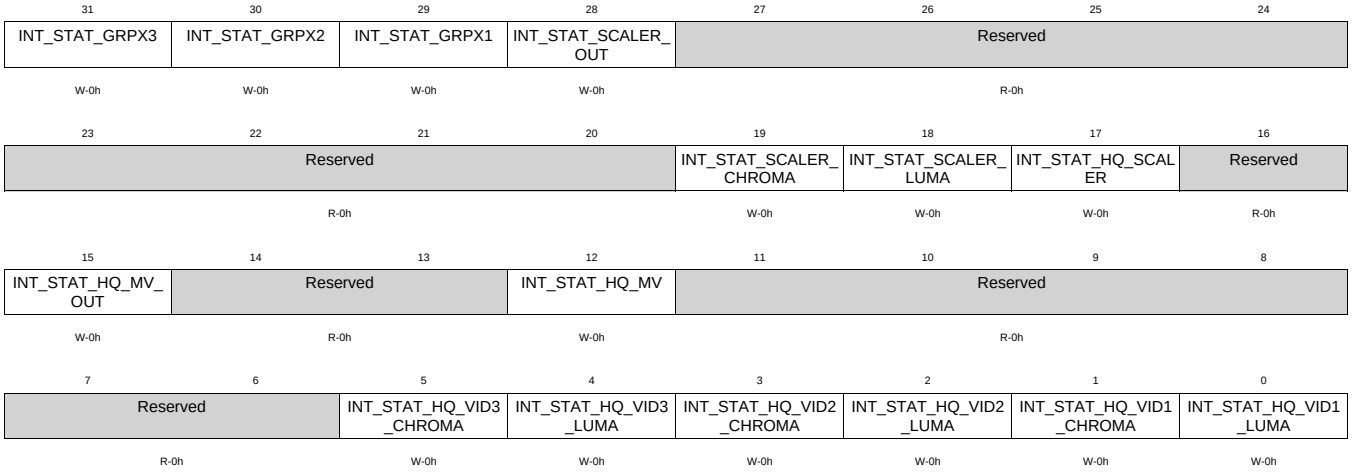
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | INT_MASK_LIST1_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_LIST1_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_LIST0_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_LIST0_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int2. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.73 VPDMA_int3_channel0_int_stat Register (offset = 130h) [reset = 0h]

VPDMA_int3_channel0_int_stat is shown in Figure 12-368 and described in Table 12-281.

**Figure 12-368. VPDMA_int3_channel0_int_stat Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-281. VPDMA_int3_channel0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_GRPX3 | W | 0h | The last read DMA transaction has occurred for channel grpx3 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_GRPX2 | W | 0h | The last read DMA transaction has occurred for channel grpx2 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_GRPX1 | W | 0h | The last read DMA transaction has occurred for channel grpx1 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_SCALER_OUT | W | 0h | The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27-20 | Reserved | R | 0h | |
| 19 | INT_STAT_SCALER_CHROMA | W | 0h | The last write DMA transaction has completed for channel scaler_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-281. VPDMA_int3_channel0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | INT_STAT_SCALER_LUMA | W | 0h | The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_HQ_SCALER | W | 0h | The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | Reserved | R | 0h | |
| 15 | INT_STAT_HQ_MV_OUT | W | 0h | The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_STAT_HQ_MV | W | 0h | The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11-6 | Reserved | R | 0h | |
| 5 | INT_STAT_HQ_VID3_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid3_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_3_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_HQ_VID3_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid3_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_3_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_HQ_VID2_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid2_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_2_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_HQ_VID2_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid2_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_2_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-281. VPDMA_int3_channel0_int_stat Register Field Descriptions (continued)**

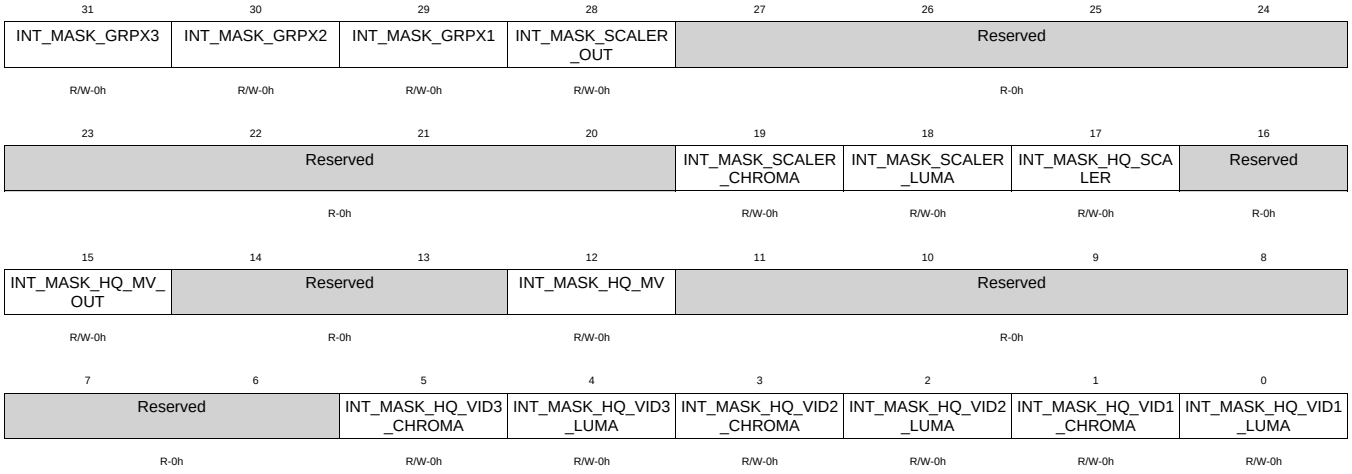
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | INT_STAT_HQ_VID1_CHROMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid1_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_1_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_HQ_VID1_LUMA | W | 0h | The last read DMA transaction has occurred for channel hq_vid1_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_1_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.74 VPDMA_int3_channel0_int_mask Register (offset = 134h) [reset = 0h]

VPDMA_int3_channel0_int_mask is shown in Figure 12-369 and described in Table 12-282.

**Figure 12-369. VPDMA_int3_channel0_int_mask Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-282. VPDMA_int3_channel0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_GRPX3 | R/W | 0h | The interrupt for Graphcis 2 Data should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_GRPX2 | R/W | 0h | The interrupt for Graphics 1 Data should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_GRPX1 | R/W | 0h | The interrupt for Graphics 0 Data should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_SCALER_OUT | R/W | 0h | The interrupt for Low Cost DEI Scalar Write to Memory should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27-20 | Reserved | R | 0h | |
| 19 | INT_MASK_SCALER_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_SCALER_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_HQ_SCALER | R/W | 0h | The interrupt for High Quality DEI Scaler Write to Memory should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | Reserved | R | 0h | |
| 15 | INT_MASK_HQ_MV_OUT | R/W | 0h | The interrupt for Low Cost DEI Motion Vector Write should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_MASK_HQ_MV | R/W | 0h | The interrupt for Low Cost DEI Motion Vector should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11-6 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-282. VPDMA_int3_channel0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_MASK_HQ_VID3_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_HQ_VID3_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_HQ_VID2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_HQ_VID2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_HQ_VID1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_HQ_VID1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.75 VPDMA_int3_channel1_int_stat Register (offset = 138h) [reset = 0h]

VPDMA_int3_channel1_int_stat is shown in Figure 12-370 and described in Table 12-283.

**Figure 12-370. VPDMA_int3_channel1_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP1_MULT_PORTB_SRC9 | INT_STAT_VIP1_MULT_PORTB_SRC8 | INT_STAT_VIP1_MULT_PORTB_SRC7 | INT_STAT_VIP1_MULT_PORTB_SRC6 | INT_STAT_VIP1_MULT_PORTB_SRC5 | INT_STAT_VIP1_MULT_PORTB_SRC4 | INT_STAT_VIP1_MULT_PORTB_SRC3 | INT_STAT_VIP1_MULT_PORTB_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP1_MULT_PORTB_SRC1 | INT_STAT_VIP1_MULT_PORTB_SRC0 | INT_STAT_VIP1_MULT_PORTA_SRC15 | INT_STAT_VIP1_MULT_PORTA_SRC14 | INT_STAT_VIP1_MULT_PORTA_SRC13 | INT_STAT_VIP1_MULT_PORTA_SRC12 | INT_STAT_VIP1_MULT_PORTA_SRC11 | INT_STAT_VIP1_MULT_PORTA_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP1_MULT_PORTA_SRC9 | INT_STAT_VIP1_MULT_PORTA_SRC8 | INT_STAT_VIP1_MULT_PORTA_SRC7 | INT_STAT_VIP1_MULT_PORTA_SRC6 | INT_STAT_VIP1_MULT_PORTA_SRC5 | INT_STAT_VIP1_MULT_PORTA_SRC4 | INT_STAT_VIP1_MULT_PORTA_SRC3 | INT_STAT_VIP1_MULT_PORTA_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_MULT_PORTA_SRC1 | INT_STAT_VIP1_MULT_PORTA_SRC0 | INT_STAT_GRPX3_CLUT | INT_STAT_GRPX2_CLUT | INT_STAT_GRPX1_CLUT | INT_STAT_GRPX3_STENCIL | INT_STAT_GRPX2_STENCIL | INT_STAT_GRPX1_STENCIL |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-283. VPDMA_int3_channel1_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP1_MULT_PORTB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP1_MULT_PORTB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP1_MULT_PORTB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP1_MULT_PORTB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_MULT_PORTB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_MULT_PORTB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-283. VPDMA_int3_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP1_MULT_PORTB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP1_MULT_PORTB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP1_MULT_PORTB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP1_MULT_PORTB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP1_MULT_PORTA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP1_MULT_PORTA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP1_MULT_PORTA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP1_MULT_PORTA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP1_MULT_PORTA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP1_MULT_PORTA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-283. VPDMA_int3_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP1_MULT_PORTA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP1_MULT_PORTA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP1_MULT_PORTA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP1_MULT_PORTA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_MULT_PORTA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_MULT_PORTA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_MULT_PORTA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_MULT_PORTA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_MULT_PORTA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_MULT_PORTA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-283. VPDMA_int3_channel1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_GRPX3_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx3_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_GRPX2_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx2_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_GRPX1_CLUT | W | 0h | The last read DMA transaction has occurred for channel grpx1_clut and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_clut_clt will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_GRPX3_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx3_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_GRPX2_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx2_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_GRPX1_STENCIL | W | 0h | The last read DMA transaction has occurred for channel grpx1_stencil and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_st will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.76 VPDMA_int3_channel1_int_mask Register (offset = 13Ch) [reset = 0h]

VPDMA_int3_channel1_int_mask is shown in Figure 12-371 and described in Table 12-284.

**Figure 12-371. VPDMA_int3_channel1_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_PORTB_SRC9 | INT_MASK_VIP1_MULT_PORTB_SRC8 | INT_MASK_VIP1_MULT_PORTB_SRC7 | INT_MASK_VIP1_MULT_PORTB_SRC6 | INT_MASK_VIP1_MULT_PORTB_SRC5 | INT_MASK_VIP1_MULT_PORTB_SRC4 | INT_MASK_VIP1_MULT_PORTB_SRC3 | INT_MASK_VIP1_MULT_PORTB_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_PORTB_SRC1 | INT_MASK_VIP1_MULT_PORTB_SRC0 | INT_MASK_VIP1_MULT_PORTA_SRC15 | INT_MASK_VIP1_MULT_PORTA_SRC14 | INT_MASK_VIP1_MULT_PORTA_SRC13 | INT_MASK_VIP1_MULT_PORTA_SRC12 | INT_MASK_VIP1_MULT_PORTA_SRC11 | INT_MASK_VIP1_MULT_PORTA_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_PORTA_SRC9 | INT_MASK_VIP1_MULT_PORTA_SRC8 | INT_MASK_VIP1_MULT_PORTA_SRC7 | INT_MASK_VIP1_MULT_PORTA_SRC6 | INT_MASK_VIP1_MULT_PORTA_SRC5 | INT_MASK_VIP1_MULT_PORTA_SRC4 | INT_MASK_VIP1_MULT_PORTA_SRC3 | INT_MASK_VIP1_MULT_PORTA_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_PORTA_SRC1 | INT_MASK_VIP1_MULT_PORTA_SRC0 | INT_MASK_GRPX3_CLUT | INT_MASK_GRPX2_CLUT | INT_MASK_GRPX1_CLUT | INT_MASK_GRPX3_STENCIL | INT_MASK_GRPX2_STENCIL | INT_MASK_GRPX1_STENCIL |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-284. VPDMA_int3_channel1_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP1_MULT_PORTB_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 9 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP1_MULT_PORTB_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 8 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP1_MULT_PORTB_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 7 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP1_MULT_PORTB_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 6 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_MULT_PORTB_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 5 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_MULT_PORTB_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 4 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP1_MULT_PORTB_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 3 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP1_MULT_PORTB_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 2 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP1_MULT_PORTB_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 1 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP1_MULT_PORTB_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 0 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP1_MULT_PORTA_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 15 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-284. VPDMA_int3_channel1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP1_MULT_PORTA_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 14 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP1_MULT_PORTA_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 13 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP1_MULT_PORTA_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 12 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP1_MULT_PORTA_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 11 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP1_MULT_PORTA_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 10 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP1_MULT_PORTA_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 9 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP1_MULT_PORTA_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 8 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP1_MULT_PORTA_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 7 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP1_MULT_PORTA_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 6 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_MULT_PORTA_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 5 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_MULT_PORTA_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 4 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_MULT_PORTA_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 3 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_MULT_PORTA_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 2 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_MULT_PORTA_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 1 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_MULT_PORTA_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port A Channel 0 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_GRPX3_CLUT | R/W | 0h | The interrupt for Graphics 2 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_GRPX2_CLUT | R/W | 0h | The interrupt for Graphics 1 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_GRPX1_CLUT | R/W | 0h | The interrupt for Graphics 0 Color Lookup Table Load from Memory should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-284. VPDMA_int3_channel1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_GRPX3_STENCIL | R/W | 0h | The interrupt for Graphics 2 Stencil should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_GRPX2_STENCIL | R/W | 0h | The interrupt for Graphics 1 Stencil should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_GRPX1_STENCIL | R/W | 0h | The interrupt for Graphics 0 Stencil should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.77 VPDMA_int3_channel2_int_stat Register (offset = 140h) [reset = 0h]

VPDMA_int3_channel2_int_stat is shown in Figure 12-372 and described in Table 12-285.

**Figure 12-372. VPDMA_int3_channel2_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP1_MULT_ANCB_SRC9 | INT_STAT_VIP1_MULT_ANCB_SRC8 | INT_STAT_VIP1_MULT_ANCB_SRC7 | INT_STAT_VIP1_MULT_ANCB_SRC6 | INT_STAT_VIP1_MULT_ANCB_SRC5 | INT_STAT_VIP1_MULT_ANCB_SRC4 | INT_STAT_VIP1_MULT_ANCB_SRC3 | INT_STAT_VIP1_MULT_ANCB_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP1_MULT_ANCB_SRC1 | INT_STAT_VIP1_MULT_ANCB_SRC0 | INT_STAT_VIP1_MULT_ANCA_SRC15 | INT_STAT_VIP1_MULT_ANCA_SRC14 | INT_STAT_VIP1_MULT_ANCA_SRC13 | INT_STAT_VIP1_MULT_ANCA_SRC12 | INT_STAT_VIP1_MULT_ANCA_SRC11 | INT_STAT_VIP1_MULT_ANCA_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP1_MULT_ANCA_SRC9 | INT_STAT_VIP1_MULT_ANCA_SRC8 | INT_STAT_VIP1_MULT_ANCA_SRC7 | INT_STAT_VIP1_MULT_ANCA_SRC6 | INT_STAT_VIP1_MULT_ANCA_SRC5 | INT_STAT_VIP1_MULT_ANCA_SRC4 | INT_STAT_VIP1_MULT_ANCA_SRC3 | INT_STAT_VIP1_MULT_ANCA_SRC2 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_MULT_ANCA_SRC1 | INT_STAT_VIP1_MULT_ANCA_SRC0 | INT_STAT_VIP1_MULT_PORTB_SRC15 | INT_STAT_VIP1_MULT_PORTB_SRC14 | INT_STAT_VIP1_MULT_PORTB_SRC13 | INT_STAT_VIP1_MULT_PORTB_SRC12 | INT_STAT_VIP1_MULT_PORTB_SRC11 | INT_STAT_VIP1_MULT_PORTB_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-285. VPDMA_int3_channel2_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP1_MULT_ANCB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP1_MULT_ANCB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP1_MULT_ANCB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP1_MULT_ANCB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_MULT_ANCB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_MULT_ANCB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-285. VPDMA_int3_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP1_MULT_ANCB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP1_MULT_ANCB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP1_MULT_ANCB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP1_MULT_ANCB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP1_MULT_ANCA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP1_MULT_ANCA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP1_MULT_ANCA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP1_MULT_ANCA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP1_MULT_ANCA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP1_MULT_ANCA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-285. VPDMA_int3_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP1_MULT_ANCA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP1_MULT_ANCA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP1_MULT_ANCA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP1_MULT_ANCA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_MULT_ANCA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_MULT_ANCA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_MULT_ANCA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_MULT_ANCA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_MULT_ANCA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_MULT_ANCA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-285. VPDMA_int3_channel2_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_MULT_PORTB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_MULT_PORTB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_MULT_PORTB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_MULT_PORTB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP1_MULT_PORTB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP1_MULT_PORTB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.78 VPDMA_int3_channel2_int_mask Register (offset = 144h) [reset = 0h]

VPDMA_int3_channel2_int_mask is shown in Figure 12-373 and described in Table 12-286.

**Figure 12-373. VPDMA_int3_channel2_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP1_MULT_ANCB_SRC9 | INT_MASK_VIP1_MULT_ANCB_SRC8 | INT_MASK_VIP1_MULT_ANCB_SRC7 | INT_MASK_VIP1_MULT_ANCB_SRC6 | INT_MASK_VIP1_MULT_ANCB_SRC5 | INT_MASK_VIP1_MULT_ANCB_SRC4 | INT_MASK_VIP1_MULT_ANCB_SRC3 | INT_MASK_VIP1_MULT_ANCB_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP1_MULT_ANCB_SRC1 | INT_MASK_VIP1_MULT_ANCB_SRC0 | INT_MASK_VIP1_MULT_ANCA_SRC15 | INT_MASK_VIP1_MULT_ANCA_SRC14 | INT_MASK_VIP1_MULT_ANCA_SRC13 | INT_MASK_VIP1_MULT_ANCA_SRC12 | INT_MASK_VIP1_MULT_ANCA_SRC11 | INT_MASK_VIP1_MULT_ANCA_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP1_MULT_ANCA_SRC9 | INT_MASK_VIP1_MULT_ANCA_SRC8 | INT_MASK_VIP1_MULT_ANCA_SRC7 | INT_MASK_VIP1_MULT_ANCA_SRC6 | INT_MASK_VIP1_MULT_ANCA_SRC5 | INT_MASK_VIP1_MULT_ANCA_SRC4 | INT_MASK_VIP1_MULT_ANCA_SRC3 | INT_MASK_VIP1_MULT_ANCA_SRC2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP1_MULT_ANCA_SRC1 | INT_MASK_VIP1_MULT_ANCA_SRC0 | INT_MASK_VIP1_MULT_PORTB_SRC15 | INT_MASK_VIP1_MULT_PORTB_SRC14 | INT_MASK_VIP1_MULT_PORTB_SRC13 | INT_MASK_VIP1_MULT_PORTB_SRC12 | INT_MASK_VIP1_MULT_PORTB_SRC11 | INT_MASK_VIP1_MULT_PORTB_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-286. VPDMA_int3_channel2_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP1_MULT_ANCB_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP1_MULT_ANCB_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP1_MULT_ANCB_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP1_MULT_ANCB_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_MULT_ANCB_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_MULT_ANCB_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP1_MULT_ANCB_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP1_MULT_ANCB_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP1_MULT_ANCB_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP1_MULT_ANCB_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP1_MULT_ANCA_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-286. VPDMA_int3_channel2_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP1_MULT_ANCA_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP1_MULT_ANCA_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP1_MULT_ANCA_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP1_MULT_ANCA_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP1_MULT_ANCA_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP1_MULT_ANCA_SRC9 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP1_MULT_ANCA_SRC8 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP1_MULT_ANCA_SRC7 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP1_MULT_ANCA_SRC6 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_MULT_ANCA_SRC5 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_MULT_ANCA_SRC4 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_MULT_ANCA_SRC3 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_MULT_ANCA_SRC2 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_MULT_ANCA_SRC1 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_MULT_ANCA_SRC0 | R/W | 0h | The interrupt for Video Input 1 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_MULT_PORTB_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 15 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_MULT_PORTB_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 14 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_MULT_PORTB_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 13 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-286. VPDMA_int3_channel2_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP1_MULT_PORTB_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 12 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP1_MULT_PORTB_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 11 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP1_MULT_PORTB_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port B Channel 10 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.79 VPDMA_int3_channel3_int_stat Register (offset = 148h) [reset = 0h]

VPDMA_int3_channel3_int_stat is shown in Figure 12-374 and described in Table 12-287.

**Figure 12-374. VPDMA_int3_channel3_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP2_MULT_PORTB_SRC3 | INT_STAT_VIP2_MULT_PORTB_SRC2 | INT_STAT_VIP2_MULT_PORTB_SRC1 | INT_STAT_VIP2_MULT_PORTB_SRC0 | INT_STAT_VIP2_MULT_PORTA_SRC15 | INT_STAT_VIP2_MULT_PORTA_SRC14 | INT_STAT_VIP2_MULT_PORTA_SRC13 | INT_STAT_VIP2_MULT_PORTA_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP2_MULT_PORTA_SRC11 | INT_STAT_VIP2_MULT_PORTA_SRC10 | INT_STAT_VIP2_MULT_PORTA_SRC9 | INT_STAT_VIP2_MULT_PORTA_SRC8 | INT_STAT_VIP2_MULT_PORTA_SRC7 | INT_STAT_VIP2_MULT_PORTA_SRC6 | INT_STAT_VIP2_MULT_PORTA_SRC5 | INT_STAT_VIP2_MULT_PORTA_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_MULT_PORTA_SRC3 | INT_STAT_VIP2_MULT_PORTA_SRC2 | INT_STAT_VIP2_MULT_PORTA_SRC1 | INT_STAT_VIP2_MULT_PORTA_SRC0 | INT_STAT_VIP1_PORTB_RGB | INT_STAT_VIP1_PORTA_RGB | INT_STAT_VIP1_PORTB_CHROMA | INT_STAT_VIP1_PORTB_LUMA |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP1_PORTA_CHROMA | INT_STAT_VIP1_PORTA_LUMA | INT_STAT_VIP1_MULT_ANCB_SRC15 | INT_STAT_VIP1_MULT_ANCB_SRC14 | INT_STAT_VIP1_MULT_ANCB_SRC13 | INT_STAT_VIP1_MULT_ANCB_SRC12 | INT_STAT_VIP1_MULT_ANCB_SRC11 | INT_STAT_VIP1_MULT_ANCB_SRC10 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-287. VPDMA_int3_channel3_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP2_MULT_PORTB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP2_MULT_PORTB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP2_MULT_PORTB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_MULT_PORTB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP2_MULT_PORTA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP2_MULT_PORTA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-287. VPDMA_int3_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP2_MULT_PORTA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP2_MULT_PORTA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP2_MULT_PORTA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP2_MULT_PORTA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP2_MULT_PORTA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP2_MULT_PORTA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP2_MULT_PORTA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP2_MULT_PORTA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_MULT_PORTA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP2_MULT_PORTA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-287. VPDMA_int3_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP2_MULT_PORTA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_MULT_PORTA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_MULT_PORTA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_MULT_PORTA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP1_PORTB_RGB | W | 0h | The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP1_PORTA_RGB | W | 0h | The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP1_PORTB_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP1_PORTB_LUMA | W | 0h | The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP1_PORTA_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP1_PORTA_LUMA | W | 0h | The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-287. VPDMA_int3_channel3_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_MULT_ANCB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_MULT_ANCB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_MULT_ANCB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_MULT_ANCB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP1_MULT_ANCB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP1_MULT_ANCB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.80 VPDMA_int3_channel3_int_mask Register (offset = 14Ch) [reset = 0h]

VPDMA_int3_channel3_int_mask is shown in Figure 12-375 and described in Table 12-288.

**Figure 12-375. VPDMA_int3_channel3_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP2_MULT_PORTB_SRC3 | INT_MASK_VIP2_MULT_PORTB_SRC2 | INT_MASK_VIP2_MULT_PORTB_SRC1 | INT_MASK_VIP2_MULT_PORTB_SRC0 | INT_MASK_VIP2_MULT_PORTA_SRC15 | INT_MASK_VIP2_MULT_PORTA_SRC14 | INT_MASK_VIP2_MULT_PORTA_SRC13 | INT_MASK_VIP2_MULT_PORTA_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP2_MULT_PORTA_SRC11 | INT_MASK_VIP2_MULT_PORTA_SRC10 | INT_MASK_VIP2_MULT_PORTA_SRC9 | INT_MASK_VIP2_MULT_PORTA_SRC8 | INT_MASK_VIP2_MULT_PORTA_SRC7 | INT_MASK_VIP2_MULT_PORTA_SRC6 | INT_MASK_VIP2_MULT_PORTA_SRC5 | INT_MASK_VIP2_MULT_PORTA_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_MULT_PORTA_SRC3 | INT_MASK_VIP2_MULT_PORTA_SRC2 | INT_MASK_VIP2_MULT_PORTA_SRC1 | INT_MASK_VIP2_MULT_PORTA_SRC0 | INT_MASK_VIP1_PORTB_RGB | INT_MASK_VIP1_PORTA_RGB | INT_MASK_VIP1_PORTB_CHROMA | INT_MASK_VIP1_PORTB_LUMA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP1_PORTA_CHROMA | INT_MASK_VIP1_PORTA_LUMA | INT_MASK_VIP1_MULT_ANCB_SRC15 | INT_MASK_VIP1_MULT_ANCB_SRC14 | INT_MASK_VIP1_MULT_ANCB_SRC13 | INT_MASK_VIP1_MULT_ANCB_SRC12 | INT_MASK_VIP1_MULT_ANCB_SRC11 | INT_MASK_VIP1_MULT_ANCB_SRC10 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-288. VPDMA_int3_channel3_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP2_MULT_PORTB_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 3 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP2_MULT_PORTB_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 2 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP2_MULT_PORTB_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 1 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_MULT_PORTB_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 0 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP2_MULT_PORTA_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 15 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP2_MULT_PORTA_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 14 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP2_MULT_PORTA_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 13 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP2_MULT_PORTA_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 12 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP2_MULT_PORTA_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 11 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP2_MULT_PORTA_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 10 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP2_MULT_PORTA_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 9 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-288. VPDMA_int3_channel3_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP2_MULT_PORTA_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 8 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP2_MULT_PORTA_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 7 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP2_MULT_PORTA_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 6 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_MULT_PORTA_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 5 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_MULT_PORTA_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 4 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_MULT_PORTA_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 3 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_MULT_PORTA_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 2 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_MULT_PORTA_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 1 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_MULT_PORTA_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port A Channel 0 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP1_PORTB_RGB | R/W | 0h | The interrupt for Video Input 1 Port B RGB Data should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP1_PORTA_RGB | R/W | 0h | The interrupt for Video Input 1 Port A RGB Data should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP1_PORTB_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP1_PORTB_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP1_PORTA_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP1_PORTA_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_MULT_ANCB_SRC15 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_MULT_ANCB_SRC14 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_MULT_ANCB_SRC13 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-288. VPDMA_int3_channel3_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP1_MULT_ANCB_SRC12 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP1_MULT_ANCB_SRC11 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP1_MULT_ANCB_SRC10 | R/W | 0h | The interrupt for Video Input 1 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.81 VPDMA_int3_channel4_int_stat Register (offset = 150h) [reset = 0h]

VPDMA_int3_channel4_int_stat is shown in Figure 12-376 and described in Table 12-289.

**Figure 12-376. VPDMA_int3_channel4_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_VIP2_MULT_ANCB_SRC3 | INT_STAT_VIP2_MULT_ANCB_SRC2 | INT_STAT_VIP2_MULT_ANCB_SRC1 | INT_STAT_VIP2_MULT_ANCB_SRC0 | INT_STAT_VIP2_MULT_ANCA_SRC15 | INT_STAT_VIP2_MULT_ANCA_SRC14 | INT_STAT_VIP2_MULT_ANCA_SRC13 | INT_STAT_VIP2_MULT_ANCA_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_VIP2_MULT_ANCA_SRC11 | INT_STAT_VIP2_MULT_ANCA_SRC10 | INT_STAT_VIP2_MULT_ANCA_SRC9 | INT_STAT_VIP2_MULT_ANCA_SRC8 | INT_STAT_VIP2_MULT_ANCA_SRC7 | INT_STAT_VIP2_MULT_ANCA_SRC6 | INT_STAT_VIP2_MULT_ANCA_SRC5 | INT_STAT_VIP2_MULT_ANCA_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_MULT_ANCA_SRC3 | INT_STAT_VIP2_MULT_ANCA_SRC2 | INT_STAT_VIP2_MULT_ANCA_SRC1 | INT_STAT_VIP2_MULT_ANCA_SRC0 | INT_STAT_VIP2_MULT_PORTB_SRC15 | INT_STAT_VIP2_MULT_PORTB_SRC14 | INT_STAT_VIP2_MULT_PORTB_SRC13 | INT_STAT_VIP2_MULT_PORTB_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_MULT_PORTB_SRC11 | INT_STAT_VIP2_MULT_PORTB_SRC10 | INT_STAT_VIP2_MULT_PORTB_SRC9 | INT_STAT_VIP2_MULT_PORTB_SRC8 | INT_STAT_VIP2_MULT_PORTB_SRC7 | INT_STAT_VIP2_MULT_PORTB_SRC6 | INT_STAT_VIP2_MULT_PORTB_SRC5 | INT_STAT_VIP2_MULT_PORTB_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-289. VPDMA_int3_channel4_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_VIP2_MULT_ANCB_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_VIP2_MULT_ANCB_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_VIP2_MULT_ANCB_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_MULT_ANCB_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP2_MULT_ANCA_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP2_MULT_ANCA_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-289. VPDMA_int3_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25 | INT_STAT_VIP2_MULT_ANCA_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VIP2_MULT_ANCA_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_VIP2_MULT_ANCA_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_VIP2_MULT_ANCA_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_VIP2_MULT_ANCA_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VIP2_MULT_ANCA_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VIP2_MULT_ANCA_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_VIP2_MULT_ANCA_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_MULT_ANCA_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_VIP2_MULT_ANCA_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-289. VPDMA_int3_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | INT_STAT_VIP2_MULT_ANCA_SRC3 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_MULT_ANCA_SRC2 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_MULT_ANCA_SRC1 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_MULT_ANCA_SRC0 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP2_MULT_PORTB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP2_MULT_PORTB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_MULT_PORTB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_MULT_PORTB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_MULT_PORTB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP2_MULT_PORTB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-289. VPDMA_int3_channel4_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP2_MULT_PORTB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP2_MULT_PORTB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP2_MULT_PORTB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP2_MULT_PORTB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP2_MULT_PORTB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP2_MULT_PORTB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.82 VPDMA_int3_channel4_int_mask Register (offset = 154h) [reset = 0h]

VPDMA_int3_channel4_int_mask is shown in Figure 12-377 and described in Table 12-290.

**Figure 12-377. VPDMA_int3_channel4_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_VIP2_MULT_ANCB_SRC3 | INT_MASK_VIP2_MULT_ANCB_SRC2 | INT_MASK_VIP2_MULT_ANCB_SRC1 | INT_MASK_VIP2_MULT_ANCB_SRC0 | INT_MASK_VIP2_MULT_ANCA_SRC15 | INT_MASK_VIP2_MULT_ANCA_SRC14 | INT_MASK_VIP2_MULT_ANCA_SRC13 | INT_MASK_VIP2_MULT_ANCA_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_VIP2_MULT_ANCA_SRC11 | INT_MASK_VIP2_MULT_ANCA_SRC10 | INT_MASK_VIP2_MULT_ANCA_SRC9 | INT_MASK_VIP2_MULT_ANCA_SRC8 | INT_MASK_VIP2_MULT_ANCA_SRC7 | INT_MASK_VIP2_MULT_ANCA_SRC6 | INT_MASK_VIP2_MULT_ANCA_SRC5 | INT_MASK_VIP2_MULT_ANCA_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_MULT_ANCA_SRC3 | INT_MASK_VIP2_MULT_ANCA_SRC2 | INT_MASK_VIP2_MULT_ANCA_SRC1 | INT_MASK_VIP2_MULT_ANCA_SRC0 | INT_MASK_VIP2_MULT_PORTB_SRC15 | INT_MASK_VIP2_MULT_PORTB_SRC14 | INT_MASK_VIP2_MULT_PORTB_SRC13 | INT_MASK_VIP2_MULT_PORTB_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_MULT_PORTB_SRC11 | INT_MASK_VIP2_MULT_PORTB_SRC10 | INT_MASK_VIP2_MULT_PORTB_SRC9 | INT_MASK_VIP2_MULT_PORTB_SRC8 | INT_MASK_VIP2_MULT_PORTB_SRC7 | INT_MASK_VIP2_MULT_PORTB_SRC6 | INT_MASK_VIP2_MULT_PORTB_SRC5 | INT_MASK_VIP2_MULT_PORTB_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-290. VPDMA_int3_channel4_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_VIP2_MULT_ANCB_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_VIP2_MULT_ANCB_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_VIP2_MULT_ANCB_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_MULT_ANCB_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP2_MULT_ANCA_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP2_MULT_ANCA_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_VIP2_MULT_ANCA_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VIP2_MULT_ANCA_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_VIP2_MULT_ANCA_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_VIP2_MULT_ANCA_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_VIP2_MULT_ANCA_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-290. VPDMA_int3_channel4_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_VIP2_MULT_ANCA_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_VIP2_MULT_ANCA_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_VIP2_MULT_ANCA_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_MULT_ANCA_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_MULT_ANCA_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_MULT_ANCA_SRC3 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_MULT_ANCA_SRC2 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_MULT_ANCA_SRC1 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_MULT_ANCA_SRC0 | R/W | 0h | The interrupt for Video Input 2 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP2_MULT_PORTB_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 15 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP2_MULT_PORTB_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 14 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_MULT_PORTB_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 13 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_MULT_PORTB_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 12 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_MULT_PORTB_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 11 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_MULT_PORTB_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 10 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP2_MULT_PORTB_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 9 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP2_MULT_PORTB_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 8 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP2_MULT_PORTB_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 7 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-290. VPDMA_int3_channel4_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP2_MULT_PORTB_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 6 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP2_MULT_PORTB_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 5 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP2_MULT_PORTB_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port B Channel 4 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.83 VPDMA_int3_channel5_int_stat Register (offset = 158h) [reset = 0h]

VPDMA_int3_channel5_int_stat is shown in Figure 12-378 and described in Table 12-291.

**Figure 12-378. VPDMA_int3_channel5_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_TRANSCODE2_CHROMA | INT_STAT_TRANSCODE2_LUMA | INT_STAT_TRANSCODE1_CHROMA | INT_STAT_TRANSCODE1_LUMA | INT_STAT_AUX_IN | INT_STAT_PIP_FRAME | INT_STAT_POST_COMP_WR | INT_STAT_VBI_SD_VENC |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | INT_STAT_NF_LAST_CHROMA | INT_STAT_NF_LAST_LUMA | INT_STAT_NF_WRITE_CHROMA | INT_STAT_NF_WRITE_LUMA | INT_STAT_NF_READ | INT_STAT_VIP2_PORTB_RGB | INT_STAT_VIP2_PORTA_RGB |
| R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_VIP2_PORTB_CHROMA | INT_STAT_VIP2_PORTB_LUMA | INT_STAT_VIP2_PORTA_CHROMA | INT_STAT_VIP2_PORTA_LUMA | INT_STAT_VIP2_MULT_ANCB_SRC15 | INT_STAT_VIP2_MULT_ANCB_SRC14 | INT_STAT_VIP2_MULT_ANCB_SRC13 | INT_STAT_VIP2_MULT_ANCB_SRC12 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_MULT_ANCB_SRC11 | INT_STAT_VIP2_MULT_ANCB_SRC10 | INT_STAT_VIP2_MULT_ANCB_SRC9 | INT_STAT_VIP2_MULT_ANCB_SRC8 | INT_STAT_VIP2_MULT_ANCB_SRC7 | INT_STAT_VIP2_MULT_ANCB_SRC6 | INT_STAT_VIP2_MULT_ANCB_SRC5 | INT_STAT_VIP2_MULT_ANCB_SRC4 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-291. VPDMA_int3_channel5_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_TRANSCODE2_CHROMA | W | 0h | The last read DMA transaction has occurred for channel transcode2_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans2_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_TRANSCODE2_LUMA | W | 0h | The last read DMA transaction has occurred for channel transcode2_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans2_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_TRANSCODE1_CHROMA | W | 0h | The last read DMA transaction has occurred for channel transcode1_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans1_chroma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_TRANSCODE1_LUMA | W | 0h | The last read DMA transaction has occurred for channel transcode1_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client trans1_luma will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_AUX_IN | W | 0h | The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-291. VPDMA_int3_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 26 | INT_STAT_PIP_FRAME | W | 0h | The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_POST_COMP_WR | W | 0h | The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_VBI_SD_VENC | W | 0h | The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | Reserved | R | 0h | |
| 22 | INT_STAT_NF_LAST_CHROMA | W | 0h | The last read DMA transaction has occurred for channel nf_last_chroma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_420_uv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_NF_LAST_LUMA | W | 0h | The last read DMA transaction has occurred for channel nf_last_luma and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_420_y_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_NF_WRITE_CHROMA | W | 0h | The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_NF_WRITE_LUMA | W | 0h | The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_NF_READ | W | 0h | The last read DMA transaction has occurred for channel nf_read and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client nf_422_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_VIP2_PORTB_RGB | W | 0h | The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-291. VPDMA_int3_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | INT_STAT_VIP2_PORTA_RGB | W | 0h | The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_VIP2_PORTB_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_VIP2_PORTB_LUMA | W | 0h | The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_VIP2_PORTA_CHROMA | W | 0h | The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_VIP2_PORTA_LUMA | W | 0h | The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_VIP2_MULT_ANCB_SRC15 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_VIP2_MULT_ANCB_SRC14 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_MULT_ANCB_SRC13 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_MULT_ANCB_SRC12 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_MULT_ANCB_SRC11 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-291. VPDMA_int3_channel5_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | INT_STAT_VIP2_MULT_ANCB_SRC10 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 5 | INT_STAT_VIP2_MULT_ANCB_SRC9 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP2_MULT_ANCB_SRC8 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP2_MULT_ANCB_SRC7 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP2_MULT_ANCB_SRC6 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_VIP2_MULT_ANCB_SRC5 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_VIP2_MULT_ANCB_SRC4 | W | 0h | The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.84 VPDMA_int3_channel5_int_mask Register (offset = 15Ch) [reset = 0h]

VPDMA_int3_channel5_int_mask is shown in Figure 12-379 and described in Table 12-292.

**Figure 12-379. VPDMA_int3_channel5_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_TRANSCODE2_CHROMA | INT_MASK_TRANSCODE2_LUMA | INT_MASK_TRANSCODE1_CHROMA | INT_MASK_TRANSCODE1_LUMA | INT_MASK_AUX_IN | INT_MASK_PIP_FRAME | INT_MASK_POST_COMP_WR | INT_MASK_VBI_SD_VENC |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | INT_MASK_NF_LAST_CHROMA | INT_MASK_NF_LAST_LUMA | INT_MASK_NF_WRITE_CHROMA | INT_MASK_NF_WRITE_LUMA | INT_MASK_NF_READ | INT_MASK_VIP2_PORTB_RGB | INT_MASK_VIP2_PORTA_RGB |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_VIP2_PORTB_CHROMA | INT_MASK_VIP2_PORTB_LUMA | INT_MASK_VIP2_PORTA_CHROMA | INT_MASK_VIP2_PORTA_LUMA | INT_MASK_VIP2_MULT_ANCB_SRC15 | INT_MASK_VIP2_MULT_ANCB_SRC14 | INT_MASK_VIP2_MULT_ANCB_SRC13 | INT_MASK_VIP2_MULT_ANCB_SRC12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_MULT_ANCB_SRC11 | INT_MASK_VIP2_MULT_ANCB_SRC10 | INT_MASK_VIP2_MULT_ANCB_SRC9 | INT_MASK_VIP2_MULT_ANCB_SRC8 | INT_MASK_VIP2_MULT_ANCB_SRC7 | INT_MASK_VIP2_MULT_ANCB_SRC6 | INT_MASK_VIP2_MULT_ANCB_SRC5 | INT_MASK_VIP2_MULT_ANCB_SRC4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-292. VPDMA_int3_channel5_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_TRANSCODE2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_TRANSCODE2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_TRANSCODE1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_TRANSCODE1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_AUX_IN | R/W | 0h | The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_PIP_FRAME | R/W | 0h | The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_POST_COMP_WR | R/W | 0h | The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_VBI_SD_VENC | R/W | 0h | The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | Reserved | R | 0h | |
| 22 | INT_MASK_NF_LAST_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_NF_LAST_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-292. VPDMA_int3_channel5_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | INT_MASK_NF_WRITE_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_NF_WRITE_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_NF_READ | R/W | 0h | The interrupt for Noise Filter Input Data 422 Interleaved should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_VIP2_PORTB_RGB | R/W | 0h | The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_VIP2_PORTA_RGB | R/W | 0h | The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_VIP2_PORTB_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_VIP2_PORTB_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_VIP2_PORTA_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_VIP2_PORTA_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_VIP2_MULT_ANCB_SRC15 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_VIP2_MULT_ANCB_SRC14 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_MULT_ANCB_SRC13 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_MULT_ANCB_SRC12 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_MULT_ANCB_SRC11 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_MULT_ANCB_SRC10 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP2_MULT_ANCB_SRC9 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP2_MULT_ANCB_SRC8 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP2_MULT_ANCB_SRC7 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-292. VPDMA_int3_channel5_int_mask Register Field Descriptions (continued)**

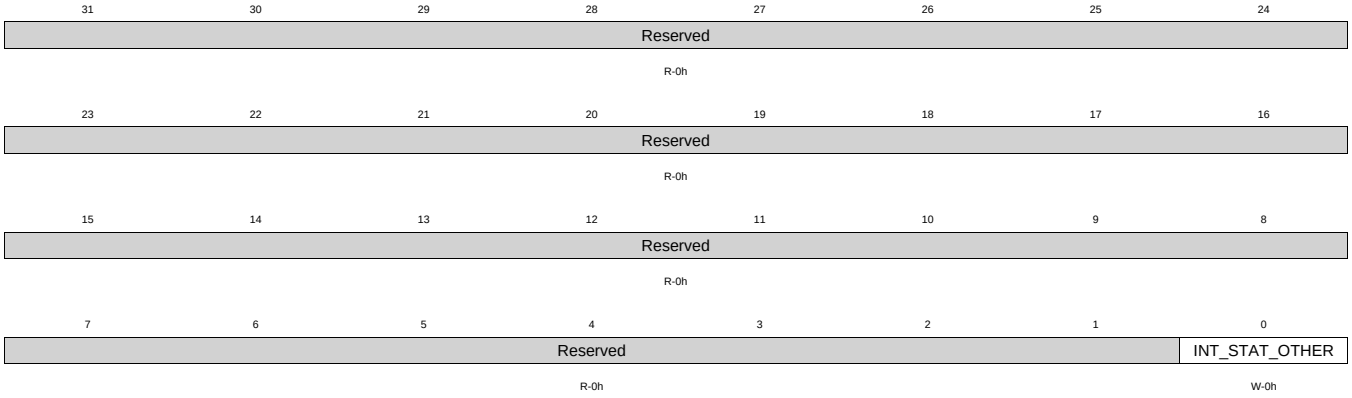
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | INT_MASK_VIP2_MULT_ANCB_SRC6 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_VIP2_MULT_ANCB_SRC5 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_VIP2_MULT_ANCB_SRC4 | R/W | 0h | The interrupt for Video Input 2 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.85 VPDMA_int3_channel6_int_stat Register (offset = 160h) [reset = 0h]

VPDMA_int3_channel6_int_stat is shown in Figure 12-380 and described in Table 12-293.

**Figure 12-380. VPDMA_int3_channel6_int_stat Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-293. VPDMA_int3_channel6_int_stat Register Field Descriptions**

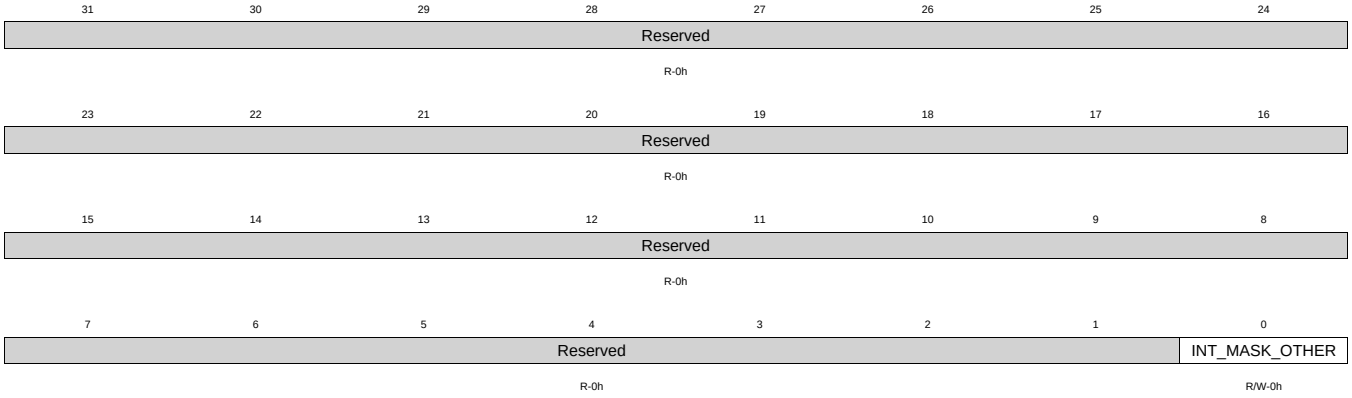
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | INT_STAT_OTHER | W | 0h | This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.86 VPDMA_int3_channel6_int_mask Register (offset = 164h) [reset = 0h]

VPDMA_int3_channel6_int_mask is shown in Figure 12-381 and described in Table 12-294.

**Figure 12-381. VPDMA_int3_channel6_int_mask Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-294. VPDMA_int3_channel6_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0h | |
| 0 | INT_MASK_OTHER | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.87 VPDMA_int3_client0_int_stat Register (offset = 168h) [reset = 0h]

VPDMA_int3_client0_int_stat is shown in Figure 12-382 and described in Table 12-295.

**Figure 12-382. VPDMA_int3_client0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_GRPX1_DATA | INT_STAT_COMP_WRBK | INT_STAT_SC_OUT | Reserved | | | | |
| W-0h | W-0h | W-0h | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | INT_STAT_SC_IN_LUMA | INT_STAT_SC_IN_CHROMA | INT_STAT_PIP_WRBK | INT_STAT_DEI_SC_OUT | Reserved |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_DEI_HQ_MV_OUT | Reserved | | INT_STAT_DEI_HQ_MV_IN | Reserved | | | |
| W-0h | R-0h | | W-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | INT_STAT_DEI_HQ_3_CHROMA | INT_STAT_DEI_HQ_3_LUMA | INT_STAT_DEI_HQ_2_CHROMA | INT_STAT_DEI_HQ_2_LUMA | INT_STAT_DEI_HQ_1_LUMA | INT_STAT_DEI_HQ_1_CHROMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-295. VPDMA_int3_client0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_GRPX1_DATA | W | 0h | The client interface grpx1_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_COMP_WRBK | W | 0h | The client interface comp_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_SC_OUT | W | 0h | The client interface sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28-21 | Reserved | R | 0h | |
| 20 | INT_STAT_SC_IN_LUMA | W | 0h | The client interface sc_in_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_SC_IN_CHROMA | W | 0h | The client interface sc_in_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-295. VPDMA_int3_client0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | INT_STAT_PIP_WRBK | W | 0h | The client interface pip_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_DEI_SC_OUT | W | 0h | The client interface dei_sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | Reserved | R | 0h | |
| 15 | INT_STAT_DEI_HQ_MV_OUT | W | 0h | The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_STAT_DEI_HQ_MV_IN | W | 0h | The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11-6 | Reserved | R | 0h | |
| 5 | INT_STAT_DEI_HQ_3_CHROMA | W | 0h | The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_DEI_HQ_3_LUMA | W | 0h | The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_DEI_HQ_2_CHROMA | W | 0h | The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_DEI_HQ_2_LUMA | W | 0h | The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-295. VPDMA_int3_client0_int_stat Register Field Descriptions (continued)**

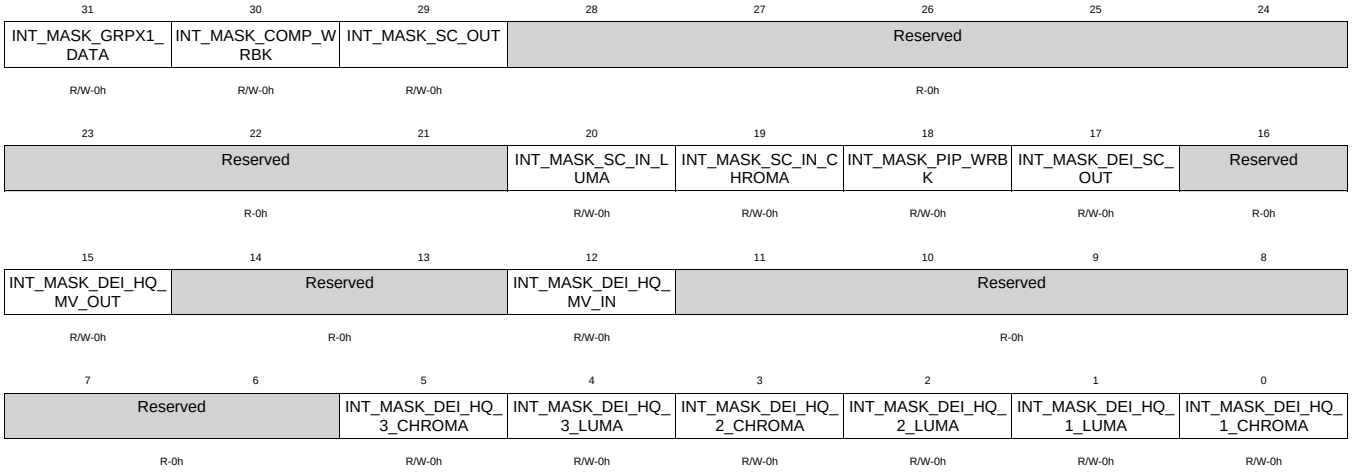
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | INT_STAT_DEI_HQ_1_LUMA | W | 0h | The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_DEI_HQ_1_CHROMA | W | 0h | The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.88 VPDMA_int3_client0_int_mask Register (offset = 16Ch) [reset = 0h]

VPDMA_int3_client0_int_mask is shown in Figure 12-383 and described in Table 12-296.

**Figure 12-383. VPDMA_int3_client0_int_mask Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-296. VPDMA_int3_client0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_GRPX1_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_COMP_WRBK | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_SC_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28-21 | Reserved | R | 0h | |
| 20 | INT_MASK_SC_IN_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_SC_IN_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_PIP_WRBK | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_DEI_SC_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | Reserved | R | 0h | |
| 15 | INT_MASK_DEI_HQ_MV_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14-13 | Reserved | R | 0h | |
| 12 | INT_MASK_DEI_HQ_MV_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11-6 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-296. VPDMA_int3_client0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_MASK_DEI_HQ_3_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_DEI_HQ_3_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_DEI_HQ_2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_DEI_HQ_2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_DEI_HQ_1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_DEI_HQ_1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.89 VPDMA_int3_client1_int_stat Register (offset = 170h) [reset = 0h]

VPDMA_int3_client1_int_stat is shown in Figure 12-384 and described in Table 12-297.

**Figure 12-384. VPDMA_int3_client1_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | INT_STAT_VIP2_ANC_B | INT_STAT_VIP2_ANC_A | INT_STAT_VIP1_ANC_B | INT_STAT_VIP1_ANC_A | INT_STAT_TRANS2_LUMA | INT_STAT_TRANS2_CHROMA |
| R-0h | | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_TRANS1_LUMA | INT_STAT_TRANS1_CHROMA | INT_STAT_HDMI_WRBK_OUT | INT_STAT_VPI_CTL | INT_STAT_VBI_SDVENC | Reserved | INT_STAT_NF_420_UV_OUT | INT_STAT_NF_420_Y_OUT |
| W-0h | W-0h | W-0h | W-0h | W-0h | R-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_NF_420_UV_IN | INT_STAT_NF_420_Y_IN | INT_STAT_NF_422_IN | INT_STAT_GRPX3_ST | INT_STAT_GRPX2_ST | INT_STAT_GRPX1_ST | INT_STAT_VIP2_UP_UV | INT_STAT_VIP2_UP_Y |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_VIP2_LO_UV | INT_STAT_VIP2_LO_Y | INT_STAT_VIP1_UP_UV | INT_STAT_VIP1_UP_Y | INT_STAT_VIP1_LO_UV | INT_STAT_VIP1_LO_Y | INT_STAT_GRPX3_DATA | INT_STAT_GRPX2_DATA |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-297. VPDMA_int3_client1_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29 | INT_STAT_VIP2_ANC_B | W | 0h | The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_VIP2_ANC_A | W | 0h | The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_VIP1_ANC_B | W | 0h | The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_VIP1_ANC_A | W | 0h | The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_TRANS2_LUMA | W | 0h | The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-297. VPDMA_int3_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 24 | INT_STAT_TRANS2_CHROMA | W | 0h | The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 23 | INT_STAT_TRANS1_LUMA | W | 0h | The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_TRANS1_CHROMA | W | 0h | The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_HDMI_WRBK_OUT | W | 0h | The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_VPI_CTL | W | 0h | The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_VBI_SDVENC | W | 0h | The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | Reserved | R | 0h | |
| 17 | INT_STAT_NF_420_UV_OUT | W | 0h | The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_NF_420_Y_OUT | W | 0h | The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_NF_420_UV_IN | W | 0h | The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-297. VPDMA_int3_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | INT_STAT_NF_420_Y_IN | W | 0h | The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_NF_422_IN | W | 0h | The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_GRPX3_ST | W | 0h | The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_GRPX2_ST | W | 0h | The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_GRPX1_ST | W | 0h | The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_VIP2_UP_UV | W | 0h | The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 8 | INT_STAT_VIP2_UP_Y | W | 0h | The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_VIP2_LO_UV | W | 0h | The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_VIP2_LO_Y | W | 0h | The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-297. VPDMA_int3_client1_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | INT_STAT_VIP1_UP_UV | W | 0h | The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_VIP1_UP_Y | W | 0h | The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_VIP1_LO_UV | W | 0h | The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_VIP1_LO_Y | W | 0h | The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_GRPX3_DATA | W | 0h | The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_GRPX2_DATA | W | 0h | The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.90 VPDMA_int3_client1_int_mask Register (offset = 174h) [reset = 0h]

VPDMA_int3_client1_int_mask is shown in Figure 12-385 and described in Table 12-298.

**Figure 12-385. VPDMA_int3_client1_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | INT_MASK_VIP2_ANC_B | INT_MASK_VIP2_ANC_A | INT_MASK_VIP1_ANC_B | INT_MASK_VIP1_ANC_A | INT_MASK_TRANS2_LUMA | INT_MASK_TRANS2_CHROMA |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_TRANS1_LUMA | INT_MASK_TRANS1_CHROMA | INT_MASK_HDMI_WRBK_OUT | INT_MASK_VPI_CTL | INT_MASK_VBI_SDVENC | Reserved | INT_MASK_NF_420_UV_OUT | INT_MASK_NF_420_Y_OUT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_NF_420_UV_IN | INT_MASK_NF_420_Y_IN | INT_MASK_NF_422_IN | INT_MASK_GRPX3_ST | INT_MASK_GRPX2_ST | INT_MASK_GRPX1_ST | INT_MASK_VIP2_UP_UV | INT_MASK_VIP2_UP_Y |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_VIP2_LO_UV | INT_MASK_VIP2_LO_Y | INT_MASK_VIP1_UP_UV | INT_MASK_VIP1_UP_Y | INT_MASK_VIP1_LO_UV | INT_MASK_VIP1_LO_Y | INT_MASK_GRPX3_DATA | INT_MASK_GRPX2_DATA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-298. VPDMA_int3_client1_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29 | INT_MASK_VIP2_ANC_B | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_VIP2_ANC_A | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_VIP1_ANC_B | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_VIP1_ANC_A | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_TRANS2_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_TRANS2_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_TRANS1_LUMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_TRANS1_CHROMA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 21 | INT_MASK_HDMI_WRBK_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 20 | INT_MASK_VPI_CTL | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-298. VPDMA_int3_client1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | INT_MASK_VBI_SDVENC | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | Reserved | R | 0h | |
| 17 | INT_MASK_NF_420_UV_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_NF_420_Y_OUT | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_NF_420_UV_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_NF_420_Y_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_NF_422_IN | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_GRPX3_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_GRPX2_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_GRPX1_ST | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_VIP2_UP_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_VIP2_UP_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_VIP2_LO_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_VIP2_LO_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_VIP1_UP_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_VIP1_UP_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 3 | INT_MASK_VIP1_LO_UV | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_VIP1_LO_Y | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_GRPX3_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-298. VPDMA_int3_client1_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | INT_MASK_GRPX2_DATA | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.91 VPDMA_int3_list0_int_stat Register (offset = 178h) [reset = 0h]

VPDMA_int3_list0_int_stat is shown in Figure 12-386 and described in Table 12-299.

**Figure 12-386. VPDMA_int3_list0_int_stat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_STAT_CONTROL_DESCRIPTOR_INT15 | INT_STAT_CONTROL_DESCRIPTOR_INT14 | INT_STAT_CONTROL_DESCRIPTOR_INT13 | INT_STAT_CONTROL_DESCRIPTOR_INT12 | INT_STAT_CONTROL_DESCRIPTOR_INT11 | INT_STAT_CONTROL_DESCRIPTOR_INT10 | INT_STAT_CONTROL_DESCRIPTOR_INT9 | INT_STAT_CONTROL_DESCRIPTOR_INT8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_STAT_CONTROL_DESCRIPTOR_INT7 | INT_STAT_CONTROL_DESCRIPTOR_INT6 | INT_STAT_CONTROL_DESCRIPTOR_INT5 | INT_STAT_CONTROL_DESCRIPTOR_INT4 | INT_STAT_CONTROL_DESCRIPTOR_INT3 | INT_STAT_CONTROL_DESCRIPTOR_INT2 | INT_STAT_CONTROL_DESCRIPTOR_INT1 | INT_STAT_CONTROL_DESCRIPTOR_INT0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_STAT_LIST7_NOTIFY | INT_STAT_LIST7_COMPLETE | INT_STAT_LIST6_NOTIFY | INT_STAT_LIST6_COMPLETE | INT_STAT_LIST5_NOTIFY | INT_STAT_LIST5_COMPLETE | INT_STAT_LIST4_NOTIFY | INT_STAT_LIST4_COMPLETE |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_STAT_LIST3_NOTIFY | INT_STAT_LIST3_COMPLETE | INT_STAT_LIST2_NOTIFY | INT_STAT_LIST2_COMPLETE | INT_STAT_LIST1_NOTIFY | INT_STAT_LIST1_COMPLETE | INT_STAT_LIST0_NOTIFY | INT_STAT_LIST0_COMPLETE |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-299. VPDMA_int3_list0_int_stat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_STAT_CONTROL_DESCRIPTOR_INT15 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 30 | INT_STAT_CONTROL_DESCRIPTOR_INT14 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 29 | INT_STAT_CONTROL_DESCRIPTOR_INT13 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 28 | INT_STAT_CONTROL_DESCRIPTOR_INT12 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 27 | INT_STAT_CONTROL_DESCRIPTOR_INT11 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 26 | INT_STAT_CONTROL_DESCRIPTOR_INT10 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 25 | INT_STAT_CONTROL_DESCRIPTOR_INT9 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 24 | INT_STAT_CONTROL_DESCRIPTOR_INT8 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-299. VPDMA_int3_list0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | INT_STAT_CONTROL_DESCRIPTOR_INT7 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 22 | INT_STAT_CONTROL_DESCRIPTOR_INT6 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 21 | INT_STAT_CONTROL_DESCRIPTOR_INT5 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 20 | INT_STAT_CONTROL_DESCRIPTOR_INT4 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 19 | INT_STAT_CONTROL_DESCRIPTOR_INT3 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 18 | INT_STAT_CONTROL_DESCRIPTOR_INT2 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 17 | INT_STAT_CONTROL_DESCRIPTOR_INT1 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 16 | INT_STAT_CONTROL_DESCRIPTOR_INT0 | W | 0h | A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 15 | INT_STAT_LIST7_NOTIFY | W | 0h | A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 14 | INT_STAT_LIST7_COMPLETE | W | 0h | List 7 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 13 | INT_STAT_LIST6_NOTIFY | W | 0h | A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 12 | INT_STAT_LIST6_COMPLETE | W | 0h | List 6 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 11 | INT_STAT_LIST5_NOTIFY | W | 0h | A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 10 | INT_STAT_LIST5_COMPLETE | W | 0h | List 5 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 9 | INT_STAT_LIST4_NOTIFY | W | 0h | A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-299. VPDMA_int3_list0_int_stat Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | INT_STAT_LIST4_COMPLETE | W | 0h | List 4 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 7 | INT_STAT_LIST3_NOTIFY | W | 0h | A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 6 | INT_STAT_LIST3_COMPLETE | W | 0h | List 3 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 5 | INT_STAT_LIST2_NOTIFY | W | 0h | A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 4 | INT_STAT_LIST2_COMPLETE | W | 0h | List 2 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 3 | INT_STAT_LIST1_NOTIFY | W | 0h | A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 2 | INT_STAT_LIST1_COMPLETE | W | 0h | List 1 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 1 | INT_STAT_LIST0_NOTIFY | W | 0h | A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |
| 0 | INT_STAT_LIST0_COMPLETE | W | 0h | List 0 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.92 VPDMA_int3_list0_int_mask Register (offset = 17Ch) [reset = 0h]

VPDMA_int3_list0_int_mask is shown in Figure 12-387 and described in Table 12-300.

**Figure 12-387. VPDMA_int3_list0_int_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INT_MASK_CONTROL_DESCRIPTOR_INT15 | INT_MASK_CONTROL_DESCRIPTOR_INT14 | INT_MASK_CONTROL_DESCRIPTOR_INT13 | INT_MASK_CONTROL_DESCRIPTOR_INT12 | INT_MASK_CONTROL_DESCRIPTOR_INT11 | INT_MASK_CONTROL_DESCRIPTOR_INT10 | INT_MASK_CONTROL_DESCRIPTOR_INT9 | INT_MASK_CONTROL_DESCRIPTOR_INT8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT_MASK_CONTROL_DESCRIPTOR_INT7 | INT_MASK_CONTROL_DESCRIPTOR_INT6 | INT_MASK_CONTROL_DESCRIPTOR_INT5 | INT_MASK_CONTROL_DESCRIPTOR_INT4 | INT_MASK_CONTROL_DESCRIPTOR_INT3 | INT_MASK_CONTROL_DESCRIPTOR_INT2 | INT_MASK_CONTROL_DESCRIPTOR_INT1 | INT_MASK_CONTROL_DESCRIPTOR_INT0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT_MASK_LIST7_NOTIFY | INT_MASK_LIST7_COMPLETE | INT_MASK_LIST6_NOTIFY | INT_MASK_LIST6_COMPLETE | INT_MASK_LIST5_NOTIFY | INT_MASK_LIST5_COMPLETE | INT_MASK_LIST4_NOTIFY | INT_MASK_LIST4_COMPLETE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT_MASK_LIST3_NOTIFY | INT_MASK_LIST3_COMPLETE | INT_MASK_LIST2_NOTIFY | INT_MASK_LIST2_COMPLETE | INT_MASK_LIST1_NOTIFY | INT_MASK_LIST1_COMPLETE | INT_MASK_LIST0_NOTIFY | INT_MASK_LIST0_COMPLETE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-300. VPDMA_int3_list0_int_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INT_MASK_CONTROL_DESCRIPTOR_INT15 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 30 | INT_MASK_CONTROL_DESCRIPTOR_INT14 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 29 | INT_MASK_CONTROL_DESCRIPTOR_INT13 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 28 | INT_MASK_CONTROL_DESCRIPTOR_INT12 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 27 | INT_MASK_CONTROL_DESCRIPTOR_INT11 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 26 | INT_MASK_CONTROL_DESCRIPTOR_INT10 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 25 | INT_MASK_CONTROL_DESCRIPTOR_INT9 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 24 | INT_MASK_CONTROL_DESCRIPTOR_INT8 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 23 | INT_MASK_CONTROL_DESCRIPTOR_INT7 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 22 | INT_MASK_CONTROL_DESCRIPTOR_INT6 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-300. VPDMA_int3_list0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | INT_MASK_CONTROL_DESCRIPTOR_INT5 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 20 | INT_MASK_CONTROL_DESCRIPTOR_INT4 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 19 | INT_MASK_CONTROL_DESCRIPTOR_INT3 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 18 | INT_MASK_CONTROL_DESCRIPTOR_INT2 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 17 | INT_MASK_CONTROL_DESCRIPTOR_INT1 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 16 | INT_MASK_CONTROL_DESCRIPTOR_INT0 | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 15 | INT_MASK_LIST7_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 14 | INT_MASK_LIST7_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 13 | INT_MASK_LIST6_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 12 | INT_MASK_LIST6_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 11 | INT_MASK_LIST5_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 10 | INT_MASK_LIST5_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 9 | INT_MASK_LIST4_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 8 | INT_MASK_LIST4_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 7 | INT_MASK_LIST3_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 6 | INT_MASK_LIST3_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 5 | INT_MASK_LIST2_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 4 | INT_MASK_LIST2_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-300. VPDMA_int3_list0_int_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | INT_MASK_LIST1_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 2 | INT_MASK_LIST1_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 1 | INT_MASK_LIST0_NOTIFY | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |
| 0 | INT_MASK_LIST0_COMPLETE | R/W | 0h | The interrupt for should generate an interrupt on interrupt vpdma_int3. Write a 1 for the interrupt event to trigger the interrupt signal. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.93 VPDMA_dei_hq_1_chroma_cstat Register (offset = 300h) [reset = 0h]

VPDMA_dei_hq_1_chroma_cstat is shown in Figure 12-388 and described in Table 12-301.

**Figure 12-388. VPDMA_dei_hq_1_chroma_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | LINE_MODE | |
| R-0h | R-0h | R/W-0h | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-301. VPDMA_dei_hq_1_chroma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-8 | LINE_MODE | R/W | 0h | Selects the output mode of the line buffer.<br>0: repeat lines twice each output data line gets 2 times the number of frame lines.<br>1: each line once with Line Buffer Disabled.. so no mirroring. Each line gets frame lines with identical data.<br>2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines.<br>3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines. |
| 7-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.94 VPDMA_dei_hq_1_luma_cstat Register (offset = 304h) [reset = 0h]

VPDMA_dei_hq_1_luma_cstat is shown in Figure 12-389 and described in Table 12-302.

**Figure 12-389. VPDMA_dei_hq_1_luma_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-302. VPDMA_dei_hq_1_luma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.95 VPDMA_dei_hq_2_luma_cstat Register (offset = 308h) [reset = 0h]

VPDMA_dei_hq_2_luma_cstat is shown in Figure 12-390 and described in Table 12-303.

**Figure 12-390. VPDMA_dei_hq_2_luma_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-303. VPDMA_dei_hq_2_luma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.96 VPDMA_dei_hq_2_chroma_cstat Register (offset = 30Ch) [reset = 0h]

VPDMA_dei_hq_2_chroma_cstat is shown in Figure 12-391 and described in Table 12-304.

**Figure 12-391. VPDMA_dei_hq_2_chroma_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | LINE_MODE | |
| R-0h | R-0h | R/W-0h | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-304. VPDMA_dei_hq_2_chroma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-8 | LINE_MODE | R/W | 0h | Selects the output mode of the line buffer.<br>0: repeat lines twice each output data line gets 2 times the number of frame lines.<br>1: each line once with Line Buffer Disabled.. so no mirroring. Each line gets frame lines with identical data.<br>2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines.<br>3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines. |
| 7-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.97 VPDMA_dei_hq_3_luma_cstat Register (offset = 310h) [reset = 0h]

VPDMA_dei_hq_3_luma_cstat is shown in Figure 12-392 and described in Table 12-305.

**Figure 12-392. VPDMA_dei_hq_3_luma_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-305. VPDMA_dei_hq_3_luma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.98 VPDMA_dei_hq_3_chroma_cstat Register (offset = 314h) [reset = 0h]

VPDMA_dei_hq_3_chroma_cstat is shown in Figure 12-393 and described in Table 12-306.

**Figure 12-393. VPDMA_dei_hq_3_chroma_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | LINE_MODE | |
| R-0h | R-0h | R/W-0h | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-306. VPDMA_dei_hq_3_chroma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-8 | LINE_MODE | R/W | 0h | Selects the output mode of the line buffer.<br>0: repeat lines twice each output data line gets 2 times the number of frame lines.<br>1: each line once with Line Buffer Disabled.. so no mirroring. Each line gets frame lines with identical data.<br>2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines.<br>3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines. |
| 7-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.99 VPDMA_dei_hq_mv_in_cstat Register (offset = 330h) [reset = 0h]

VPDMA_dei_hq_mv_in_cstat is shown in Figure 12-394 and described in Table 12-307.

**Figure 12-394. VPDMA_dei_hq_mv_in_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-307. VPDMA_dei_hq_mv_in_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.100 VPDMA_dei_hq_mv_out_cstat Register (offset = 33Ch) [reset = 0h]

VPDMA_dei_hq_mv_out_cstat is shown in Figure 12-395 and described in Table 12-308.

**Figure 12-395. VPDMA_dei_hq_mv_out_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-308. VPDMA_dei_hq_mv_out_cstat Register Field Descriptions**

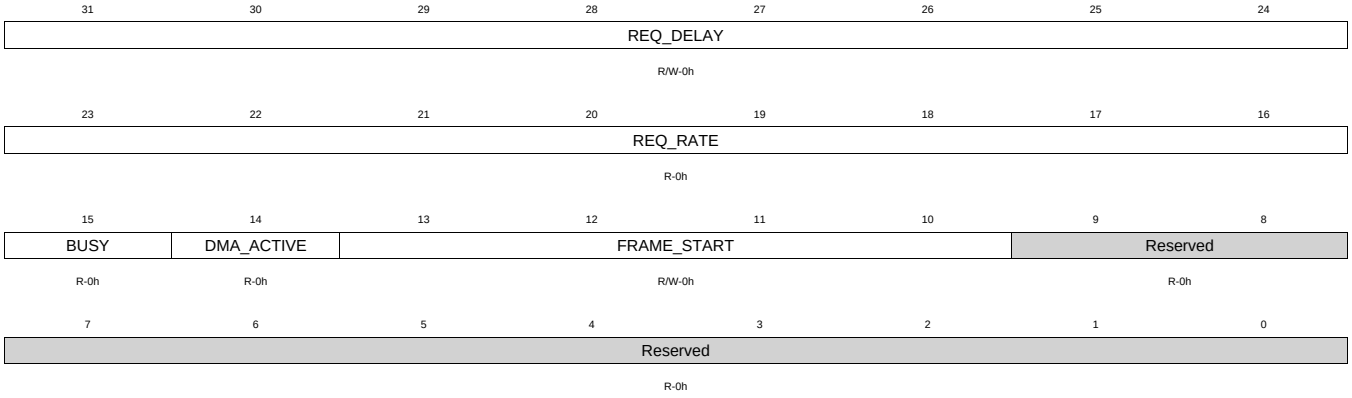
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.101 VPDMA_dei_sc_out_cstat Register (offset = 344h) [reset = 0h]

VPDMA_dei_sc_out_cstat is shown in Figure 12-396 and described in Table 12-309.

**Figure 12-396. VPDMA_dei_sc_out_cstat Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-309. VPDMA_dei_sc_out_cstat Register Field Descriptions**

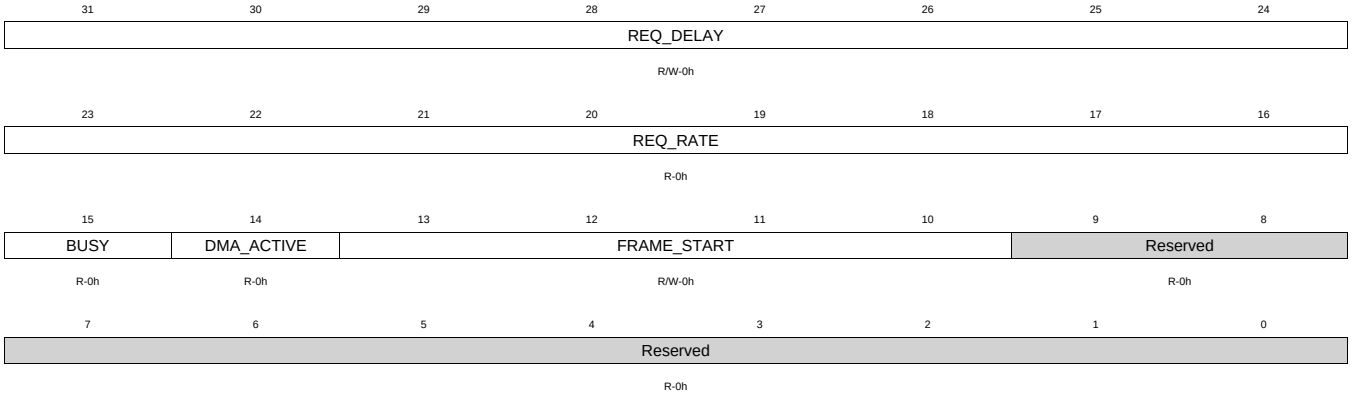
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.102 VPDMA_pip_wrbk_cstat Register (offset = 348h) [reset = 0h]

VPDMA_pip_wrbk_cstat is shown in Figure 12-397 and described in Table 12-310.

**Figure 12-397. VPDMA_pip_wrbk_cstat Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-310. VPDMA_pip_wrbk_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.103 VPDMA_sc_in_chroma_cstat Register (offset = 34Ch) [reset = 0h]

VPDMA_sc_in_chroma_cstat is shown in Figure 12-398 and described in Table 12-311.

**Figure 12-398. VPDMA_sc_in_chroma_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | LINE_MODE | |
| R-0h | R-0h | R/W-0h | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-311. VPDMA_sc_in_chroma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-8 | LINE_MODE | R/W | 0h | Selects the output mode of the line buffer.<br>0 = repeat lines twice each output data line gets 2 times the number of frame lines.<br>1 = each line once with Line Buffer Disabled, so no mirroring. Each line gets frame lines with identical data.<br>2 = Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines.<br>3 = each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines. |
| 7-0 | Reserved | R | 0h | |

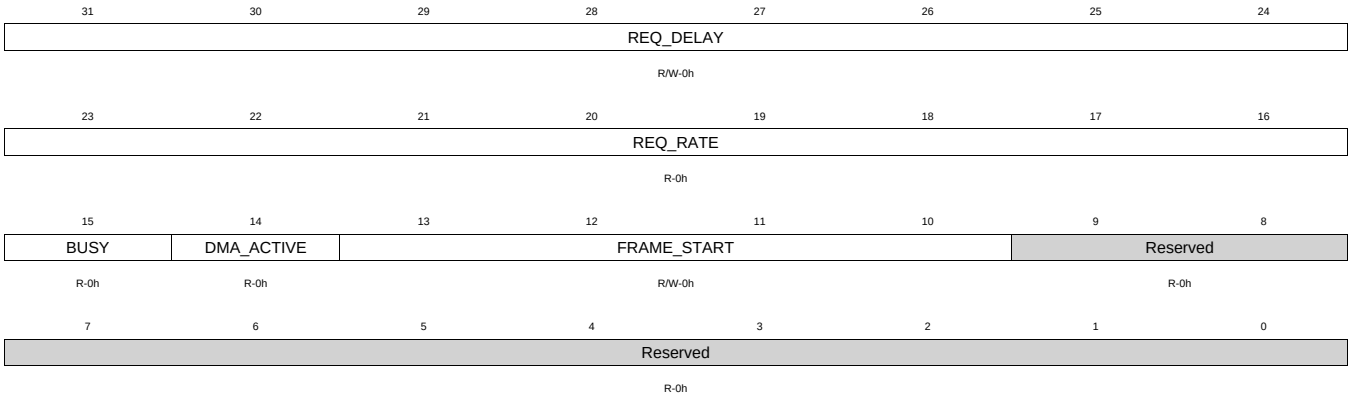


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.104 VPDMA_sc_in_luma_cstat Register (offset = 350h) [reset = 0h]

VPDMA_sc_in_luma_cstat is shown in Figure 12-399 and described in Table 12-312.

**Figure 12-399. VPDMA_sc_in_luma_cstat Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-312. VPDMA_sc_in_luma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.105 VPDMA_sc_out_cstat Register (offset = 374h) [reset = 0h]

VPDMA_sc_out_cstat is shown in Figure 12-400 and described in Table 12-313.

**Figure 12-400. VPDMA_sc_out_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-313. VPDMA_sc_out_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |

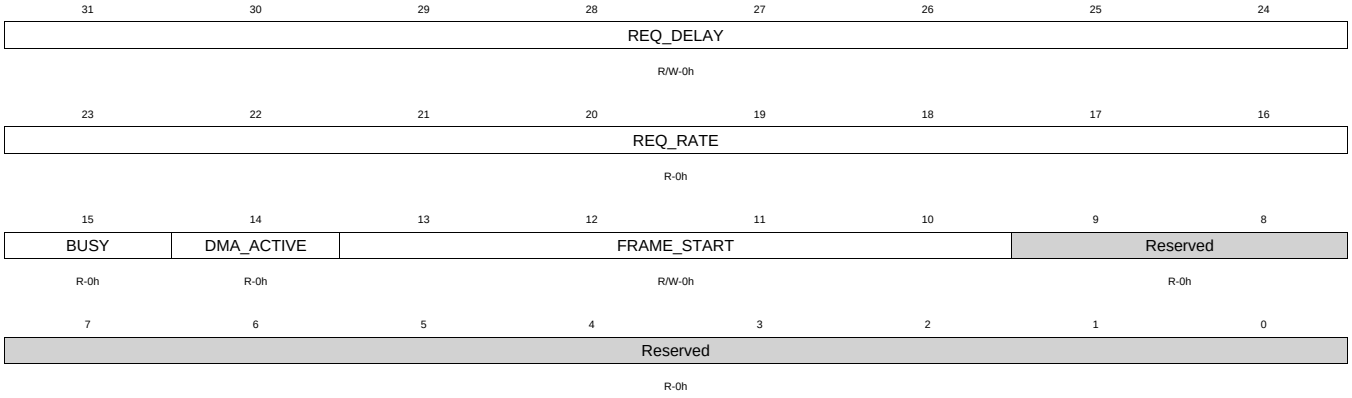


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.106 VPDMA_comp_wrbk_cstat Register (offset = 378h) [reset = 0h]

VPDMA_comp_wrbk_cstat is shown in Figure 12-401 and described in Table 12-314.

**Figure 12-401. VPDMA_comp_wrbk_cstat Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-314. VPDMA_comp_wrbk_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.107 VPDMA_grpx1_data_cstat Register (offset = 37Ch) [reset = 0h]

VPDMA_grpx1_data_cstat is shown in Figure 12-402 and described in Table 12-315.

**Figure 12-402. VPDMA_grpx1_data_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-315. VPDMA_grpx1_data_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.108 VPDMA_grpx2_data_cstat Register (offset = 380h) [reset = 0h]

VPDMA_grpx2_data_cstat is shown in Figure 12-403 and described in Table 12-316.

**Figure 12-403. VPDMA_grpx2_data_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-316. VPDMA_grpx2_data_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.109 VPDMA_grpx3_data_cstat Register (offset = 384h) [reset = 0h]

VPDMA_grpx3_data_cstat is shown in Figure 12-404 and described in Table 12-317.

**Figure 12-404. VPDMA_grpx3_data_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-317. VPDMA_grpx3_data_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.110 VPDMA_vip1_lo_y_cstat Register (offset = 388h) [reset = 0h]

VPDMA_vip1_lo_y_cstat is shown in Figure 12-405 and described in Table 12-318.

**Figure 12-405. VPDMA_vip1_lo_y_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-318. VPDMA_vip1_lo_y_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.111 VPDMA_vip1_lo_uv_cstat Register (offset = 38Ch) [reset = 0h]

VPDMA_vip1_lo_uv_cstat is shown in Figure 12-406 and described in Table 12-319.

**Figure 12-406. VPDMA_vip1_lo_uv_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-319. VPDMA_vip1_lo_uv_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.112 VPDMA_vip1_up_y_cstat Register (offset = 390h) [reset = 0h]

VPDMA_vip1_up_y_cstat is shown in Figure 12-407 and described in Table 12-320.

**Figure 12-407. VPDMA_vip1_up_y_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-320. VPDMA_vip1_up_y_cstat Register Field Descriptions**

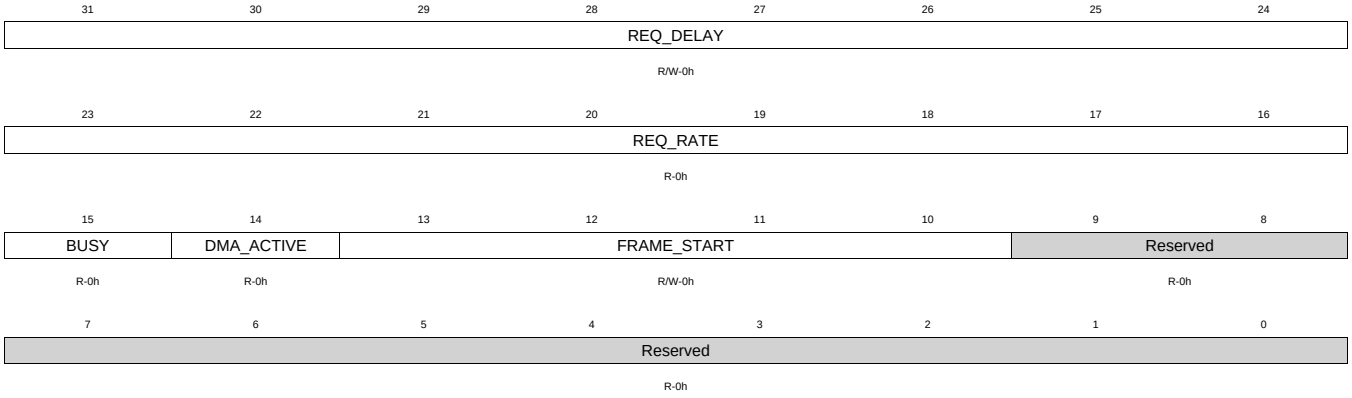
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.113 VPDMA_vip1_up_uv_cstat Register (offset = 394h) [reset = 0h]

VPDMA_vip1_up_uv_cstat is shown in Figure 12-408 and described in Table 12-321.

**Figure 12-408. VPDMA_vip1_up_uv_cstat Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-321. VPDMA_vip1_up_uv_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.114 VPDMA_vip2_lo_y_cstat Register (offset = 398h) [reset = 0h]

VPDMA_vip2_lo_y_cstat is shown in Figure 12-409 and described in Table 12-322.

**Figure 12-409. VPDMA_vip2_lo_y_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-322. VPDMA_vip2_lo_y_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.115 VPDMA_vip2_lo_uv_cstat Register (offset = 39Ch) [reset = 0h]

VPDMA_vip2_lo_uv_cstat is shown in Figure 12-410 and described in Table 12-323.

**Figure 12-410. VPDMA_vip2_lo_uv_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-323. VPDMA_vip2_lo_uv_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.116 VPDMA_vip2_up_y_cstat Register (offset = 3A0h) [reset = 0h]

VPDMA_vip2_up_y_cstat is shown in Figure 12-411 and described in Table 12-324.

**Figure 12-411. VPDMA_vip2_up_y_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-324. VPDMA_vip2_up_y_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.117 VPDMA_vip2_up_uv_cstat Register (offset = 3A4h) [reset = 0h]

VPDMA_vip2_up_uv_cstat is shown in Figure 12-412 and described in Table 12-325.

**Figure 12-412. VPDMA_vip2_up_uv_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-325. VPDMA_vip2_up_uv_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.118 VPDMA_grpx1_st_cstat Register (offset = 3A8h) [reset = 0h]

VPDMA_grpx1_st_cstat is shown in Figure 12-413 and described in Table 12-326.

**Figure 12-413. VPDMA_grpx1_st_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-326. VPDMA_grpx1_st_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.119 VPDMA_grpx2_st_cstat Register (offset = 3ACh) [reset = 0h]

VPDMA_grpx2_st_cstat is shown in Figure 12-414 and described in Table 12-327.

**Figure 12-414. VPDMA_grpx2_st_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-327. VPDMA_grpx2_st_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.120 VPDMA_grpx3_st_cstat Register (offset = 3B0h) [reset = 0h]

VPDMA_grpx3_st_cstat is shown in Figure 12-415 and described in Table 12-328.

**Figure 12-415. VPDMA_grpx3_st_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-328. VPDMA_grpx3_st_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.121 VPDMA_nf_422_in_cstat Register (offset = 3B4h) [reset = 0h]

VPDMA_nf_422_in_cstat is shown in Figure 12-416 and described in Table 12-329.

**Figure 12-416. VPDMA_nf_422_in_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-329. VPDMA_nf_422_in_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.122 VPDMA_nf_420_y_in_cstat Register (offset = 3B8h) [reset = 0h]

VPDMA_nf_420_y_in_cstat is shown in Figure 12-417 and described in Table 12-330.

**Figure 12-417. VPDMA_nf_420_y_in_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-330. VPDMA_nf_420_y_in_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.123 VPDMA_nf_420_uv_in_cstat Register (offset = 3BCh) [reset = 0h]

VPDMA_nf_420_uv_in_cstat is shown in Figure 12-418 and described in Table 12-331.

**Figure 12-418. VPDMA_nf_420_uv_in_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-331. VPDMA_nf_420_uv_in_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.124 VPDMA_nf_420_y_out_cstat Register (offset = 3C0h) [reset = 0h]

VPDMA_nf_420_y_out_cstat is shown in Figure 12-419 and described in Table 12-332.

**Figure 12-419. VPDMA_nf_420_y_out_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-332. VPDMA_nf_420_y_out_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.125 VPDMA_nf_420_uv_out_cstat Register (offset = 3C4h) [reset = 0h]

VPDMA_nf_420_uv_out_cstat is shown in Figure 12-420 and described in Table 12-333.

**Figure 12-420. VPDMA_nf_420_uv_out_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-333. VPDMA_nf_420_uv_out_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.126 VPDMA_vbi_sdvenc_cstat Register (offset = 3CCh) [reset = 0h]

VPDMA_vbi_sdvenc_cstat is shown in Figure 12-421 and described in Table 12-334.

**Figure 12-421. VPDMA_vbi_sdvenc_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-334. VPDMA_vbi_sdvenc_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.127 VPDMA_vpi_ctl_cstat Register (offset = 3D0h) [reset = 0h]

VPDMA_vpi_ctl_cstat is shown in Figure 12-422 and described in Table 12-335.

**Figure 12-422. VPDMA_vpi_ctl_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-335. VPDMA_vpi_ctl_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.128 VPDMA_hdmi_wrbk_out_cstat Register (offset = 3D4h) [reset = 0h]

VPDMA_hdmi_wrbk_out_cstat is shown in Figure 12-423 and described in Table 12-336.

**Figure 12-423. VPDMA_hdmi_wrbk_out_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-336. VPDMA_hdmi_wrbk_out_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.129 VPDMA_trans1_chroma_cstat Register (offset = 3D8h) [reset = 0h]

VPDMA_trans1_chroma_cstat is shown in Figure 12-424 and described in Table 12-337.

**Figure 12-424. VPDMA_trans1_chroma_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | LINE_MODE | |
| R-0h | R-0h | R/W-0h | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-337. VPDMA_trans1_chroma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-8 | LINE_MODE | R/W | 0h | Selects the output mode of the line buffer.<br>0: repeat lines twice each output data line gets 2 times the number of frame lines.<br>1: each line once with Line Buffer Disabled.. so no mirroring. Each line gets frame lines with identical data.<br>2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines.<br>3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines. |
| 7-0 | Reserved | R | 0h | |

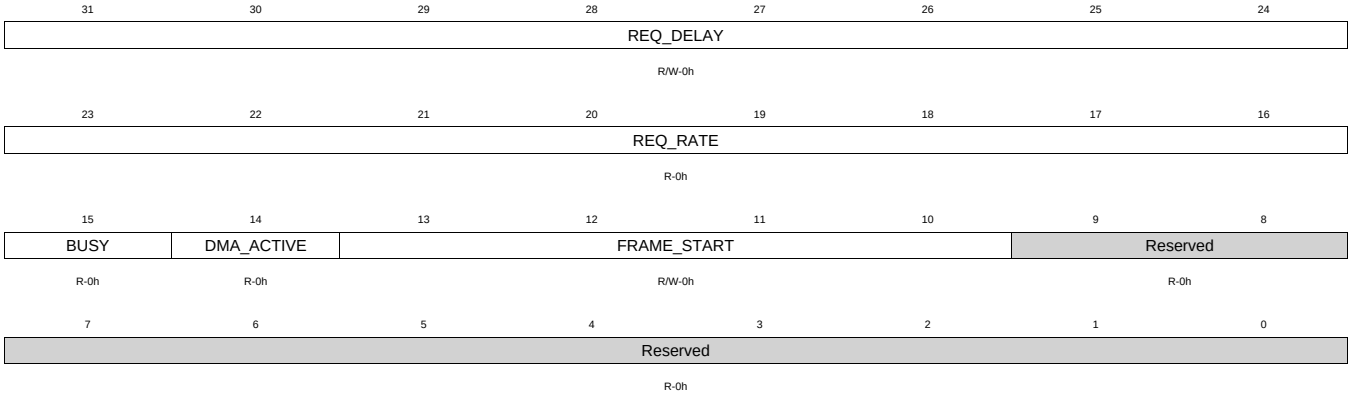


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.130 VPDMA_trans1_luma_cstat Register (offset = 3DCh) [reset = 0h]

VPDMA_trans1_luma_cstat is shown in Figure 12-425 and described in Table 12-338.

**Figure 12-425. VPDMA_trans1_luma_cstat Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-338. VPDMA_trans1_luma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.131 VPDMA_trans2_chroma_cstat Register (offset = 3E0h) [reset = 0h]

VPDMA_trans2_chroma_cstat is shown in Figure 12-426 and described in Table 12-339.

**Figure 12-426. VPDMA_trans2_chroma_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | LINE_MODE | |
| R-0h | R-0h | R/W-0h | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-339. VPDMA_trans2_chroma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-8 | LINE_MODE | R/W | 0h | Selects the output mode of the line buffer.<br>0: repeat lines twice each output data line gets 2 times the number of frame lines.<br>1: each line once with Line Buffer Disabled.. so no mirroring. Each line gets frame lines with identical data.<br>2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines.<br>3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines. |
| 7-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.132 VPDMA_trans2_luma_cstat Register (offset = 3E4h) [reset = 0h]

VPDMA_trans2_luma_cstat is shown in Figure 12-427 and described in Table 12-340.

**Figure 12-427. VPDMA_trans2_luma_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-340. VPDMA_trans2_luma_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.133 VPDMA_vip1_anc_a_cstat Register (offset = 3E8h) [reset = 0h]

VPDMA_vip1_anc_a_cstat is shown in Figure 12-428 and described in Table 12-341.

**Figure 12-428. VPDMA_vip1_anc_a_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-341. VPDMA_vip1_anc_a_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.134 VPDMA_vip1_anc_b_cstat Register (offset = 3ECh) [reset = 0h]

VPDMA_vip1_anc_b_cstat is shown in Figure 12-429 and described in Table 12-342.

**Figure 12-429. VPDMA_vip1_anc_b_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-342. VPDMA_vip1_anc_b_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.135 VPDMA_vip2_anc_a_cstat Register (offset = 3F0h) [reset = 0h]

VPDMA_vip2_anc_a_cstat is shown in Figure 12-430 and described in Table 12-343.

**Figure 12-430. VPDMA_vip2_anc_a_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-343. VPDMA_vip2_anc_a_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.8.136 VPDMA_vip2_anc_b_cstat Register (offset = 3F4h) [reset = 0h]

VPDMA_vip2_anc_b_cstat is shown in Figure 12-431 and described in Table 12-344.

**Figure 12-431. VPDMA_vip2_anc_b_cstat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQ_DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQ_RATE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BUSY | DMA_ACTIVE | FRAME_START | | | | Reserved | |
| R-0h | R-0h | R/W-0h | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-344. VPDMA_vip2_anc_b_cstat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | REQ_DELAY | R/W | 0h | The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible. |
| 23-16 | REQ_RATE | R | 0h | The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins. |
| 15 | BUSY | R | 0h | Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory. |
| 14 | DMA_ACTIVE | R | 0h | Signals if the client is currently actively sending DMA requests |
| 13-10 | FRAME_START | R/W | 0h | The source of the start frame event for the client.<br>0 = Change in hdmi_field_id<br>1 = Change in dvo2_field_id<br>2 = RESERVED<br>3 = Change in value of sd_field_id<br>4 = Change in value of List Manager Internal Field - 0<br>5 = Change in value of List Manager Internal Field - 1<br>6 = Change in value of List Manager Internal Field - 2<br>7 = Start whenever channel is free |
| 9-0 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9 HD_VENC_D Registers

Table 12-345 lists the memory-mapped registers for the HD_VENC_D. All register offset addresses not listed in Table 12-345 should be considered as reserved locations and the register contents should not be modified.

**Table 12-345. HD_VENC_D REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | HD_VENC_D_cfg0 | VENC Mode Register | Section 12.3.9.1 |
| 4h | HD_VENC_D_cfg1 | Color Space Converter Coefficient Register | Section 12.3.9.2 |
| 8h | HD_VENC_D_cfg2 | Color Space Converter Coefficient Register | Section 12.3.9.3 |
| Ch | HD_VENC_D_cfg3 | Color Space Converter Coefficient Register | Section 12.3.9.4 |
| 10h | HD_VENC_D_cfg4 | Color Space Converter Coefficient Register | Section 12.3.9.5 |
| 14h | HD_VENC_D_cfg5 | Color Space Converter Coefficient Register | Section 12.3.9.6 |
| 18h | HD_VENC_D_cfg6 | Color Space Converter Coefficient Register | Section 12.3.9.7 |
| 1Ch | HD_VENC_D_cfg7 | Reserved Register | Section 12.3.9.8 |
| 20h | HD_VENC_D_cfg8 | Reserved Register | Section 12.3.9.9 |
| 24h | HD_VENC_D_cfg9 | Reserved Register | Section 12.3.9.10 |
| 28h | HD_VENC_D_cfg10 | Frame Size Register | Section 12.3.9.11 |
| 2Ch | HD_VENC_D_cfg11 | Reserved Register | Section 12.3.9.12 |
| 30h | HD_VENC_D_cfg12 | Active Pixels per Line Register | Section 12.3.9.13 |
| 34h | HD_VENC_D_cfg13 | VENC Control Register | Section 12.3.9.14 |
| 38h | HD_VENC_D_cfg14 | Reserved Register | Section 12.3.9.15 |
| 3Ch | HD_VENC_D_cfg15 | DVO Control Register | Section 12.3.9.16 |
| 40h | HD_VENC_D_cfg16 | DVO Control Register | Section 12.3.9.17 |
| 44h | HD_VENC_D_cfg17 | DVO Control Register | Section 12.3.9.18 |
| 48h | HD_VENC_D_cfg18 | DVO Control Register | Section 12.3.9.19 |
| 4Ch | HD_VENC_D_cfg19 | DVO Control Register | Section 12.3.9.20 |
| 50h | HD_VENC_D_cfg20 | DVO/Compositor IF Control Register | Section 12.3.9.21 |
| 54h | HD_VENC_D_cfg21 | Compositor IF Control Register | Section 12.3.9.22 |
| 58h | HD_VENC_D_cfg22 | Compositor IF Control Register | Section 12.3.9.23 |
| 5Ch | HD_VENC_D_cfg23 | Compositor IF Control Register | Section 12.3.9.24 |
| 60h | HD_VENC_D_cfg24 | Compositor IF Control Register | Section 12.3.9.25 |
| 64h | HD_VENC_D_cfg25 | Compositor IF Control Register | Section 12.3.9.26 |
| 1000h-1FFFh | HD_VENC_D_GAMMA_LUT | Gamma Lookup Table | Section 12.3.9.27 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.1 HD_VENC_D_cfg0 Register (offset = 0h) [reset = 0h]

HD_VENC_D_cfg0 is shown in Figure 12-432 and described in Table 12-346.

VENC Mode Register

**Figure 12-432. HD_VENC_D_cfg0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | START | JED | Reserved | DVO_OFF | S_422 | I_DVO_A | I_DVO_H |
| R-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| I_DVO_V | I_DVO_F | IVT_FID | Reserved | | DVO_FMT | | |
| R/W-0h | R/W-0h | R/W-0h | R-0h | | R/W-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| STEST | Reserved | BYPS_GC | Reserved | | | | |
| R/W-0h | R-0h | R/W-0h | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | BYPS_CS | Reserved | I_PN | Reserved | | |
| R-0h | | R/W-0h | R-0h | R/W-0h | R-0h | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-346. HD_VENC_D_cfg0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Reserved | R | 0h | Reserved |
| 30 | START | R/W | 0h | This bit will start the operation of encoder.<br>0: encoder is in standby mode.. no data and signals will be outputted<br>1: encoder is in the normal operation mode.<br>During the normal operation, this bit should be kept at 1. Normally software will set the bit whenever the operation needs to be started. |
| 29 | JED | R/W | 0h | This bit can enable the JEIDA output format<br>0: DVO will output data in normal format<br>1: DVO will output data in JEIDA format<br>The JEIDA format is used to interface certain LCD panels. It is used in RGB discrete sync mode only. |
| 28 | Reserved | R | 0h | Reserved |
| 27 | DVO_OFF | R/W | 0h | This bit forces the DVO to 0s.<br>0: for normal operation<br>1: to force the DVO to 0s |
| 26 | S_422 | R/W | 0h | When the DVO output is in 4:2:2 YCbCr format, this bit controls the CbCr format.<br>0: The CBCR data is generated by a 444-to422- decimation filter.<br>1: The CBCR data are captured by every-other pixel. (skip one pixel at a time) |
| 25 | I_DVO_A | R/W | 0h | This bit controls the polarity of DVO_ACTVID signal.<br>0: Active High<br>1: Active Low |
| 24 | I_DVO_H | R/W | 0h | This bit controls the polarity of DVO_HS signal.<br>0: Active High<br>1: Active Low |
| 23 | I_DVO_V | R/W | 0h | This bit controls the polarity of DVO_VS signal.<br>0: Active High<br>1: Active Low |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-346. HD_VENC_D_cfg0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 22 | I_DVO_F | R/W | 0h | This bit controls the polarity of DVO_FID signal.<br>0: Not invert<br>1: The DVO_FID signal will be inverted |
| 21 | IVT_FID | R/W | 0h | This bit controls the polarity of DTV_FID signal.<br>0: Not invert<br>1: The DTV_FID signal will be inverted |
| 20-19 | Reserved | R | 0h | Reserved |
| 18-16 | DVO_FMT | R/W | 0h | These three bits define the format of digital video output port.<br>000: Single channel 10-bit CCIR656 video stream for SD outputs.<br>The clock rate of the strem is 2X pixel rate.<br>For 480i format, clk2x = 27 MHz.<br>For 480p format, clk2x = 54 MHz.<br>001: Two channel 10-bit CCIR656 video streams. The video data is in YCbCr 422 format. The data rate equals to the pixel rate.<br>010: Three channel 10-bit video streams with embedded sync SAV/EAV. The video format can by YCbCr 444 or RGB. And the data rate equals to the pixel rate.<br>011: Three channel 10-bit video streams with dedicated HS/VS/FID/ACTVID sync signals. The three channel component digital video can be RGB or YCbCr444 format. It can be used for all display formats. And the data rate equals to the pixel rate.<br>100: Two channel 10-bit video streams with dedicated HS/VS/FID/ACTVID sync signals. The video data format is YCbCr422. The first channel is Y. and CbCr are multiplexed on the second channel. It can be used for all display formats. And the data rate equals to the pixel rate. |
| 15 | STEST | R/W | 0h | Self test mode:<br>0: The HD encoder will be in normal operation mode<br>1: The HD encoder will enter self-test mode. The encoder will generate a 100% color bar internally. The video data from OSD will be ignored. |
| 14 | Reserved | R | 0h | Reserved |
| 13 | BYPS_GC | R/W | 0h | Bypass Gamma Correction block:<br>0 : Apply Gamma-correction to video data<br>1 : Bypassed Gamma-correction |
| 12-6 | Reserved | R | 0h | Reserved |
| 5 | BYPS_CS | R/W | 0h | Bypass color space converter. This means that no color space conversion is needed. If it is not bypassed, the proper coefficient should be set in the corresponding registers.<br>0 : Color space conversion is engaged<br>1 : Bypass |
| 4 | Reserved | R | 0h | Reserved |
| 3 | I_PN | R/W | 0h | Scan format:<br>0 : Progressive<br>1 : Interlace |
| 2-0 | Reserved | R | 0h | Reserved |

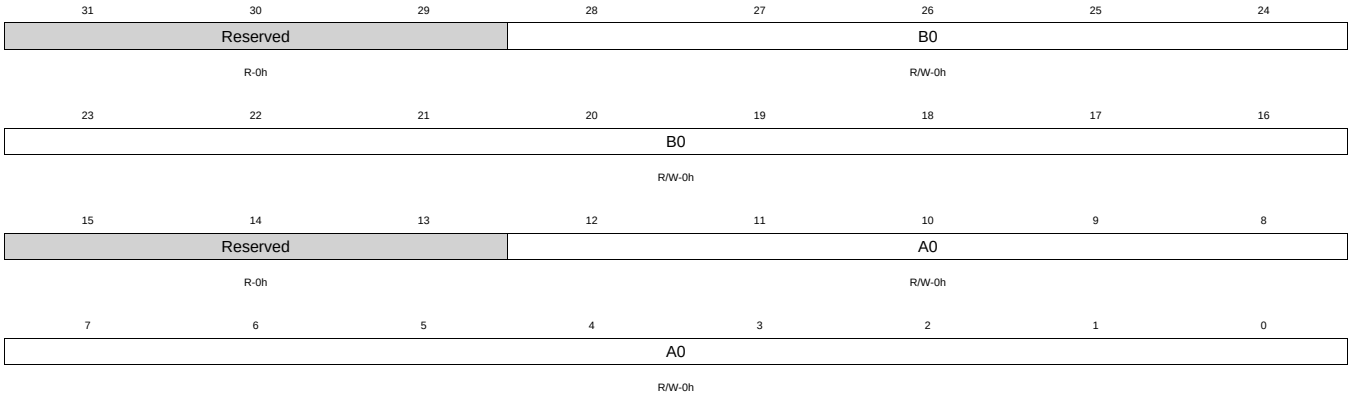

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.2 HD_VENC_D_cfg1 Register (offset = 4h) [reset = 0h]

HD_VENC_D_cfg1 is shown in Figure 12-433 and described in Table 12-347.

Color Space Converter Coefficient Register

**Figure 12-433. HD_VENC_D_cfg1 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-347. HD_VENC_D_cfg1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-16 | B0 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in HD_VENC_D_cfg1) |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-0 | A0 | R/W | 0h | Its is represented as Q3.10 number. So the value ranges from -4 to +4. To convert a decimal number, multiply the number by 1024 and write it in the register in hex format. For example, to program 0.673, 0x2B1 should be written in the register.<br>(round)(0.673 X 1024) = (round)689.152 = 689 = 0x2B1.<br>If the real number is negative, then multiply it by 1024, and convert it to 2's compliment format in 12-bit. For example, if a coefficient is -1.893, 0x186E needs to be written in the register.<br>(round)(-1.893*1024)= -1938 = 0x186E (2'S compliment format of -1938 in 13-bit width) |

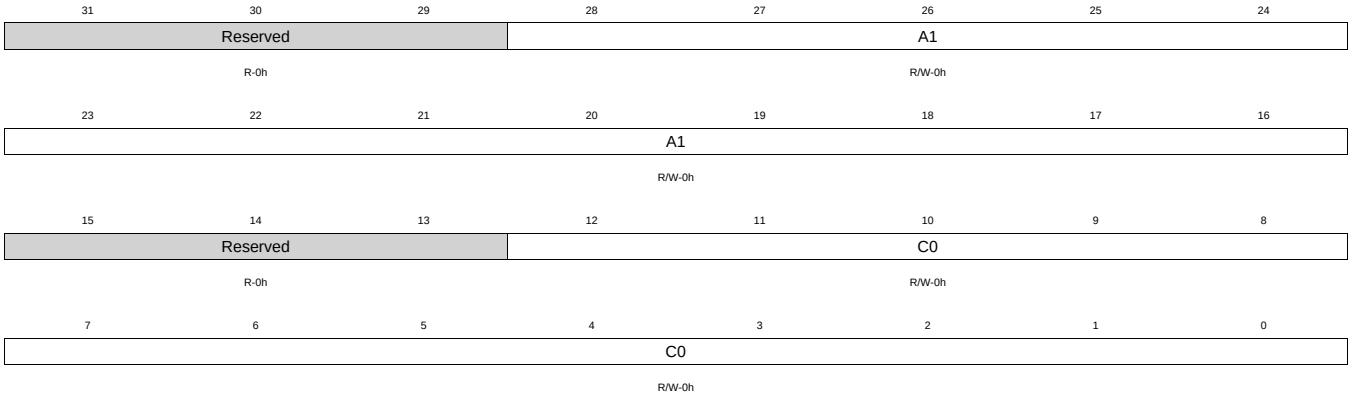

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.3 HD_VENC_D_cfg2 Register (offset = 8h) [reset = 0h]

HD_VENC_D_cfg2 is shown in Figure 12-434 and described in Table 12-348.

Color Space Converter Coefficient Register

**Figure 12-434. HD_VENC_D_cfg2 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-348. HD_VENC_D_cfg2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-16 | A1 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in HD_VENC_D_cfg1) |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-0 | C0 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in HD_VENC_D_cfg1) |

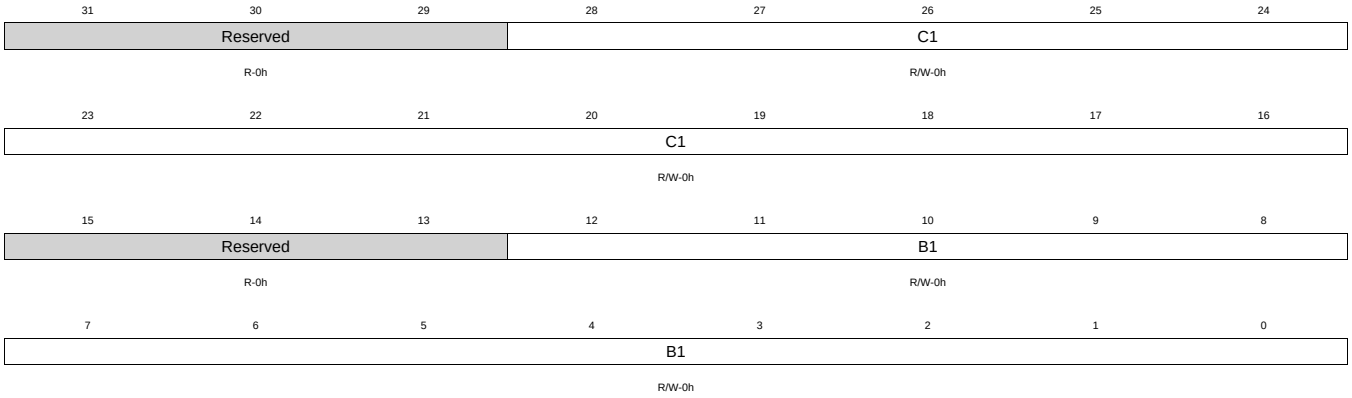

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.4 HD_VENC_D_cfg3 Register (offset = Ch) [reset = 0h]

HD_VENC_D_cfg3 is shown in Figure 12-435 and described in Table 12-349.

Color Space Converter Coefficient Register

**Figure 12-435. HD_VENC_D_cfg3 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-349. HD_VENC_D_cfg3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-16 | C1 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in HD_VENC_D_cfg1) |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-0 | B1 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in HD_VENC_D_cfg1) |

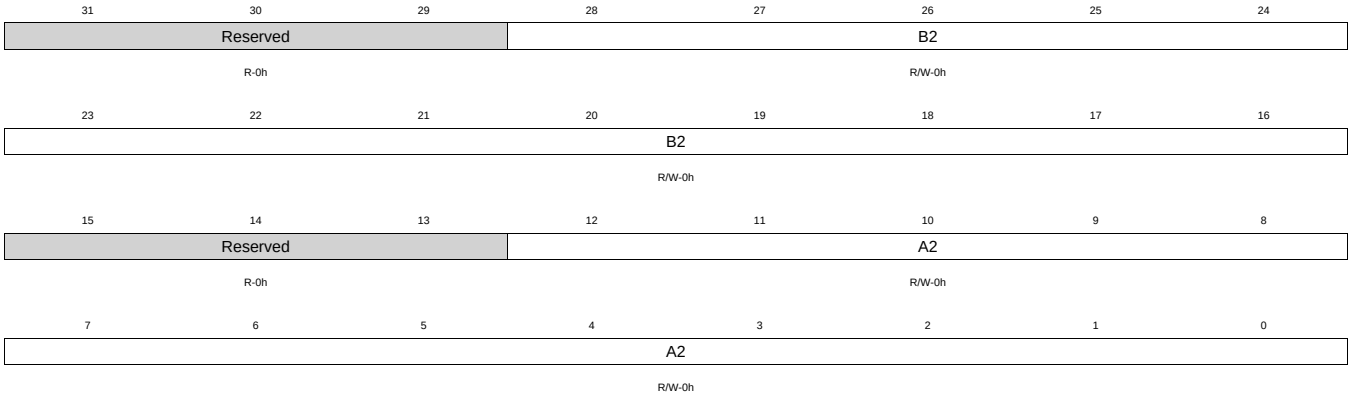

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.5 HD_VENC_D_cfg4 Register (offset = 10h) [reset = 0h]

HD_VENC_D_cfg4 is shown in Figure 12-436 and described in Table 12-350.

Color Space Converter Coefficient Register

**Figure 12-436. HD_VENC_D_cfg4 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-350. HD_VENC_D_cfg4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | Reserved |
| 28-16 | B2 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in HD_VENC_D_cfg1) |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-0 | A2 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in HD_VENC_D_cfg1) |

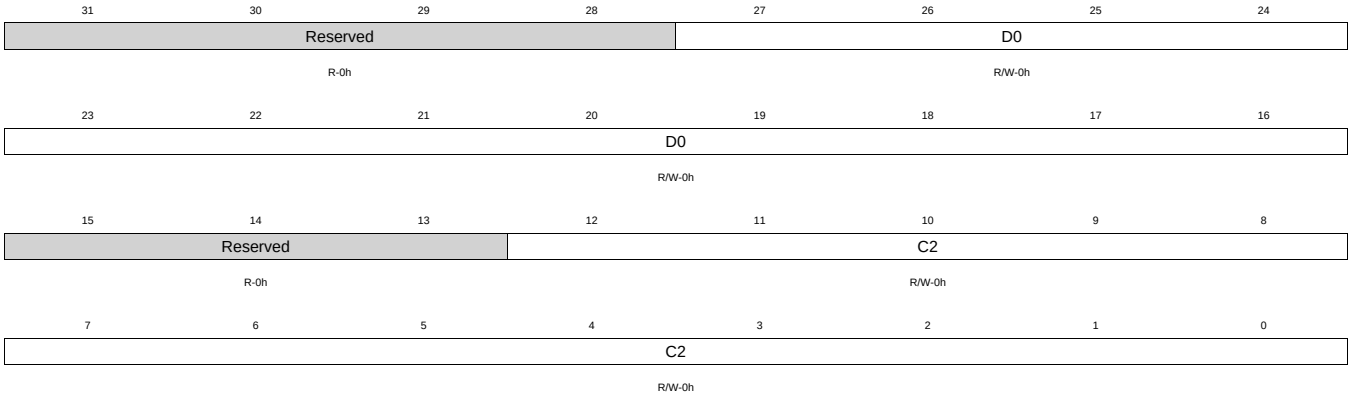

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.6 HD_VENC_D_cfg5 Register (offset = 14h) [reset = 0h]

HD_VENC_D_cfg5 is shown in Figure 12-437 and described in Table 12-351.

Color Space Converter Coefficient Register

**Figure 12-437. HD_VENC_D_cfg5 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-351. HD_VENC_D_cfg5 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | Reserved |
| 27-16 | D0 | R/W | 0h | Coefficients of color space converter. This coefficient is an integer number in the range of -2048 to 2047. It is in 12-bit wide 2's compliment format. The MSB is sign bit. For example, if this coefficient is 749, then 0x2ED (hex format) should be assigned to this register. Another example, if this coefficient is -1021, then 0xC03 should be assigned to this register. |
| 15-13 | Reserved | R | 0h | Reserved |
| 12-0 | C2 | R/W | 0h | Coefficients of color space converter. This coefficient is a real number in the range of -4 to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in HD_VENC_D_cfg1) |

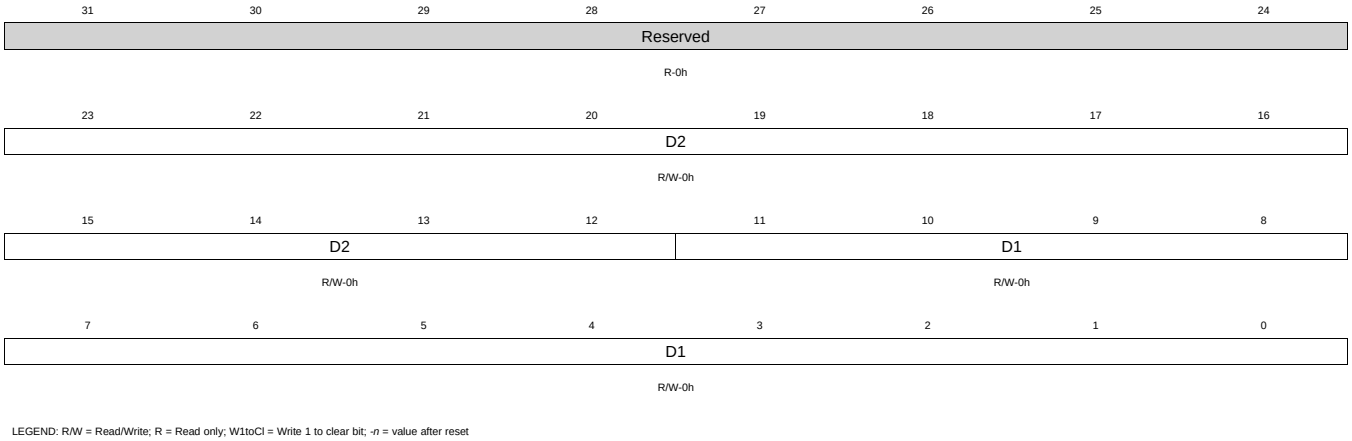

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.7 HD_VENC_D_cfg6 Register (offset = 18h) [reset = 0h]

HD_VENC_D_cfg6 is shown in Figure 12-438 and described in Table 12-352.

Color Space Converter Coefficient Register

**Figure 12-438. HD_VENC_D_cfg6 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-352. HD_VENC_D_cfg6 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-12 | D2 | R/W | 0h | Coefficients of color space converter. This coefficient is an integer number in the range of -2048 to 2047. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0 in HD_VENC_D_cfg5) |
| 11-0 | D1 | R/W | 0h | Coefficients of color space converter. This coefficient is an integer number in the range of -2048 to 2047. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0 in HD_VENC_D_cfg5) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.8 HD_VENC_D_cfg7 Register (offset = 1Ch) [reset = 0h]

HD_VENC_D_cfg7 is shown in Figure 12-439 and described in Table 12-353.

Reserved Register

**Figure 12-439. HD_VENC_D_cfg7 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 12-353. HD_VENC_D_cfg7 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.9 HD_VENC_D_cfg8 Register (offset = 20h) [reset = 0h]

HD_VENC_D_cfg8 is shown in Figure 12-440 and described in Table 12-354.

Reserved Register

**Figure 12-440. HD_VENC_D_cfg8 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 12-354. HD_VENC_D_cfg8 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.10 HD_VENC_D_cfg9 Register (offset = 24h) [reset = 0h]

HD_VENC_D_cfg9 is shown in Figure 12-441 and described in Table 12-355.

Reserved Register

**Figure 12-441. HD_VENC_D_cfg9 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 12-355. HD_VENC_D_cfg9 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | Reserved |

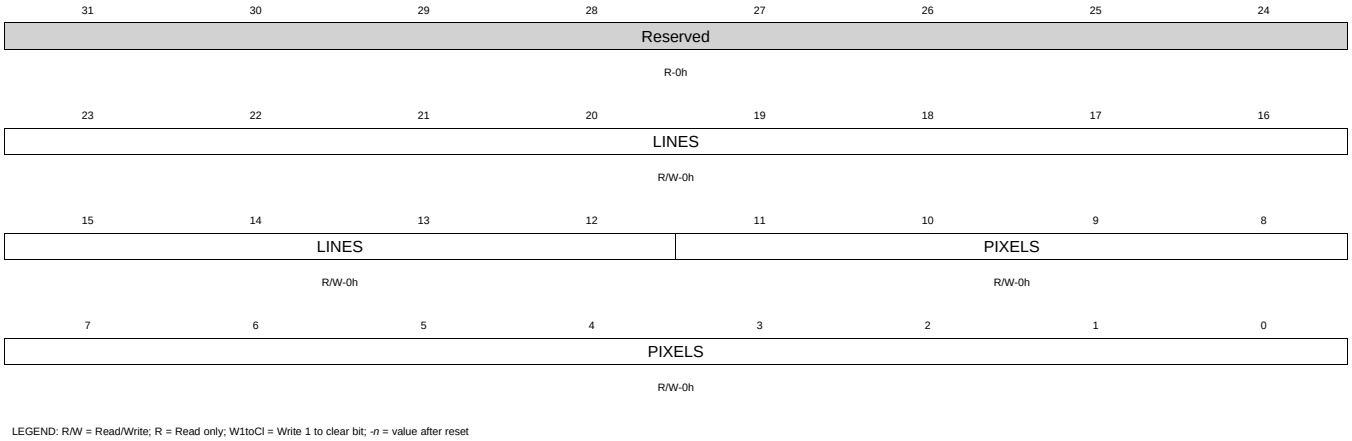

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.11 HD_VENC_D_cfg10 Register (offset = 28h) [reset = 0h]

HD_VENC_D_cfg10 is shown in Figure 12-442 and described in Table 12-356.

Frame Size Register

**Figure 12-442. HD_VENC_D_cfg10 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-356. HD_VENC_D_cfg10 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-12 | LINES | R/W | 0h | Defines the total number of lines per frame. |
| 11-0 | PIXELS | R/W | 0h | Defines the total number of pixels per line. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.12 HD_VENC_D_cfg11 Register (offset = 2Ch) [reset = 0h]

HD_VENC_D_cfg11 is shown in Figure 12-443 and described in Table 12-357.

Reserved Register

**Figure 12-443. HD_VENC_D_cfg11 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 12-357. HD_VENC_D_cfg11 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |

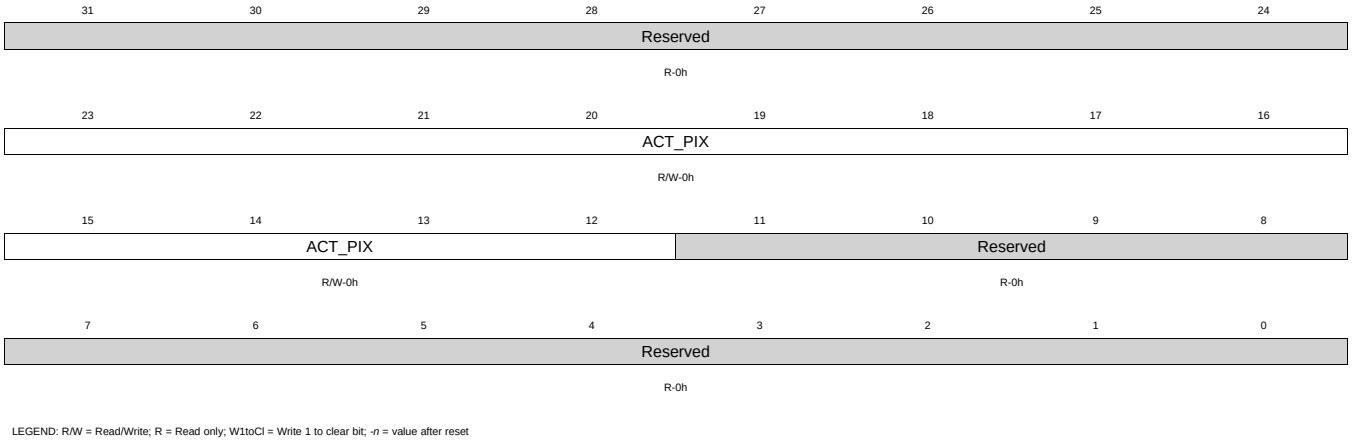

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.13 HD_VENC_D_cfg12 Register (offset = 30h) [reset = 0h]

HD_VENC_D_cfg12 is shown in Figure 12-444 and described in Table 12-358.

Active Pixels per Line Register

**Figure 12-444. HD_VENC_D_cfg12 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-358. HD_VENC_D_cfg12 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-12 | ACT_PIX | R/W | 0h | Defines the number of active pixels in one video line. |
| 11-0 | Reserved | R | 0h | Reserved |

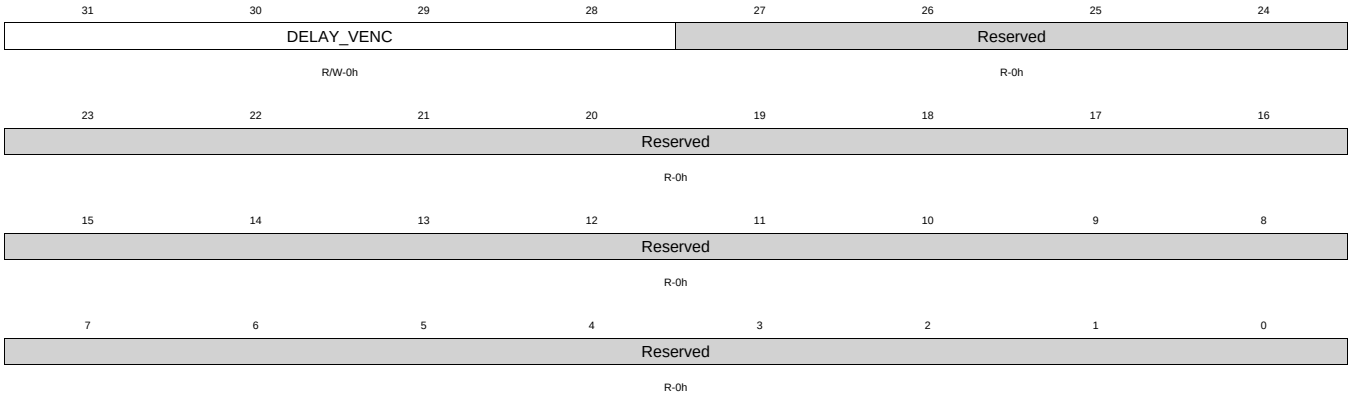

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.14 HD_VENC_D_cfg13 Register (offset = 34h) [reset = 0h]

HD_VENC_D_cfg13 is shown in Figure 12-445 and described in Table 12-359.

VENC Control Register

**Figure 12-445. HD_VENC_D_cfg13 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-359. HD_VENC_D_cfg13 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | DELAY_VENC | R/W | 0h | The high-4-bits of the delay counter of VENC_EN signal.. the low 8-bit of this counter is specified in CFG20 |
| 27-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.15 HD_VENC_D_cfg14 Register (offset = 38h) [reset = 0h]

HD_VENC_D_cfg14 is shown in Figure 12-446 and described in Table 12-360.

Reserved Register

**Figure 12-446. HD_VENC_D_cfg14 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 12-360. HD_VENC_D_cfg14 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.16 HD_VENC_D_cfg15 Register (offset = 3Ch) [reset = 0h]

HD_VENC_D_cfg15 is shown in Figure 12-447 and described in Table 12-361.

DVO Control Register

**Figure 12-447. HD_VENC_D_cfg15 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DVO_HS_WD | | | | | | | |
| R/W-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| DVO_AVD_HW | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DVO_AVD_HW | | | | DVO_AVST_H | | | |
| R/W-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DVO_AVST_H | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-361. HD_VENC_D_cfg15 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | DVO_HS_WD | R/W | 0h | Defines the width of DVO_HS pulse (in number of pixels). |
| 23-12 | DVO_AVD_HW | R/W | 0h | Defines the width of each active video line (in number of pixels). |
| 11-0 | DVO_AVST_H | R/W | 0h | Active Video Start Time<br>DVO_AVST_H = CFG10.PIXELS – CFG12.ACT_PIX<br>This parameter defines the SAV location in embedded sync mode and the location of the first active pixel on the DVO output in discrete sync mode. |

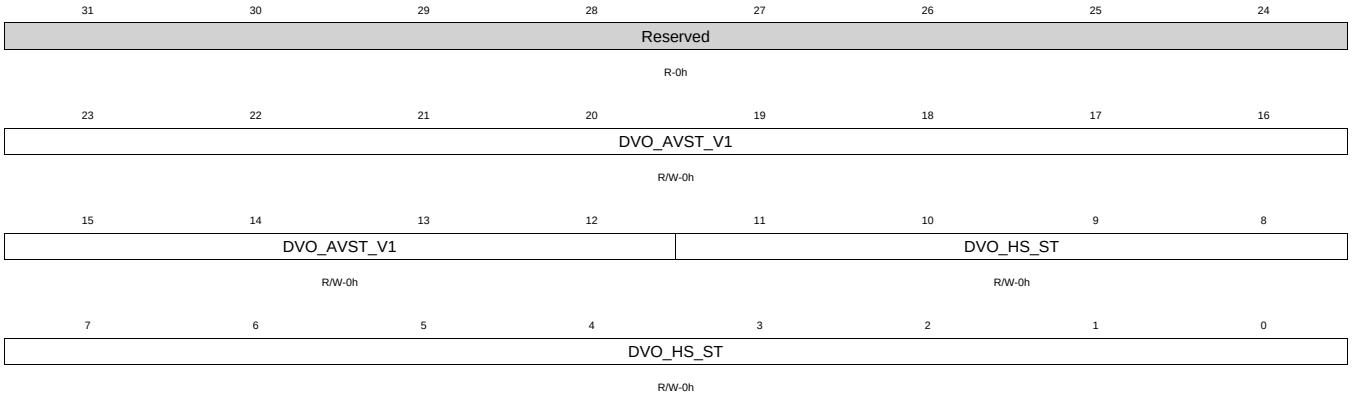

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.17 HD_VENC_D_cfg16 Register (offset = 40h) [reset = 0h]

HD_VENC_D_cfg16 is shown in Figure 12-448 and described in Table 12-362.

DVO Control Register

**Figure 12-448. HD_VENC_D_cfg16 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-362. HD_VENC_D_cfg16 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-12 | DVO_AVST_V1 | R/W | 0h | In progressive mode, it defines the first active line in a frame. In interlace mode, it defines the first active line of the first field in a frame. |
| 11-0 | DVO_HS_ST | R/W | 0h | Defines the starting location of the DVO_HS pulse on each line. |

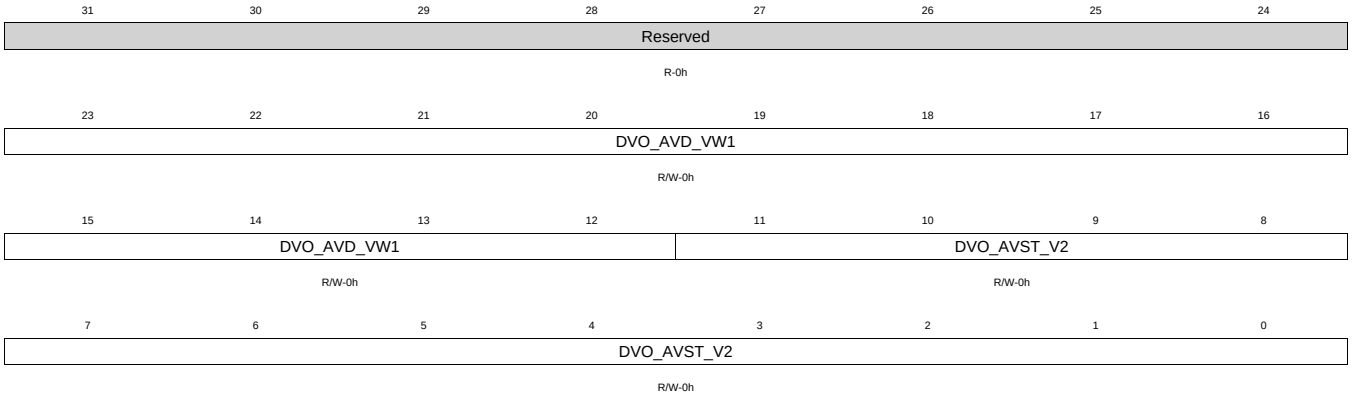

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.18 HD_VENC_D_cfg17 Register (offset = 44h) [reset = 0h]

HD_VENC_D_cfg17 is shown in Figure 12-449 and described in Table 12-363.

DVO Control Register

**Figure 12-449. HD_VENC_D_cfg17 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-363. HD_VENC_D_cfg17 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-12 | DVO_AVD_VW1 | R/W | 0h | In progressive mode, it defines the number of active video lines in a frame. In interlace mode, it defines the number of active lines in the first field. |
| 11-0 | DVO_AVST_V2 | R/W | 0h | Defines the first active line of second field in a frame. This parameter is only used in interlace mode. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.19 HD_VENC_D_cfg18 Register (offset = 48h) [reset = 0h]

HD_VENC_D_cfg18 is shown in Figure 12-450 and described in Table 12-364.

DVO Control Register

**Figure 12-450. HD_VENC_D_cfg18 Register**

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-364. HD_VENC_D_cfg18 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | DVO_VS_WD1 | R/W | 0h | In progressive mode. it defines the width of the DVO_VS pulse In interlace mode. it defines the width of the DVO_VS pulse of the first field. |
| 23-12 | DVO_VS_ST1 | R/W | 0h | In progressive mode, it defines the starting location of the DVO_VS pulse in a frame. In interlace mode, it defines the starting location of DVO_VS pulse of the first field. |
| 11-0 | DVO_AVD_VW2 | R/W | 0h | Defines the number of active lines in the second field. This parameter is only used in interlace mode. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.20 HD_VENC_D_cfg19 Register (offset = 4Ch) [reset = 0h]

HD_VENC_D_cfg19 is shown in Figure 12-451 and described in Table 12-365.

DVO Control Register

**Figure 12-451. HD_VENC_D_cfg19 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DVO_VS_WD2 | | | | | | | |
| R/W-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| DVO_FID_ST1 | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DVO_FID_ST1 | | | | DVO_VS_ST2 | | | |
| R/W-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DVO_VS_ST2 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-365. HD_VENC_D_cfg19 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | DVO_VS_WD2 | R/W | 0h | Defines the width of the DVO_VS pulse of the second field (in number of lines) |
| 23-12 | DVO_FID_ST1 | R/W | 0h | This parameter defines the top field line number at which the DVO_FID will switch (only applicable in the interlaced mode). |
| 11-0 | DVO_VS_ST2 | R/W | 0h | Defines the starting location of the DVO_VS of the second field. This parameter is only used in interlace mode. |

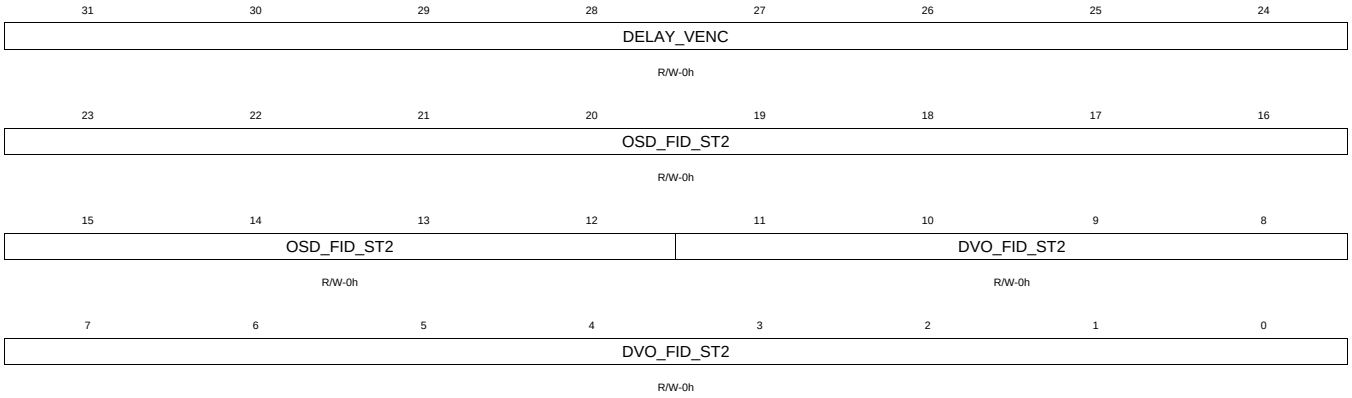

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.21 HD_VENC_D_cfg20 Register (offset = 50h) [reset = 0h]

HD_VENC_D_cfg20 is shown in Figure 12-452 and described in Table 12-366.

DVO/Compositor IF Control Register

**Figure 12-452. HD_VENC_D_cfg20 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-366. HD_VENC_D_cfg20 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | DELAY_VENC | R/W | 0h | These are the low 8-bits of the delay counter of VENC_EN signal from top. The high 4-bit is in CFG13. The maximum delay from the VENC_EN's activation is 4096 pixel clock periods. |
| 23-12 | OSD_FID_ST2 | R/W | 0h | This parameter defines the bottom field line number at which the DTV_FID will switch. This parameter is only used in interlace mode. OSD_FID_ST2 = int ( cfg010.lines/2 + OSD_FID_ST1 ) |
| 11-0 | DVO_FID_ST2 | R/W | 0h | Defines the starting location of the second field in interlace mode. This parameter is only used in interlace mode. |

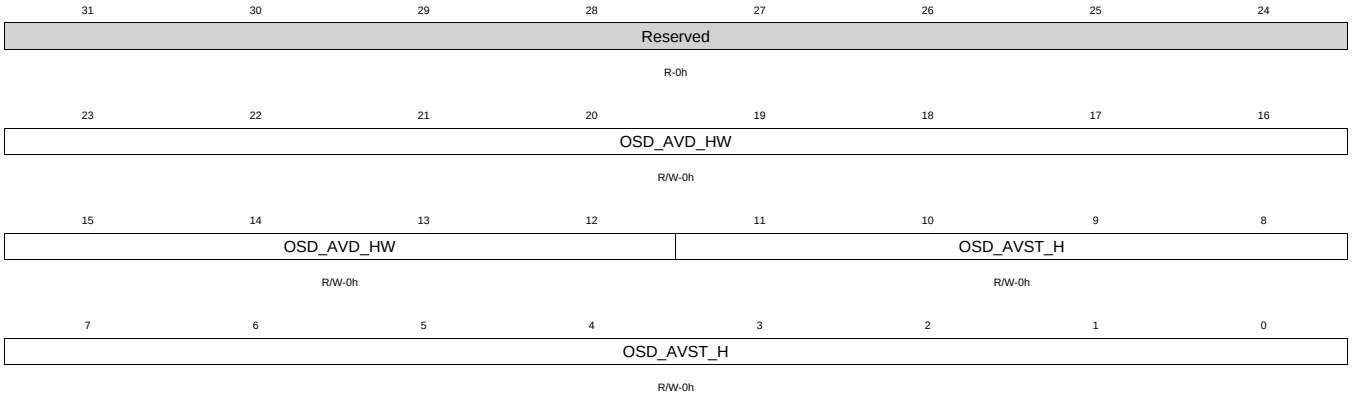

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.22 HD_VENC_D_cfg21 Register (offset = 54h) [reset = 0h]

HD_VENC_D_cfg21 is shown in Figure 12-453 and described in Table 12-367.

Compositor IF Control Register

**Figure 12-453. HD_VENC_D_cfg21 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-367. HD_VENC_D_cfg21 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-12 | OSD_AVD_HW | R/W | 0h | Defines the width of each active video line (in number of pixels). |
| 11-0 | OSD_AVST_H | R/W | 0h | This parameters defines when the first pixel on each line must appear at the input of the HD_VENC (a fixed offset from DVO_AVST_H). OSD_AVST_H = DVO_AVST_H – 8. |

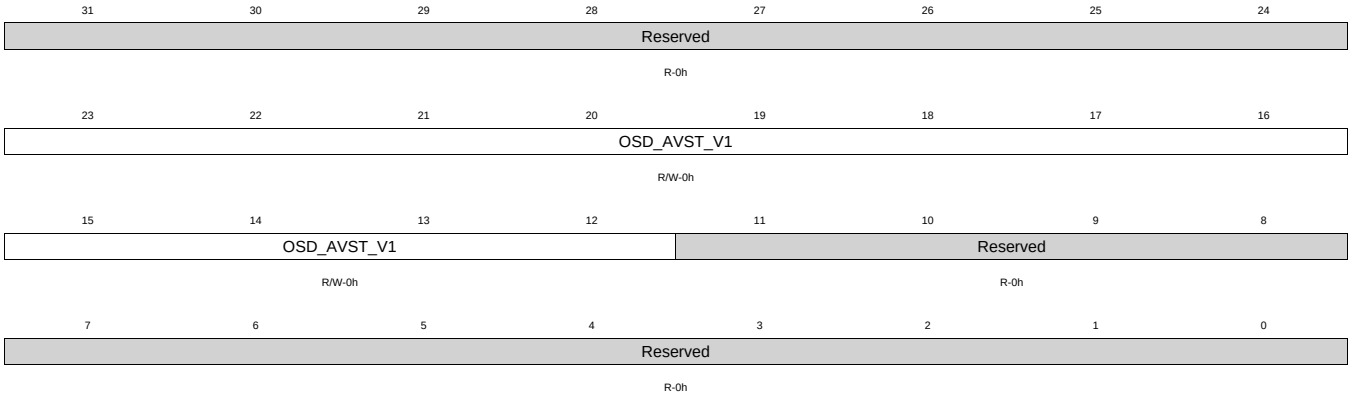

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.23 HD_VENC_D_cfg22 Register (offset = 58h) [reset = 0h]

HD_VENC_D_cfg22 is shown in Figure 12-454 and described in Table 12-368.

Compositor IF Control Register

**Figure 12-454. HD_VENC_D_cfg22 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-368. HD_VENC_D_cfg22 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-12 | OSD_AVST_V1 | R/W | 0h | In progressive mode. it defines the first active line in a frame. In interlace mode. it defines the first active line of first field in a frame (The line number starts at 0) |
| 11-0 | Reserved | R | 0h | Reserved |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.24 HD_VENC_D_cfg23 Register (offset = 5Ch) [reset = 0h]

HD_VENC_D_cfg23 is shown in Figure 12-455 and described in Table 12-369.

Compositor IF Control Register

**Figure 12-455. HD_VENC_D_cfg23 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| OSD_AVD_VW1 | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| OSD_AVD_VW1 | | | | OSD_AVST_V2 | | | |
| R/W-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| OSD_AVST_V2 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-369. HD_VENC_D_cfg23 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-12 | OSD_AVD_VW1 | R/W | 0h | Defines the number of active lines for the input frame (or first field) of the HD_VENC_D.<br>Normally, this should be same as DVO_AVD_VW1. |
| 11-0 | OSD_AVST_V2 | R/W | 0h | Defines the first active line of second field in a frame. This parameter is only used in interlace mode. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.25 HD_VENC_D_cfg24 Register (offset = 60h) [reset = 0h]

HD_VENC_D_cfg24 is shown in Figure 12-456 and described in Table 12-370.

Compositor IF Control Register

**Figure 12-456. HD_VENC_D_cfg24 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | OSD_AVD_VW2 | | | |
| R-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| OSD_AVD_VW2 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-370. HD_VENC_D_cfg24 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | Reserved | R | 0h | Reserved |
| 11-0 | OSD_AVD_VW2 | R/W | 0h | Defines the number of active lines in the second field. This parameter is only used in interface mode. |



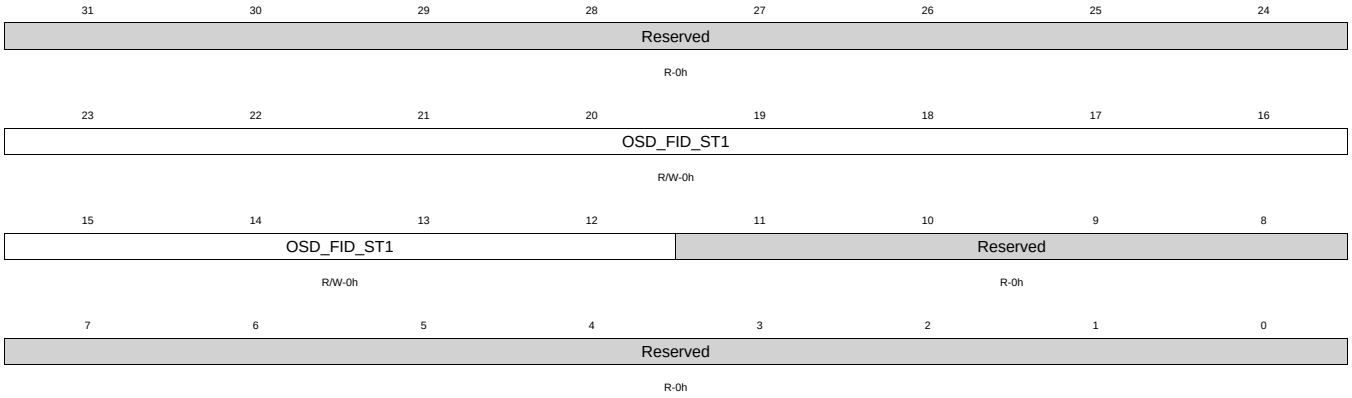
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.26 HD_VENC_D_cfg25 Register (offset = 64h) [reset = 0h]

HD_VENC_D_cfg25 is shown in Figure 12-457 and described in Table 12-371.

Compositor IF Control Register

**Figure 12-457. HD_VENC_D_cfg25 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-371. HD_VENC_D_cfg25 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | Reserved |
| 23-12 | OSD_FID_ST1 | R/W | 0h | This parameter defines the frame line (for progressive mode) or top field line (for interlaced mode) number at which the DTV_FID will switch.<br>Progressive mode:<br>OSD_FID_ST1 = int ( (cfg010.lines – cfg023.osd_avd_vw1)/3 )<br>Interlace mode:<br>OSD_FID_ST1 = int ( (cfg010.lines/2 – cfg023.osd_avd_vw1)/3 ) |
| 11-0 | Reserved | R | 0h | Reserved |

**Note:** DTV_FID switches to both progressive and interlaced mode to indicate to internal processing modules that a new frame/field is starting. The above OSD_FID_ST1 parameter setup configures the DTV_FID to switch about 1/3 into the VBI period, giving sufficient timing margin for DMA set-up and video pipeline fill-up prior to the next active video period.


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.9.27 HD_VEND_D_GAMMA_LUT Register (offset = 1000h-1FFFh) [reset = 0h]

HD_VEND_D_GAMMA_LUT is shown in Figure 12-458 and described in Table 12-372.

**Figure 12-458. HD_VEND_D_GAMMA_LUT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | lut2[a] | | | | | |
| R-0 | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| lut2[a] | | | | lut1[a] | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| lut1[a] | | | | | | lut0[a] | |
| R/W-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| lut0[a] | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 12-372. HD_VEND_D_GAMMA_LUT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | Reserved |
| 29-20 | lut2[a] | R/W | - | Value for lut2[a], where a corresponds to bits [11:2] of the source address |
| 19-10 | lut1[a] | R/W | - | Value for lu12[a], where a corresponds to bits [11:2] of the source address |
| 9-0 | lut0[a] | R/W | - | Value for lut0[a], where a corresponds to bits [11:2] of the source address |

**Note:** The reset value of the HD_VENC_D_GAMMA_LUT registers are undefined. If Gamma Correction functionality is to be used (controlled by HD_VENC_D_cfg0.BYPS_GC), this MMR area must be loaded by software before enabling HD_VENC_D.


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10 NOISE_FILTER Registers

Table 12-373 lists the memory-mapped registers for the NOISE_FILTER. All register offset addresses not listed in Table 12-373 should be considered as reserved locations and the register contents should not be modified.

**Table 12-373. NOISE_FILTER REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | nf_reg0 | NF Mode Reg | Section 12.3.10.1 |
| 4h | nf_reg1 | NF Video Size Config Reg | Section 12.3.10.2 |
| 8h | nf_reg2 | NF Spatial Strength Y Reg | Section 12.3.10.3 |
| Ch | nf_reg3 | NF Spatial Strength UV Reg | Section 12.3.10.4 |
| 10h | nf_reg4 | NF Temporal Filter Config Reg | Section 12.3.10.5 |
| 14h | nf_reg5 | NF Noise Config Reg | Section 12.3.10.6 |
| 18h | nf_reg6 | NF BW Threshold Config Reg | Section 12.3.10.7 |
| 1Ch | nf_reg7 | NF Saved Noise Frame Index Reg | Section 12.3.10.8 |
| 20h | nf_reg8 | NF Saved Noise Frame Y Data Reg | Section 12.3.10.9 |
| 24h | nf_reg9 | NF Saved Noise Frame UV Data Reg | Section 12.3.10.10 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10.1 nf_reg0 Register (offset = 0h) [reset = 0h]

nf_reg0 is shown in Figure 12-459 and described in Table 12-374.

**Figure 12-459. nf_reg0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | NF_UPDATE_FRAME_NOISE | NF_LOAD_FRAME_NOISE | NF_BYPASS_CFG | | NF_VIDEO_INDEX |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NF_VIDEO_INDEX | | | | Reserved | NF_REF_CFG | | NF_EN |
| R/W-0h | | | | R-0h | R/W-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-374. nf_reg0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | Reserved | R | 0h | Reserved (Write 0x0 only) |
| 12 | NF_UPDATE_FRAME_NOISE | R/W | 0h | NF Frame Noise Update Enable<br>0: Do not Update<br>1: Update<br>When set to 1, frameNoise_previous registers get updated at the end of current frame processing. When set to 0.. updates are not done allowing next frame to use the same initial frame noise value used by the previous frame. |
| 11 | NF_LOAD_FRAME_NOISE | R/W | 0h | NF Frame Noise Load Enable<br>When a new video is assigned to nf_video_index.. this bit needs to be set to initialize (clear or reload) internally stored frameNoise_previous registers.<br>0: Do not overwrite<br>1: Inititalize (overwrite)<br>When set to 1, values last written into NF_REG8 and NF_REG9 are used as new initial values. |
| 10-9 | NF_BYPASS_CFG | R/W | 0h | NF Bypass Configuration - Reserved for DEBUG purpose only (Important : Write only 0)<br>To perform configure DSS_NF in a bypass mode (spatial input bypass), configure the DSS_NF normally and set the spatial and temporal strength registers to 0x0. |
| 8-4 | NF_VIDEO_INDEX | R/W | 0h | NF Video Index: Index to select stored noise level (0 - 31) |
| 3 | Reserved | R | 0h | |
| 2-1 | NF_REF_CFG | R/W | 0h | NF Reference Frame Configuration<br>00: DMA enabled.. Data Passed<br>01: DMA enabled.. Output Forced to Black<br>1x: DMA disabled.. Output Forced to Black (for first pass through the NF processing with no valid REF frame in the memory) |
| 0 | NF_EN | R/W | 0h | NF Enable:<br>0: Disabled<br>1: Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10.2 nf_reg1 Register (offset = 4h) [reset = 0h]

nf_reg1 is shown in Figure 12-460 and described in Table 12-375.

**Figure 12-460. nf_reg1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | HEIGHT | | |
| R-0h | | | | | R/W-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| HEIGHT | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | WIDTH | | |
| R-0h | | | | | R/W-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WIDTH | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-375. nf_reg1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | HEIGHT | R/W | 0h | Number of lines per frame |
| 15-11 | Reserved | R | 0h | |
| 10-0 | WIDTH | R/W | 0h | Number of pixels per line |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10.3 nf_reg2 Register (offset = 8h) [reset = 00000808h]

nf_reg2 is shown in Figure 12-461 and described in Table 12-376.

**Figure 12-461. nf_reg2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SPATIAL_STRENGTH_Y_HIGH | | | | | | | |
| R/W-8h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPATIAL_STRENGTH_Y_LOW | | | | | | | |
| R/W-8h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-376. nf_reg2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-8 | SPATIAL_STRENGTH_Y_HIGH | R/W | 8h | Spatial high-frequency filter strength of Y channel. 0 means disabled. |
| 7-0 | SPATIAL_STRENGTH_Y_LOW | R/W | 8h | Spatial low-frequency filter strength of Y channel. 0 means disabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10.4 nf_reg3 Register (offset = Ch) [reset = 10101010h]

nf_reg3 is shown in Figure 12-462 and described in Table 12-377.

**Figure 12-462. nf_reg3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SPATIAL_STRENGTH_V_HIGH | | | | | | | |
| R/W-10h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SPATIAL_STRENGTH_V_LOW | | | | | | | |
| R/W-10h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SPATIAL_STRENGTH_U_HIGH | | | | | | | |
| R/W-10h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPATIAL_STRENGTH_U_LOW | | | | | | | |
| R/W-10h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-377. nf_reg3 Register Field Descriptions**

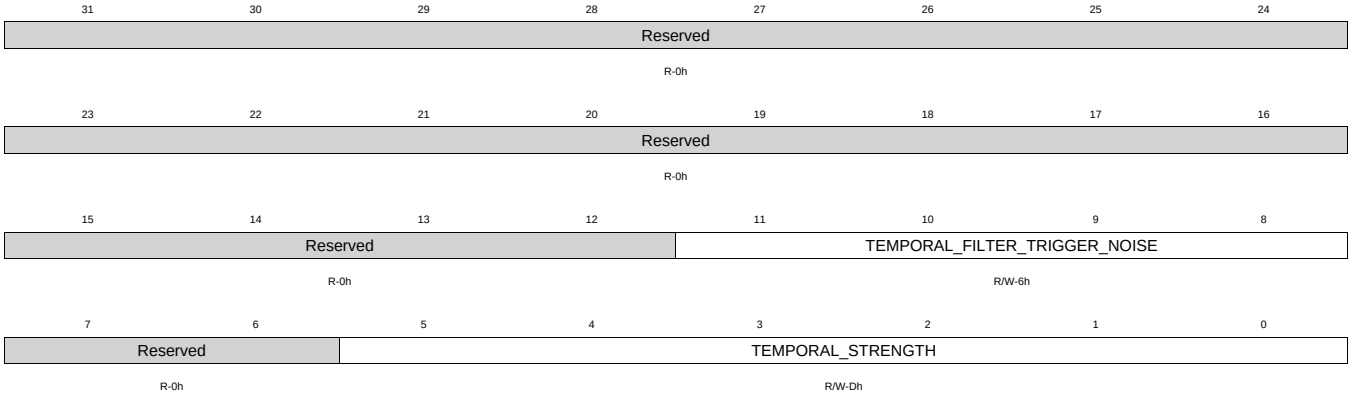
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | SPATIAL_STRENGTH_V_HIGH | R/W | 10h | Spatial high-frequency filter strength of V channel. 0 means disabled. |
| 23-16 | SPATIAL_STRENGTH_V_LOW | R/W | 10h | Spatial low-frequency filter strength of V channel. 0 means disabled. |
| 15-8 | SPATIAL_STRENGTH_U_HIGH | R/W | 10h | Spatial high-frequency filter strength of U channel. 0 means disabled. |
| 7-0 | SPATIAL_STRENGTH_U_LOW | R/W | 10h | Spatial low-frequency filter strength of U channel. 0 means disabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10.5 nf_reg4 Register (offset = 10h) [reset = 0000060Dh]

nf_reg4 is shown in Figure 12-463 and described in Table 12-378.

**Figure 12-463. nf_reg4 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-378. nf_reg4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | Reserved | R | 0h | |
| 11-8 | TEMPORAL_FILTER_TRIGGER_NOISE | R/W | 6h | The smallest noise level to fully trigger temporal filter. |
| 7-6 | Reserved | R | 0h | |
| 5-0 | TEMPORAL_STRENGTH | R/W | Dh | Temporal filter strength. 0 means disabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10.6 nf_reg5 Register (offset = 14h) [reset = 000000ADh]

nf_reg5 is shown in Figure 12-464 and described in Table 12-379.

**Figure 12-464. nf_reg5 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | MAX_NOISE |
| R-0h | | | | | | | R/W-Ah |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAX_NOISE | | | | NOISE_IIR_COEFFICIENT | | | |
| R/W-Ah | | | | R/W-Dh | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-379. nf_reg5 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | Reserved | R | 0h | |
| 8-4 | MAX_NOISE | R/W | Ah | Max of the possible noise level. |
| 3-0 | NOISE_IIR_COEFFICIENT | R/W | Dh | Noise level IIR filter coefficient. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10.7 nf_reg6 Register (offset = 18h) [reset = 00000F0Fh]

nf_reg6 is shown in Figure 12-465 and described in Table 12-380.

**Figure 12-465. nf_reg6 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | PURE_WHITE_THRESHOLD | | | | | |
| R-0h | | R/W-Fh | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PURE_BLACK_THRESHOLD | | | | | |
| R-0h | | R/W-Fh | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-380. nf_reg6 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-14 | Reserved | R | 0h | |
| 13-8 | PURE_WHITE_THRESHOLD | R/W | Fh | A tile is considered pure white if all pixel values are above pure_white_threshold. |
| 7-6 | Reserved | R | 0h | |
| 5-0 | PURE_BLACK_THRESHOLD | R/W | Fh | A tile is considered pure black if all pixel values are below pure_black_threshold. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10.8 nf_reg7 Register (offset = 1Ch) [reset = 0h]

nf_reg7 is shown in Figure 12-466 and described in Table 12-381.

**Figure 12-466. nf_reg7 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | FRAME_NOISE_READ_INDEX | | | | |
| R-0h | | | R/W-0h | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-381. nf_reg7 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | Reserved | R | 0h | |
| 4-0 | FRAME_NOISE_READ_INDEX | R/W | 0h | Saved Frame Noise Register bank index (used to select one of 32 registers when accessing NF_REG8 or NF_REG9) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10.9 nf_reg8 Register (offset = 20h) [reset = 0h]

nf_reg8 is shown in Figure 12-467 and described in Table 12-382.

**Figure 12-467. nf_reg8 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | FRAME_NOISE_Y | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FRAME_NOISE_Y | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-382. nf_reg8 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | Reserved | R | 0h | |
| 12-0 | FRAME_NOISE_Y | R/W | 0h | Saved Frame Noise Luma (y) (Read) Save Frame_noise_y selected by Frame_noise_read_index register (Write) Value to be written to the Frame_noise_y selected by NF_REG0(nf_video_index) at the beginning of a new frame |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.10.10 nf_reg9 Register (offset = 24h) [reset = 0h]

nf_reg9 is shown in Figure 12-468 and described in Table 12-383.

**Figure 12-468. nf_reg9 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | FRAME_NOISE_U | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FRAME_NOISE_U | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | FRAME_NOISE_V | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FRAME_NOISE_V | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-383. nf_reg9 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | FRAME_NOISE_U | R/W | 0h | Saved Frame Noise Chroma(cb/u) (Read) Save Frame_noise_v selected by Frame_noise_read_index register (Write) Value to be written to the Frame_noise_u selected by NF_REG0(nf_video_index) at the beginning of a new frame |
| 15-13 | Reserved | R | 0h | |
| 12-0 | FRAME_NOISE_V | R/W | 0h | Saved Frame Noise Chroma(cr/v) (Read) Save Frame_noise_v selected by Frame_noise_read_index register (Write) Value to be written to the Frame_noise_v selected by NF_REG0(nf_video_index) at the beginning of a new frame |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11 SC_M Registers

Table 12-384 lists the memory-mapped registers for the SC_M. All register offset addresses not listed in Table 12-384 should be considered as reserved locations and the register contents should not be modified.

**Table 12-384. SC_M REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | SC_M_cfg_sc0 | SC Mode Configuration Reg | Section 12.3.11.1 |
| 4h | SC_M_cfg_sc1 | SC Vertical Scaler Config Reg 0 | Section 12.3.11.2 |
| 8h | SC_M_cfg_sc2 | SC Vertical Scaler Config Reg 1 | Section 12.3.11.3 |
| Ch | SC_M_cfg_sc3 | SC Vertical Scaler Config Reg 2 | Section 12.3.11.4 |
| 10h | SC_M_cfg_sc4 | SC Target Size Reg | Section 12.3.11.5 |
| 14h | SC_M_cfg_sc5 | SC Source Size Reg | Section 12.3.11.6 |
| 18h | SC_M_cfg_sc6 | SC Vertical Scaler Config Reg 3 | Section 12.3.11.7 |
| 20h | SC_M_cfg_sc8 | SC Horizontal Scaler Config Reg 0 | Section 12.3.11.8 |
| 24h | SC_M_cfg_sc9 | SC Horizontal Scaler Config Reg 1 | Section 12.3.11.9 |
| 28h | SC_M_cfg_sc10 | SC Horizontal Scaler Config Reg 2 | Section 12.3.11.10 |
| 2Ch | SC_M_cfg_sc11 | SC Horizontal Scaler Config Reg 3 | Section 12.3.11.11 |
| 30h | SC_M_cfg_sc12 | SC Horizontal Scaler Config Reg 4 | Section 12.3.11.12 |
| 34h | SC_M_cfg_sc13 | SC Vertical Scaler Config Reg 4 | Section 12.3.11.13 |
| 44h | SC_M_cfg_sc17 | SC Vertical Scaler Config Reg 5 | Section 12.3.11.14 |
| 48h | SC_M_cfg_sc18 | SC Vertical Scaler Config Reg 6 | Section 12.3.11.15 |
| 4Ch | SC_M_cfg_sc19 | SC Peaking Filter Config Reg 0 | Section 12.3.11.16 |
| 50h | SC_M_cfg_sc20 | SC Peaking Filter Config Reg 1 | Section 12.3.11.17 |
| 54h | SC_M_cfg_sc21 | SC Peaking Filter Config Reg 2 | Section 12.3.11.18 |
| 58h | SC_M_cfg_sc22 | SC Peaking Filter Config Reg 3 | Section 12.3.11.19 |
| 5Ch | SC_M_cfg_sc23 | SC Edeg Detection Config Reg 3 | Section 12.3.11.20 |
| 60h | SC_M_cfg_sc24 | SC Trimmer Config Reg 0 | Section 12.3.11.21 |
| 64h | SC_M_cfg_sc25 | SC Trimmer Config Reg 1 | Section 12.3.11.22 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.1 SC_M_cfg_sc0 Register (offset = 0h) [reset = 0h]

SC_M_cfg_sc0 is shown in Figure 12-469 and described in Table 12-385.

**Figure 12-469. SC_M_cfg_sc0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | CFG_FID_SELFGEN |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CFG_TRIM | CFG_Y_PK_EN | Reserved | | | CFG_INTERLACE_I | CFG_HP_BYPASS | CFG_DCM_4X |
| R/W-0h | R/W-0h | R-0h | | | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFG_DCM_2X | CFG_AUTO_HS | Reserved | CFG_USE_RAV | CFG_INVT_FID | CFG_SC_BYPASS | CFG_LINEAR | CFG_INTERLACE_O |
| R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-385. SC_M_cfg_sc0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | Reserved | R | 0h | |
| 16 | CFG_FID_SELFGEN | R/W | 0h | FID self generate enable. When input is progressive and this bit is set, the SC generates self-toggling (top/bottom) output FID when performing interlacing. |
| 15 | CFG_TRIM | R/W | 0h | Trimming enable. When 1, the input image whose size is specified by orgW and orgH registers is trimmed to the size with srcW and srcH from the offset specified by offW and offH.<br>0: disable trimming<br>1: enable trimming |
| 14 | CFG_Y_PK_EN | R/W | 0h | This parameter is used by peaking block.<br>0: disable luma peaking<br>1: enable luma peaking |
| 13-11 | Reserved | R | 0h | |
| 10 | CFG_INTERLACE_I | R/W | 0h | This parameter is used by both horizontal and vertical scaling<br>0: the input video format is progressive<br>1: the input video format is interlace |
| 9 | CFG_HP_BYPASS | R/W | 0h | This parameter is used by horizontal scaling.<br>If cfg_auto_hs is 0, horizontal polyphase filter is always enabled. In this case, this register is DON'T CARE.<br>If cfg_auto_hs is 1,<br>0 : The polyphase scaler is always used regardless of the scaling ratio.<br>1 : The polyphase scaler is bypassed only when (tar_w == src_w) or (tar_w == src_w/2) or (tar_w == src_w/4) |
| 8 | CFG_DCM_4X | R/W | 0h | This parameter is used by horizontal scaling.<br>0: the 4X decimation filter is disabled<br>1: the 4X decimation filter is enabled<br>Note:<br>(1) Either 2X or 4X can be enabled, but they cannot be enabled simultaneously.<br>(2) This register is only set to 1 when it makes sense to do so. Typically, it is used when (horizontal scale ratio < 0.25).<br>(3) This register is DON'T CARE when cfg_auto_hs = 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-385. SC_M_cfg_sc0 Register Field Descriptions (continued)**

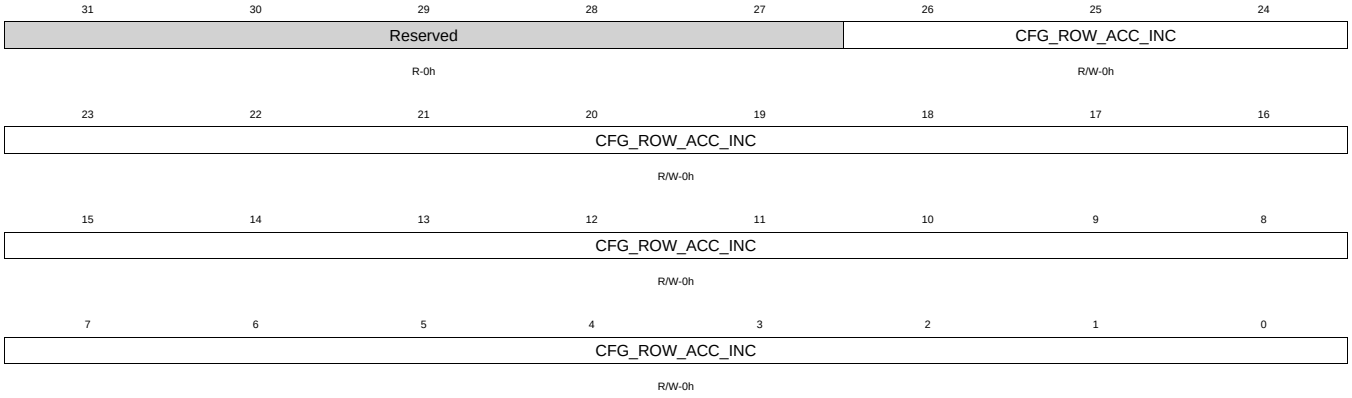
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | CFG_DCM_2X | R/W | 0h | This parameter is used by horizontal scaling.<br>0: the 2X decimation filter is disabled<br>1: the 2X decimation filter is enabled<br>Note:<br>(1) Either 2X or 4X can be enabled, but they cannot be enabled simultaneously.<br>(2) This register is only set to 1 when it makes sense to do so. Typically, it is used when (0.25 < horizontal scale ratio < 0.5).<br>(3) This register is DON'T CARE when cfg_auto_hs = 1. |
| 6 | CFG_AUTO_HS | R/W | 0h | This parameter is used by horizontal scaling.<br>0: the cfg_dcm_2x and cfg_dcm_4x bits will enable appropriate decimation filters<br>1: HW will decide whether up-scaling or down-scaling is required based on horizontal scaling ratio (SR).<br>SR > 0.5 : horizontal polyphase filter is enabled, all decimation filters are disabled<br>SR = 0.5 : dcm_2x is enabled, horizontal polyphase filter is enabled or disabled based on cfg_hp_bypass<br>0.5 > SR > 0.25 : dcm_2x and horizontal polyphase filter both are enabled<br>SR = 0.25 : dcm_4x is enabled, horizontal polyphase filter is enabled or disabled based on cfg_hp_bypass<br>0.25 > SR > 0.125 : dcm_4x and horizontal polyphase filter are both enabled<br>SR <= 0.125 : Functionally supported, but not recommended in auto mode for image quality concerns |
| 5 | Reserved | R | 0h | |
| 4 | CFG_USE_RAV | R/W | 0h | This parameter is used by vertical scaling.<br>0: Poly-phase filter will be used for the vertical scaling<br>1: Running average filter will be used for the vertical scaling (down scaling only) |
| 3 | CFG_INVT_FID | R/W | 0h | This parameter inverts the polarity of the incoming FID bit before applying to Scaler function.<br>0: treat incoming FID=0 to be bottom field and FID=1 to be top field data<br>1: treat incoming FID=1 to be bottom field and FID=0 to be top field data<br>Incoming FID is the FID value programmed in the VPDMA descriptor.<br>For this device, CFG_INVT_FID should be set to 0x1 always. |
| 2 | CFG_SC_BYPASS | R/W | 0h | This parameter is a general purpose.<br>0: Scaling module will engaged<br>1: Scaling module will be bypassed |
| 1 | CFG_LINEAR | R/W | 0h | This parameter is used by horizontal scaling.<br>0: Anamorphic scaling...<br>1: Linear scaling |
| 0 | CFG_INTERLACE_O | R/W | 0h | This parameter is used by vertical scaling.<br>0: The output format of SC is progressive<br>1: The output format of SC is interlace |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.2 SC_M_cfg_sc1 Register (offset = 4h) [reset = 0h]

SC_M_cfg_sc1 is shown in Figure 12-470 and described in Table 12-386.

**Figure 12-470. SC_M_cfg_sc1 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-386. SC_M_cfg_sc1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-0 | CFG_ROW_ACC_INC | R/W | 0h | This parameter is used by vertical scaling. It defines the increment of the row accumulator in vertical poly-phase filter. It can be calculated by following formula:<br>row_acc_inc = round(2^16 *(src_h)/(tar_h))<br>In case of interlaced input, srcH is input field height In case of interlaced output, tarH is output field height. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.3 SC_M_cfg_sc2 Register (offset = 8h) [reset = 0h]

SC_M_cfg_sc2 is shown in Figure 12-471 and described in Table 12-387.

**Figure 12-471. SC_M_cfg_sc2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_ROW_ACC_OFFSET | | | |
| R-0h | | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CFG_ROW_ACC_OFFSET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CFG_ROW_ACC_OFFSET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CFG_ROW_ACC_OFFSET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-387. SC_M_cfg_sc2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | |
| 27-0 | CFG_ROW_ACC_OFFSET | R/W | 0h | This parameter is used by vertical scaling. It defines the vertical offset during vertical scaling. In progressive mode: this offset will be applied to a frame. In interlace mode: this offset will be applied to the top field. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.4 SC_M_cfg_sc3 Register (offset = Ch) [reset = 0h]

SC_M_cfg_sc3 is shown in Figure 12-472 and described in Table 12-388.

**Figure 12-472. SC_M_cfg_sc3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_ROW_ACC_OFFSET_B | | | |
| R-0h | | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CFG_ROW_ACC_OFFSET_B | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CFG_ROW_ACC_OFFSET_B | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CFG_ROW_ACC_OFFSET_B | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-388. SC_M_cfg_sc3 Register Field Descriptions**

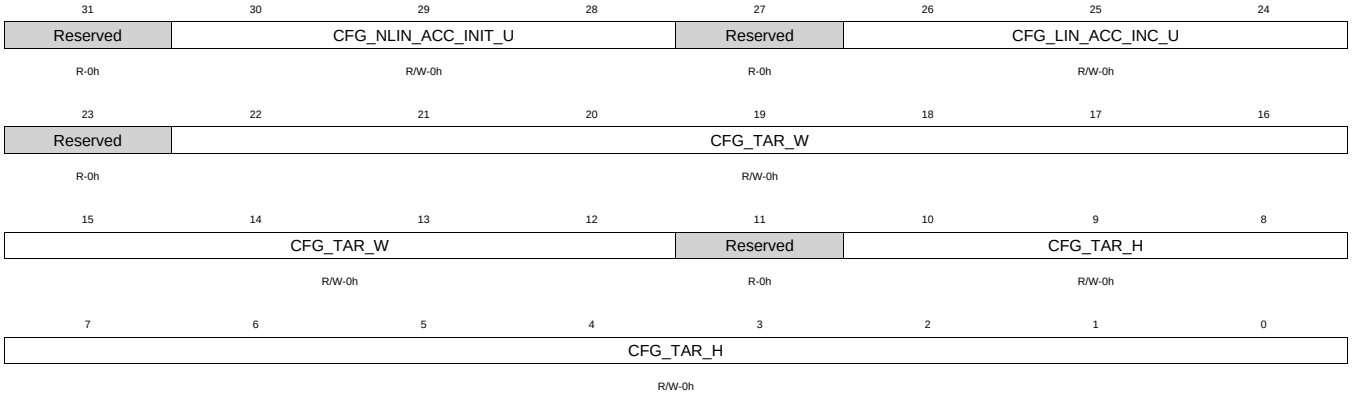
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | |
| 27-0 | CFG_ROW_ACC_OFFSET_B | R/W | 0h | This parameter is used by vertical scaling. It defines the vertical offset during vertical scaling. In progressive mode: this parameter will not be used. In interlace mode: this offset will be applied to the bottom field. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.5 SC_M_cfg_sc4 Register (offset = 10h) [reset = 0h]

SC_M_cfg_sc4 is shown in Figure 12-473 and described in Table 12-389.

**Figure 12-473. SC_M_cfg_sc4 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-389. SC_M_cfg_sc4 Register Field Descriptions**

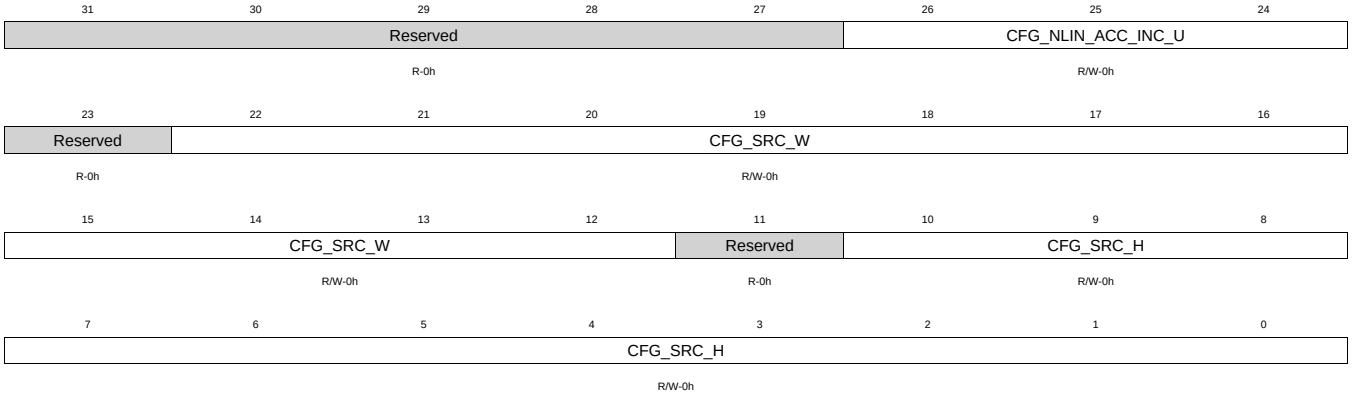
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Reserved | R | 0h | |
| 30-28 | CFG_NLIN_ACC_INIT_U | R/W | 0h | This parameter is used by horizontal scaling. The 3 MSBbits of 'nlin_acc_init' that is defined in SC_CFG10 |
| 27 | Reserved | R | 0h | |
| 26-24 | CFG_LIN_ACC_INC_U | R/W | 0h | This parameter is used by horizontal scaling. The 3 MSBbits of 'lin_acc_inc' that is defined in SC_CFG9 |
| 23 | Reserved | R | 0h | |
| 22-12 | CFG_TAR_W | R/W | 0h | This parameter is a general purpose. Scaled target picture width.. unit is pixel... This parameter defines the final output picture size |
| 11 | Reserved | R | 0h | |
| 10-0 | CFG_TAR_H | R/W | 0h | This parameter is a general purpose. Scaled target picture height.. unit is line... This parameter defines the final output picture size. For the interlace output.. it should be the number of lines per field. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.6 SC_M_cfg_sc5 Register (offset = 14h) [reset = 0h]

SC_M_cfg_sc5 is shown in Figure 12-474 and described in Table 12-390.

**Figure 12-474. SC_M_cfg_sc5 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-390. SC_M_cfg_sc5 Register Field Descriptions**

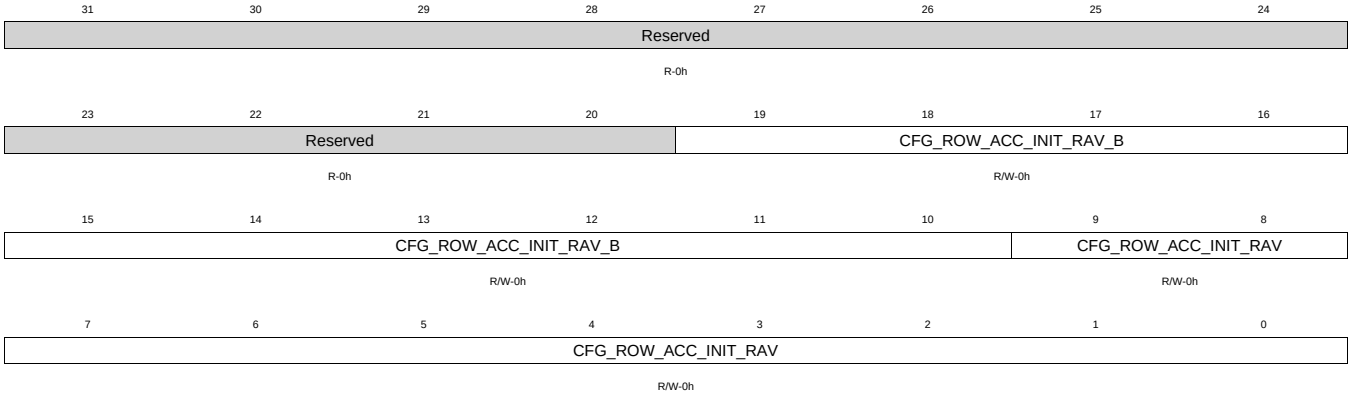
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-24 | CFG_NLIN_ACC_INC_U | R/W | 0h | This parameter is used by horizontal scaling. The 3 MSBbits of 'nlin_acc_inc' that is defined in SC_CFG11 |
| 23 | Reserved | R | 0h | |
| 22-12 | CFG_SRC_W | R/W | 0h | This parameter is a general purpose. This parameter defines the width of the source image |
| 11 | Reserved | R | 0h | |
| 10-0 | CFG_SRC_H | R/W | 0h | This parameter is a general purpose. This parameter defines the height of the source image. For the interlace input.. it should be the number of lines per field. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.7 SC_M_cfg_sc6 Register (offset = 18h) [reset = 0h]

SC_M_cfg_sc6 is shown in Figure 12-475 and described in Table 12-391.

**Figure 12-475. SC_M_cfg_sc6 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-391. SC_M_cfg_sc6 Register Field Descriptions**

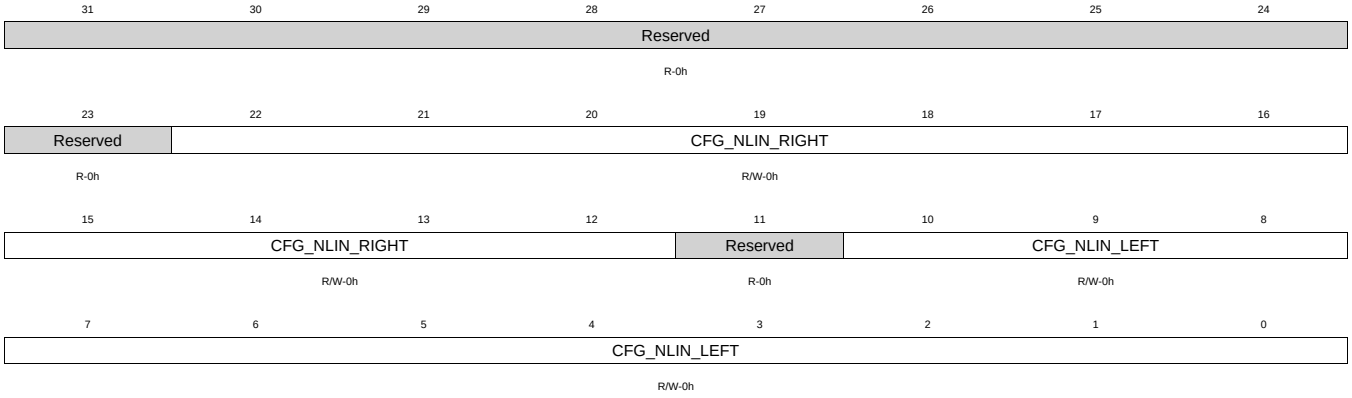
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | Reserved | R | 0h | |
| 19-10 | CFG_ROW_ACC_INIT_RAV_B | R/W | 0h | This parameter is used by vertical scaling... it is used only when the input is interlace format. In vertical down scaling.. the running average filter is applied. This parameter sets the initialization value of the row accumulator in running average filter (for bottom field of interlace format) |
| 9-0 | CFG_ROW_ACC_INIT_RAV | R/W | 0h | This parameter is used by vertical scaling. In vertical down scaling.. the running average filter is applied. This parameter sets the initialization value of the row accumulator in running average filter (for progressive format or top field of interlace format) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.8 SC_M_cfg_sc8 Register (offset = 20h) [reset = 0h]

SC_M_cfg_sc8 is shown in Figure 12-476 and described in Table 12-392.

**Figure 12-476. SC_M_cfg_sc8 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-392. SC_M_cfg_sc8 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | Reserved | R | 0h | |
| 22-12 | CFG_NLIN_RIGHT | R/W | 0h | This parameter is used by horizontal scaling. In anamorphic mode.. this parameter defines the width of the strip on right-hand side. In other words.. it defines the location of the last pixel where the linear scaling is ended. The unit is the 'pixel location' in an active video line. This parameter will not be used in linear scaling |
| 11 | Reserved | R | 0h | |
| 10-0 | CFG_NLIN_LEFT | R/W | 0h | This parameter is used by horizontal scaling. In anamorphic mode.. this parameter defines the width of the strip on left-hand side. In other words.. it defines the location of the last pixel in the left-side-nonlinear strip. The unit is the 'pixel location' in an active video line. This parameter will not be used in linear scaling |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.9 SC_M_cfg_sc9 Register (offset = 24h) [reset = 0h]

SC_M_cfg_sc9 is shown in Figure 12-477 and described in Table 12-393.

**Figure 12-477. SC_M_cfg_sc9 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CFG_LIN_ACC_INC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-393. SC_M_cfg_sc9 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CFG_LIN_ACC_INC | R/W | 0h | This parameter is used by horizontal scaling. It defines the increment of the linear accumulator.<br>if ((cfg_dcm_2x==0) && (cfg_dcm_4x==0) ) then lin_acc_inc = round(2^24*(srcWi) /(tarWi))<br>if ((cfg_dcm_2x==1) && (cfg_dcm_4x==0) ) then lin_acc_inc = round(2^24*(srcWi/2) /(tarWi))<br>if ((cfg_dcm_2x==0) && (cfg_dcm_4x==1) ) then lin_acc_inc = round(2^24*(srcWi/4) /(tarWi))<br>where srcWi and tarWi are the inner source width and the inner target width respectively. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.10 SC_M_cfg_sc10 Register (offset = 28h) [reset = 0h]

SC_M_cfg_sc10 is shown in Figure 12-478 and described in Table 12-394.

**Figure 12-478. SC_M_cfg_sc10 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CFG_NLIN_ACC_INIT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-394. SC_M_cfg_sc10 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CFG_NLIN_ACC_INIT | R/W | 0h | This parameter is used by horizontal scaling. It is used by nonlinear scaling only. It defines the initialization value of the nonlinear accumulator. nlin_acc_init = K*(1-2*d) Here the definitions of K and d are the same as in SC_CFG11 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.11 SC_M_cfg_sc11 Register (offset = 2Ch) [reset = 0h]

SC_M_cfg_sc11 is shown in Figure 12-479 and described in Table 12-395.

**Figure 12-479. SC_M_cfg_sc11 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CFG_NLIN_ACC_INC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-395. SC_M_cfg_sc11 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CFG_NLIN_ACC_INC | R/W | 0h | This parameter is used by horizontal scaling. It is used by nonlinear scaling only. It defines the increment of the nonlinear accumulator.<br>if upscaling then<br>d = 0<br>if Ltar !=0 then<br>K =round[2^24*Lsrc/(Ltar*Ltar) ]<br>where Lsrc= (srcW-srcWi)/2<br>else K = 0<br>elseif downscaling<br>d = (tarW-1)/2<br>if Ltar!=0 then K = round[ 2^24 * Lsrc / (Ltar*(Ltar-2d))]<br>where Lsrc= (srcW-srcWi)/(2n) and n=1..2 or 4<br>else K = 0<br>nlin_acc_inc = 2*K (negative for downscaling) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.12 SC_M_cfg_sc12 Register (offset = 30h) [reset = 0h]

SC_M_cfg_sc12 is shown in Figure 12-480 and described in Table 12-396.

**Figure 12-480. SC_M_cfg_sc12 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | CFG_COL_ACC_OFFSET |
| R-0h | | | | | | | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CFG_COL_ACC_OFFSET | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CFG_COL_ACC_OFFSET | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CFG_COL_ACC_OFFSET | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-396. SC_M_cfg_sc12 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | Reserved | R | 0h | |
| 24-0 | CFG_COL_ACC_OFFSET | R/W | 0h | This parameter is used in horizontal scaling. It defines the luma accumulator's offset. Normally this parameter can be set as 0 if no horizontal offset is involved. In some applications.. such as Pan and Scan.. a corresponding offset value should be set. The format is 1.24. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.13 SC_M_cfg_sc13 Register (offset = 34h) [reset = 0h]

SC_M_cfg_sc13 is shown in Figure 12-481 and described in Table 12-397.

**Figure 12-481. SC_M_cfg_sc13 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | CFG_SC_FACTOR_RAV | |
| R-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFG_SC_FACTOR_RAV | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-397. SC_M_cfg_sc13 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | Reserved | R | 0h | |
| 9-0 | CFG_SC_FACTOR_RAV | R/W | 0h | This parameter is used by vertical scaling. Vertical scaling factor: It is defined as following: 1024*tarH/srcH. It is used for downscaling by the running average filter |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.14 SC_M_cfg_sc17 Register (offset = 44h) [reset = 0h]

SC_M_cfg_sc17 is shown in Figure 12-482 and described in Table 12-398.

**Figure 12-482. SC_M_cfg_sc17 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 12-398. SC_M_cfg_sc17 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.15 SC_M_cfg_sc18 Register (offset = 48h) [reset = 0h]

SC_M_cfg_sc18 is shown in Figure 12-483 and described in Table 12-399.

**Figure 12-483. SC_M_cfg_sc18 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 12-399. SC_M_cfg_sc18 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.16 SC_M_cfg_sc19 Register (offset = 4Ch) [reset = 0h]

SC_M_cfg_sc19 is shown in Figure 12-484 and described in Table 12-400.

**Figure 12-484. SC_M_cfg_sc19 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CFG_HPF_COEF3 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CFG_HPF_COEF2 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CFG_HPF_COEF1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFG_HPF_COEF0 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-400. SC_M_cfg_sc19 Register Field Descriptions**

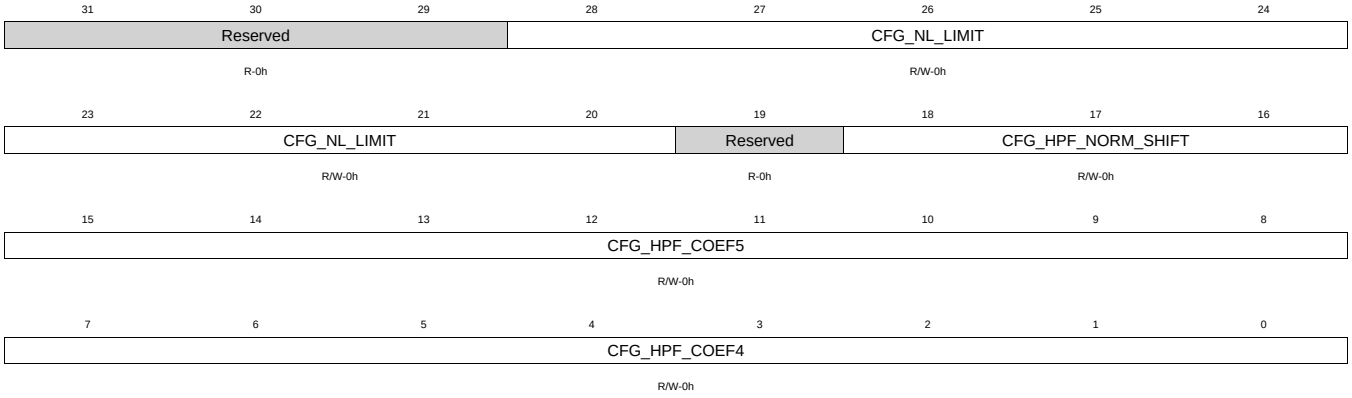
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | CFG_HPF_COEF3 | R/W | 0h | This parameter is used by the peaking block. Defines the coefficient 3 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift. |
| 23-16 | CFG_HPF_COEF2 | R/W | 0h | This parameter is used by the peaking block. Defines the coefficient 2 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift. |
| 15-8 | CFG_HPF_COEF1 | R/W | 0h | This parameter is used by the peaking block. Defines the coefficient 1 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift. |
| 7-0 | CFG_HPF_COEF0 | R/W | 0h | This parameter is used by the peaking block. Defines the coefficient 0 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.17 SC_M_cfg_sc20 Register (offset = 50h) [reset = 0h]

SC_M_cfg_sc20 is shown in Figure 12-485 and described in Table 12-401.

**Figure 12-485. SC_M_cfg_sc20 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-401. SC_M_cfg_sc20 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-20 | CFG_NL_LIMIT | R/W | 0h | This parameter is used by the peaking block. The maximum of clipping. |
| 19 | Reserved | R | 0h | |
| 18-16 | CFG_HPF_NORM_SHIFT | R/W | 0h | This parameter is used by the peaking block. Defines the decimal point of the hpf coefficient. |
| 15-8 | CFG_HPF_COEF5 | R/W | 0h | This parameter is used by the peaking block. Defines the coefficient 5 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm. |
| 7-0 | CFG_HPF_COEF4 | R/W | 0h | This parameter is used by the peaking block. Defines the coefficient 4 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.18 SC_M_cfg_sc21 Register (offset = 54h) [reset = 0h]

SC_M_cfg_sc21 is shown in Figure 12-486 and described in Table 12-402.

**Figure 12-486. SC_M_cfg_sc21 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CFG_NL_LO_SLOPE | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | CFG_NL_LO_THR |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFG_NL_LO_THR | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-402. SC_M_cfg_sc21 Register Field Descriptions**

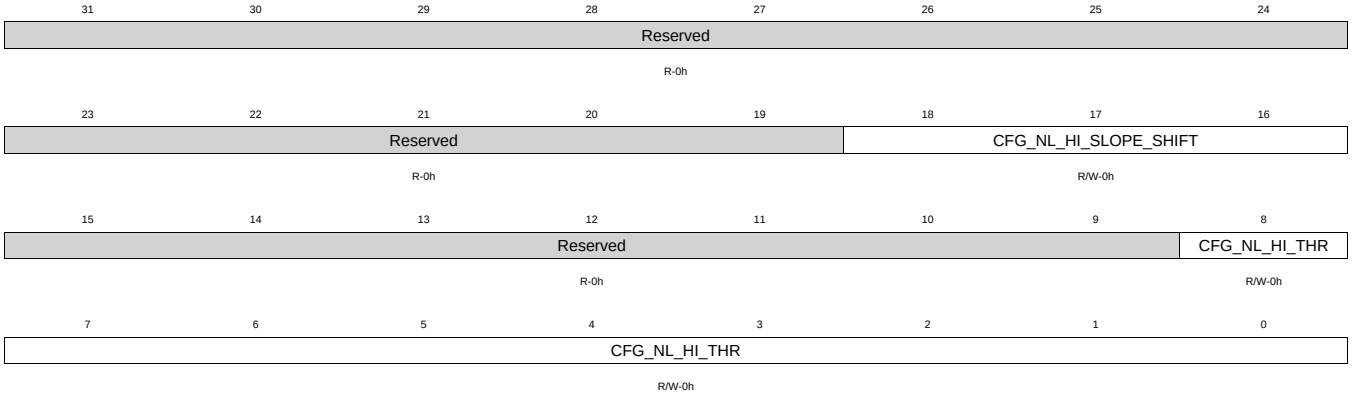
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-16 | CFG_NL_LO_SLOPE | R/W | 0h | This parameter is used by the peaking block. Slope of the nonlinear peaking function. The format is fixed point 4.4. |
| 15-9 | Reserved | R | 0h | |
| 8-0 | CFG_NL_LO_THR | R/W | 0h | This parameter is used by the peaking block. Threshold for the nonlinear peaking function. Must be 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.19 SC_M_cfg_sc22 Register (offset = 58h) [reset = 0h]

SC_M_cfg_sc22 is shown in Figure 12-487 and described in Table 12-403.

**Figure 12-487. SC_M_cfg_sc22 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-403. SC_M_cfg_sc22 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | Reserved | R | 0h | |
| 18-16 | CFG_NL_HI_SLOPE_SHIFT | R/W | 0h | This parameter is used by the peaking block. Slope of the nonlinear peaking function. The gain is 2^(nl_hi_slope_shift-3). |
| 15-9 | Reserved | R | 0h | |
| 8-0 | CFG_NL_HI_THR | R/W | 0h | This parameter is used by the peaking block. Threshold for the nonlinear peaking function. Must be nl_hi_thr. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.20 SC_M_cfg_sc23 Register (offset = 5Ch) [reset = 0h]

SC_M_cfg_sc23 is shown in Figure 12-488 and described in Table 12-404.

**Figure 12-488. SC_M_cfg_sc23 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 12-404. SC_M_cfg_sc23 Register Field Descriptions**

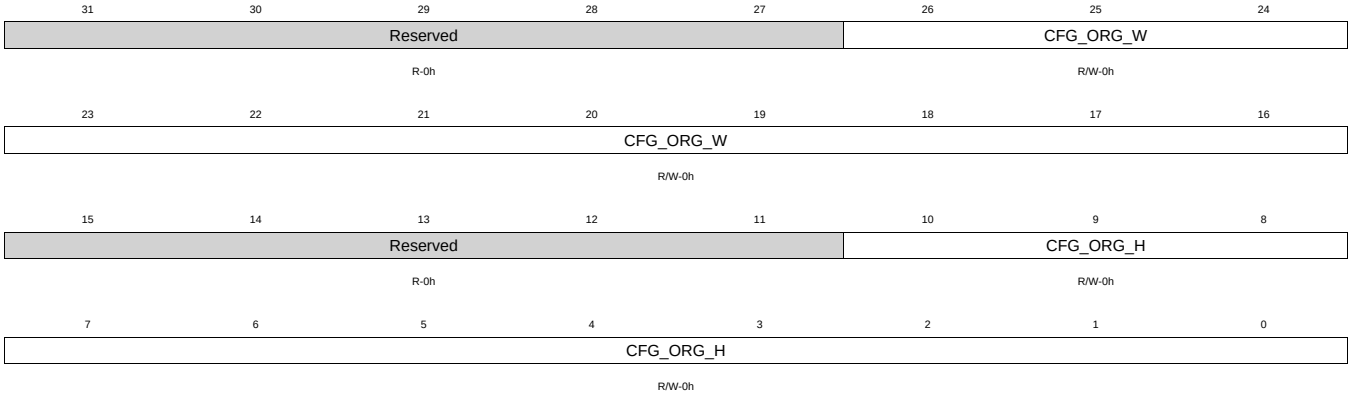
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | Reserved | R | 0h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.21 SC_M_cfg_sc24 Register (offset = 60h) [reset = 0h]

SC_M_cfg_sc24 is shown in Figure 12-489 and described in Table 12-405.

**Figure 12-489. SC_M_cfg_sc24 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-405. SC_M_cfg_sc24 Register Field Descriptions**

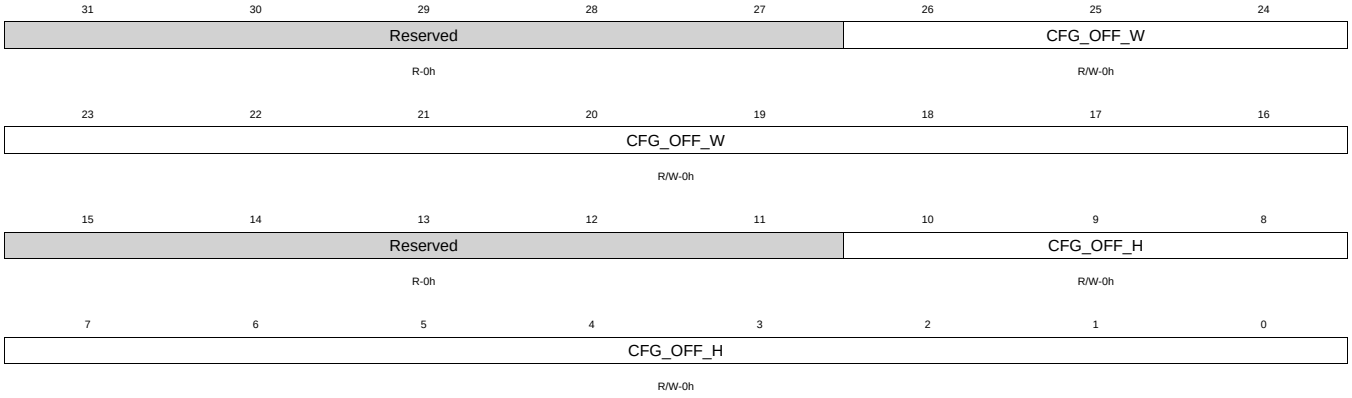
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | CFG_ORG_W | R/W | 0h | This parameter is used by the trimmer. Width of the original input image. |
| 15-11 | Reserved | R | 0h | |
| 10-0 | CFG_ORG_H | R/W | 0h | This parameter is used by the trimmer. Height of the original input image. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.11.22 SC_M_cfg_sc25 Register (offset = 64h) [reset = 0h]

SC_M_cfg_sc25 is shown in Figure 12-490 and described in Table 12-406.

**Figure 12-490. SC_M_cfg_sc25 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-406. SC_M_cfg_sc25 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | CFG_OFF_W | R/W | 0h | This parameter is used by the trimmer. Horizontal offset from the left of the original input image. |
| 15-11 | Reserved | R | 0h | |
| 10-0 | CFG_OFF_H | R/W | 0h | This parameter is used by the trimmer. Vertical offset from the top of the original input image. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12 SD_VENC Registers

Table 12-407 lists the memory-mapped registers for the SD_VENC. All register offset addresses not listed in Table 12-407 should be considered as reserved locations and the register contents should not be modified.

**Table 12-407. SD_VENC REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | SD_VENC_pid | Revision Register | Section 12.3.12.1 |
| 4h | SD_VENC_vmod | VENC Mode | Section 12.3.12.2 |
| 8h | SD_VENC_slave | Slave Control | Section 12.3.12.3 |
| Ch | SD_VENC_size | Picture Size | Section 12.3.12.4 |
| 10h | SD_VENC_pol | Polarity | Section 12.3.12.5 |
| 14h | SD_VENC_dtvs0 | DTV Sync Timing 0 | Section 12.3.12.6 |
| 18h | SD_VENC_dtvs1 | DTV Sync Timing 1 | Section 12.3.12.7 |
| 1Ch | SD_VENC_dtvs2 | DTV Sync Timing 2 | Section 12.3.12.8 |
| 20h | SD_VENC_dtvs3 | DTV Sync Timing 3 | Section 12.3.12.9 |
| 24h | SD_VENC_dtvs4 | DTV Sync Timing 4 | Section 12.3.12.10 |
| 28h | SD_VENC_dtvs5 | DTV Sync Timing 5 | Section 12.3.12.11 |
| 2Ch | SD_VENC_dtvs6 | DTV Sync Timing 6 | Section 12.3.12.12 |
| 30h | SD_VENC_dtvs7 | DTV Sync Timing 7 | Section 12.3.12.13 |
| 54h | SD_VENC_tvdetgp0 | TVDETGP Timing 0 | Section 12.3.12.14 |
| 58h | SD_VENC_tvdetgp1 | TVDETGP Timing 1 | Section 12.3.12.15 |
| 5Ch | SD_VENC_irq0 | IRQ Timing | Section 12.3.12.16 |
| 80h | SD_VENC_estat | Encoder Status | Section 12.3.12.17 |
| 84h | SD_VENC_ectl | Encoder Control | Section 12.3.12.18 |
| 88h | SD_VENC_etmg0 | Encoder Timing 0 | Section 12.3.12.19 |
| 8Ch | SD_VENC_etmg1 | Encoder Timing 1 | Section 12.3.12.20 |
| 90h | SD_VENC_etmg2 | Encoder Timing 2 | Section 12.3.12.21 |
| 94h | SD_VENC_etmg3 | Encoder Timing 3 | Section 12.3.12.22 |
| 98h | SD_VENC_etmg4 | Encoder Timing 4 | Section 12.3.12.23 |
| 9Ch | SD_VENC_cvbs0 | CVBS Control 0 | Section 12.3.12.24 |
| A0h | SD_VENC_cvbs1 | CVBS Control 1 | Section 12.3.12.25 |
| ACh | SD_VENC_ccsc0 | CVBS Color Space Conversion 0 | Section 12.3.12.26 |
| B0h | SD_VENC_ccsc1 | CVBS Color Space Conversion 1 | Section 12.3.12.27 |
| B4h | SD_VENC_ccsc2 | CVBS Color Space Conversion 2 | Section 12.3.12.28 |
| B8h | SD_VENC_ccsc3 | CVBS Color Space Conversion 3 | Section 12.3.12.29 |
| BCh | SD_VENC_ccsc4 | CVBS Color Space Conversion 4 | Section 12.3.12.30 |
| C0h | SD_VENC_ccsc5 | CVBS Color Space Conversion 5 | Section 12.3.12.31 |
| C4h | SD_VENC_ccsc6 | CVBS Color Space Conversion 6 | Section 12.3.12.32 |
| C8h | SD_VENC_ccsc7 | CVBS Color Space Conversion 7 | Section 12.3.12.33 |
| CCh | SD_VENC_ccsc8 | CVBS Color Space Conversion 8 | Section 12.3.12.34 |
| F4h | SD_VENC_cygclp | CVBS Y Clip | Section 12.3.12.35 |
| F8h | SD_VENC_cubclp | CVBS U Clip | Section 12.3.12.36 |
| FCh | SD_VENC_cvrclp | CVBS V Clip | Section 12.3.12.37 |
| 10Ch | SD_VENC_ylpf0 | CVBS Luma LPF Coefficient 0 | Section 12.3.12.38 |
| 110h | SD_VENC_ylpf1 | CVBS Luma LPF Coefficient 1 | Section 12.3.12.39 |
| 114h | SD_VENC_clpf0 | CVBS Chroma LPF Coefficient 0 | Section 12.3.12.40 |
| 118h | SD_VENC_clpf1 | CVBS Chroma LPF Coefficient 1 | Section 12.3.12.41 |
| 11Ch | SD_VENC_upf0 | 2x Upsampling Coefficient 0 | Section 12.3.12.42 |
| 120h | SD_VENC_upf1 | 2x Upsampling Coefficient 1 | Section 12.3.12.43 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-407. SD_VENC REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 124h | SD_VENC_l21ctl | Line 21 Control | Section 12.3.12.44 |
| 128h | SD_VENC_l21do | Line 21 Data Even Field | Section 12.3.12.45 |
| 12Ch | SD_VENC_l21de | Line 21 Data Odd Field | Section 12.3.12.46 |
| 130h | SD_VENC_wss | WSS Control | Section 12.3.12.47 |
| 148h | SD_VENC_scctl0 | Sub-carrier Frequency Control 0 | Section 12.3.12.48 |
| 14Ch | SD_VENC_scctl1 | Sub-carrier Frequency Control 1 | Section 12.3.12.49 |
| 150h | SD_VENC_dacsel | DAC Output Signal Select | Section 12.3.12.50 |
| 154h | SD_VENC_dupf0 | DAC 2x Upsampling Coefficient 0 | Section 12.3.12.51 |
| 158h | SD_VENC_dupf1 | DAC 2x Upsampling Coefficient 1 | Section 12.3.12.52 |
| 15Ch | SD_VENC_dactst | DAC Test Mode | Section 12.3.12.53 |
| 1FCh | SD_VENC_vtest | Internal Test Register | Section 12.3.12.54 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.1 SD_VENC_pid Register (offset = 0h) [reset = 4FFF0000h]

SD_VENC_pid is shown in Figure 12-491 and described in Table 12-408.

Revision Register

**Figure 12-491. SD_VENC_pid Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-4FFF0000h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-408. SD_VENC_pid Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | PID | R | 4FFF0000h | Revision ID. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.2 SD_VENC_vmod Register (offset = 4h) [reset = 4h]

SD_VENC_vmod is shown in Figure 12-492 and described in Table 12-409.

VENC Mode

**Figure 12-492. SD_VENC_vmod Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | DIIV | CBAR | UEL | ITLC | Reserved | VIEN |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-409. SD_VENC_vmod Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | Reserved |
| 5 | DIIV | R/W | 0h | Input data inversion<br>0: No inversion<br>1: Input inverted |
| 4 | CBAR | R/W | 0h | Color-bar mode<br>0: Normal output<br>1: Color-bar output |
| 3 | UEL | R/W | 0h | Unequal line per field. When 1, the line per field alternately toggles between (VITV-1)/2 and (VITV+1)/2. When 0, the line per field is always VITV/2. Effective in interlace mode (ITLC=1).<br>0: Normal<br>1: Unequal lines per field. |
| 2 | ITLC | R/W | 1h | Scan mode.<br>0: Progressive<br>1: Interlace |
| 1 | Reserved | R | 0h | Reserved |
| 0 | VIEN | R/W | 0h | Video encoder enable. Setting 1 brings this module into operation. Setting 0 resets internal circuits in this module.<br>0: Video encoder reset<br>1: Video encoder enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.3 SD_VENC_slave Register (offset = 8h) [reset = 0h]

SD_VENC_slave is shown in Figure 12-493 and described in Table 12-410.

Slave Control

**Figure 12-493. SD_VENC_slave Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | FMD | | FIP | VIP | HIP | SLV |
| R-0h | | R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-410. SD_VENC_slave Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5-4 | FMD | R/W | 0h | Field detection mode. Effective in slave operation (SLAVE=1). 0: latch external field at external vsync rise edge. 1: use raw external field 2: use external vsync as field ID 3: detect external vsync phase |
| 3 | FIP | R/W | 0h | FID input polarity. Effective in slave operation (SLAVE=1). External field signal is inverted at EXFEN=1. 0: Non-inverse 1: Inverse |
| 2 | VIP | R/W | 0h | VSYNC input polarity. Effective in slave operation (SLAVE=1). 0: Active H 1: Active L |
| 1 | HIP | R/W | 0h | HSYNC input polarity. Effective in slave operation (SLAVE=1). 0: Active H 1: Active L |
| 0 | SLV | R/W | 0h | Master/slave select. Set 1 to operate this module in slave mode in synchronization with external sync signal. 0: Master mode 1: Slave mode |

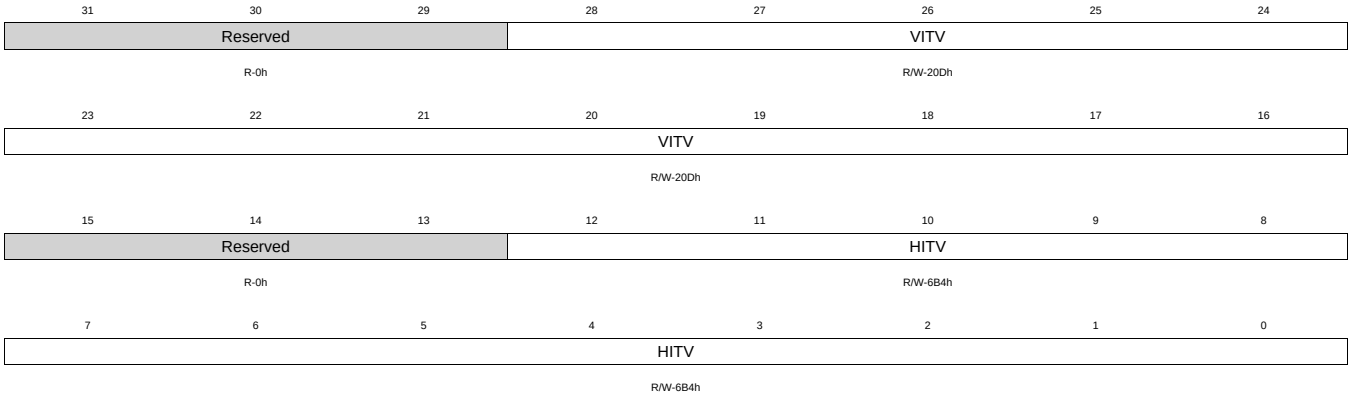

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.4 SD_VENC_size Register (offset = Ch) [reset = 020D06B4h]

SD_VENC_size is shown in Figure 12-494 and described in Table 12-411.

Picture Size

**Figure 12-494. SD_VENC_size Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-411. SD_VENC_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | VITV | R/W | 20Dh | Vertical interval. Specify the number of lines per frame. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | HITV | R/W | 6B4h | Horizontal interval. Specify the number of clocks per line. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.5 SD_VENC_pol Register (offset = 10h) [reset = 0h]

SD_VENC_pol is shown in Figure 12-495 and described in Table 12-412.

Polarity

**Figure 12-495. SD_VENC_pol Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | TVDETGP_POL |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | | | DTV_AVID_POL | DTV_FID_POL | DTV_VS_POL | DTV_HS_POL |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-412. SD_VENC_pol Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | Reserved | R | 0h | |
| 8 | TVDETGP_POL | R/W | 0h | TVDETGP output polarity. 0: Active H 1: Active L |
| 7-4 | Reserved | R | 0h | |
| 3 | DTV_AVID_POL | R/W | 0h | DTV AVID output polarity. 0: Active H 1: Active L |
| 2 | DTV_FID_POL | R/W | 0h | DTV FID output polarity 0: Non-inverse 1: Inverse |
| 1 | DTV_VS_POL | R/W | 0h | DTV VSYNC output polarity. 0: Active H 1: Active L |
| 0 | DTV_HS_POL | R/W | 0h | DTV HSYNC output polarity 0: Active H 1: Active L |



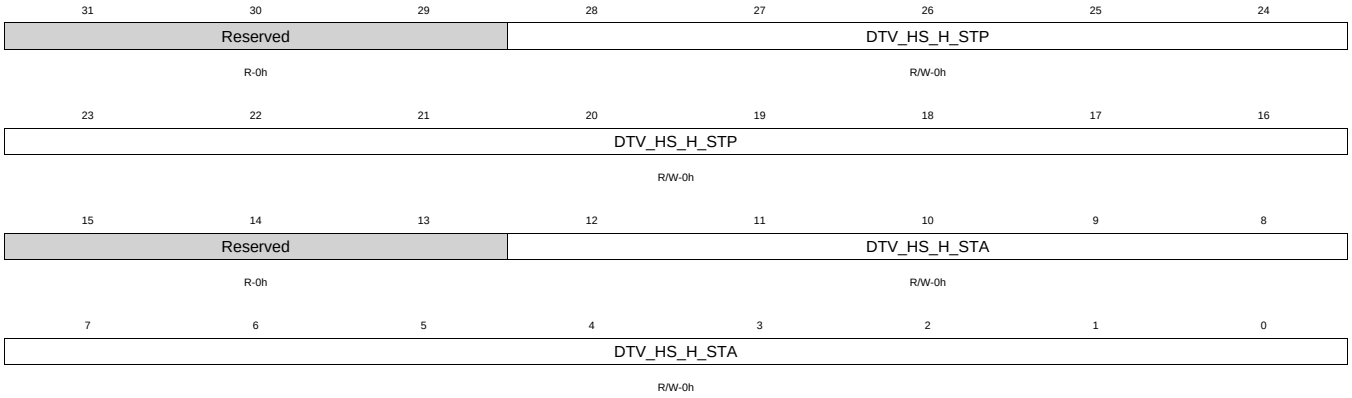
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.6 SD_VENC_dtvs0 Register (offset = 14h) [reset = 0h]

SD_VENC_dtvs0 is shown in Figure 12-496 and described in Table 12-413.

DTV Sync Timing 0

**Figure 12-496. SD_VENC_dtvs0 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-413. SD_VENC_dtvs0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | DTV_HS_H_STP | R/W | 0h | DTV HSYNC output stop pixel. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | DTV_HS_H_STA | R/W | 0h | DTV HSYNC output start pixel. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.7 SD_VENC_dtvs1 Register (offset = 18h) [reset = 0h]

SD_VENC_dtvs1 is shown in Figure 12-497 and described in Table 12-414.

DTV Sync Timing 1

**Figure 12-497. SD_VENC_dtvs1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | DTV_VS_H_STP | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DTV_VS_H_STP | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | DTV_VS_H_STA | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DTV_VS_H_STA | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-414. SD_VENC_dtvs1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | DTV_VS_H_STP | R/W | 0h | DTV VSYNC output stop pixel. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | DTV_VS_H_STA | R/W | 0h | DTV VSYNC output start pixel. |



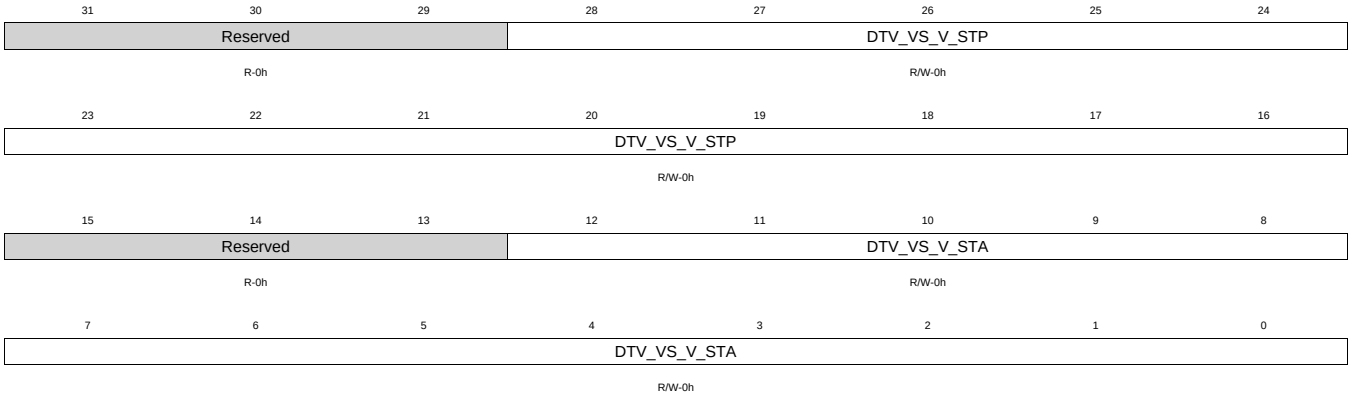
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.8 SD_VENC_dtvs2 Register (offset = 1Ch) [reset = 0h]

SD_VENC_dtvs2 is shown in Figure 12-498 and described in Table 12-415.

DTV Sync Timing 2

**Figure 12-498. SD_VENC_dtvs2 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-415. SD_VENC_dtvs2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | DTV_VS_V_STP | R/W | 0h | DTV VSYNC output stop line. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | DTV_VS_V_STA | R/W | 0h | DTV VSYNC output start line. |

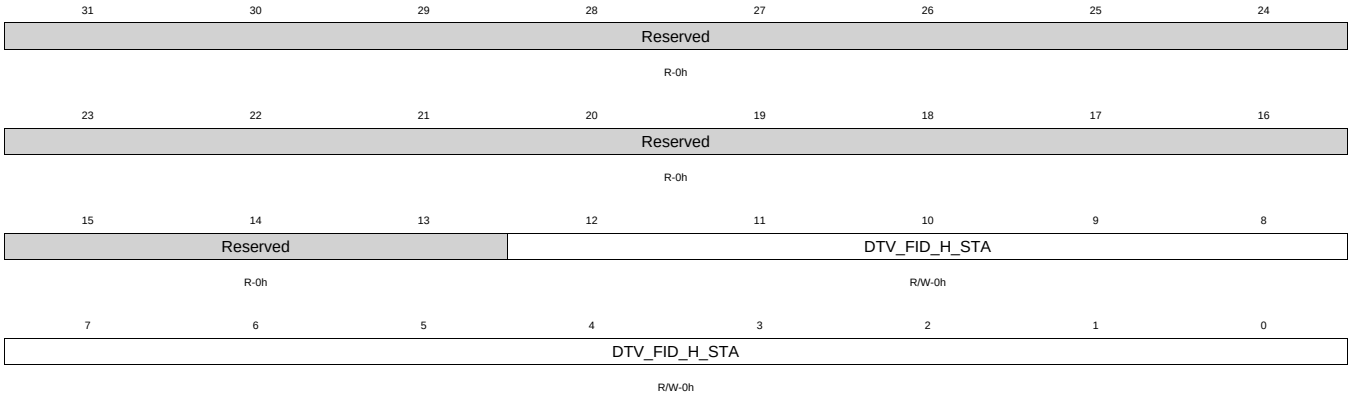

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.9 SD_VENC_dtvs3 Register (offset = 20h) [reset = 0h]

SD_VENC_dtvs3 is shown in Figure 12-499 and described in Table 12-416.

DTV Sync Timing 3

**Figure 12-499. SD_VENC_dtvs3 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-416. SD_VENC_dtvs3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | Reserved | R | 0h | |
| 12-0 | DTV_FID_H_STA | R/W | 0h | DTV FID output toggle pixel position. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.10 SD_VENC_dtvs4 Register (offset = 24h) [reset = 20000000h]

SD_VENC_dtvs4 is shown in Figure 12-500 and described in Table 12-417.

DTV Sync Timing 4

**Figure 12-500. SD_VENC_dtvs4 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | DTV_FID_F_STA1 | DTV_FID_V_STA1 | | | | |
| R-0h | | R/W-1h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DTV_FID_V_STA1 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | DTV_FID_F_STA0 | DTV_FID_V_STA0 | | | | |
| R-0h | | R/W-0h | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DTV_FID_V_STA0 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-417. SD_VENC_dtvs4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | Reserved | R | 0h | |
| 29 | DTV_FID_F_STA1 | R/W | 1h | DTV FID output start field for fid=1. |
| 28-16 | DTV_FID_V_STA1 | R/W | 0h | DTV FID output start line for fid=1. |
| 15-14 | Reserved | R | 0h | |
| 13 | DTV_FID_F_STA0 | R/W | 0h | DTV FID output start field for fid=0. Optionally works as force FID toggle enable in progressive or non-interlace mode. |
| 12-0 | DTV_FID_V_STA0 | R/W | 0h | DTV FID output start line for fid=0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.11 SD_VENC_dtvs5 Register (offset = 28h) [reset = 0h]

SD_VENC_dtvs5 is shown in Figure 12-501 and described in Table 12-418.

DTV Sync Timing 5

**Figure 12-501. SD_VENC_dtvs5 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | DTV_AVID_H_STP | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DTV_AVID_H_STP | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | DTV_AVID_H_STA | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DTV_AVID_H_STA | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-418. SD_VENC_dtvs5 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | DTV_AVID_H_STP | R/W | 0h | DTV AVID output stop pixel. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | DTV_AVID_H_STA | R/W | 0h | DTV AVID output start pixel. |



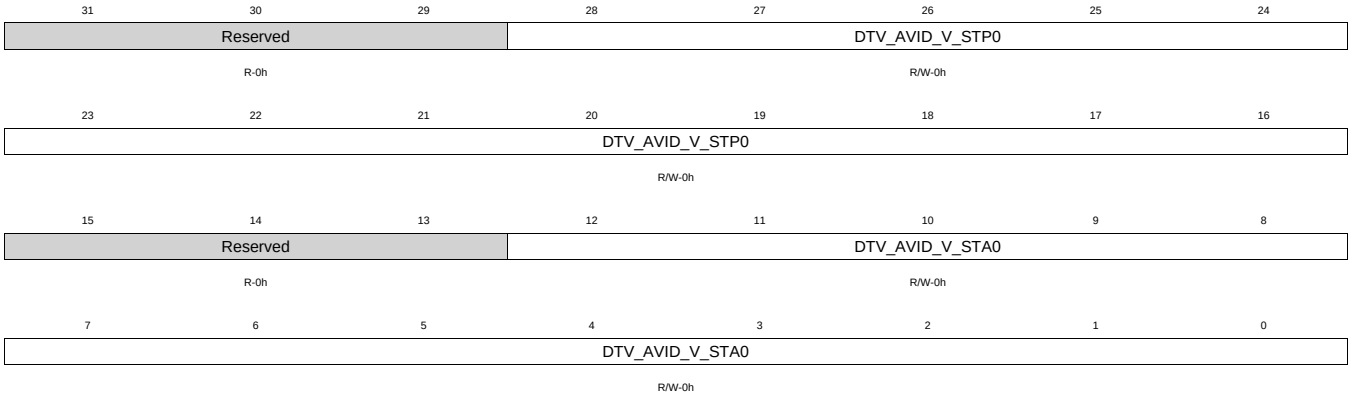
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.12 SD_VENC_dtvs6 Register (offset = 2Ch) [reset = 0h]

SD_VENC_dtvs6 is shown in Figure 12-502 and described in Table 12-419.

DTV Sync Timing 6

**Figure 12-502. SD_VENC_dtvs6 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-419. SD_VENC_dtvs6 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | DTV_AVID_V_STP0 | R/W | 0h | DTV AVID output stop line for fid=0. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | DTV_AVID_V_STA0 | R/W | 0h | DTV AVID output start line for fid=0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.13 SD_VENC_dtvs7 Register (offset = 30h) [reset = 0h]

SD_VENC_dtvs7 is shown in Figure 12-503 and described in Table 12-420.

DTV Sync Timing 7

**Figure 12-503. SD_VENC_dtvs7 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | DTV_AVID_V_STP1 | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DTV_AVID_V_STP1 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | DTV_AVID_V_STA1 | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DTV_AVID_V_STA1 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-420. SD_VENC_dtvs7 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | DTV_AVID_V_STP1 | R/W | 0h | DTV AVID output stop line for fid=1. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | DTV_AVID_V_STA1 | R/W | 0h | DTV AVID output start line for fid=1. |



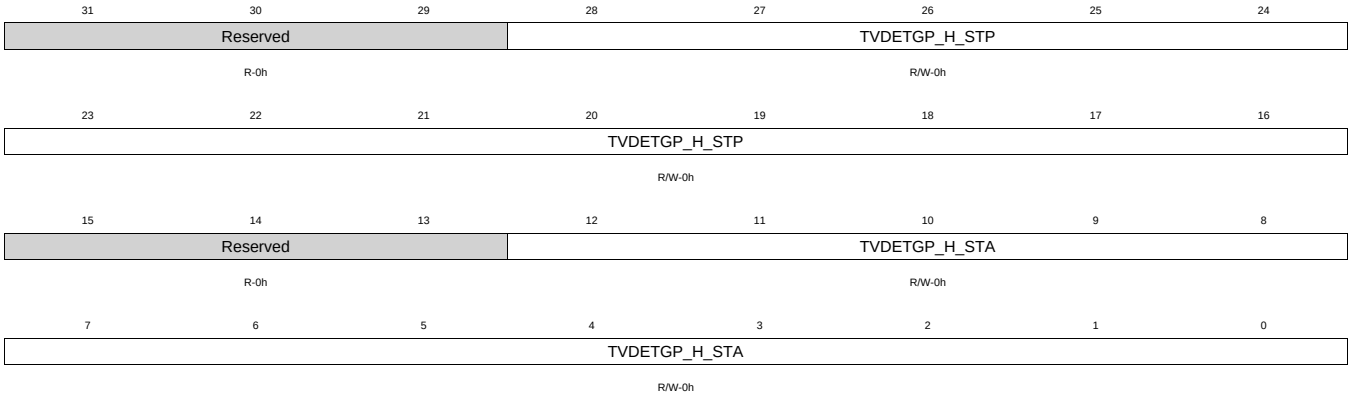
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.14 SD_VENC_tvdetgp0 Register (offset = 54h) [reset = 0h]

SD_VENC_tvdetgp0 is shown in Figure 12-504 and described in Table 12-421.

TVDETGP Timing 0

**Figure 12-504. SD_VENC_tvdetgp0 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-421. SD_VENC_tvdetgp0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | TVDETGP_H_STP | R/W | 0h | TVDETGP output stop pixel. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | TVDETGP_H_STA | R/W | 0h | TVDETGP output start pixel. |

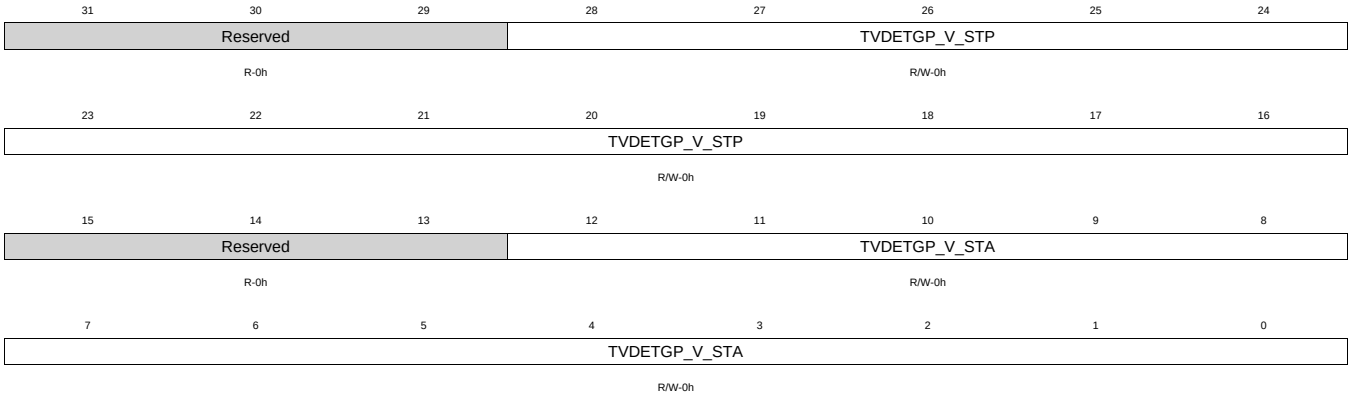

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.15 SD_VENC_tvdetgp1 Register (offset = 58h) [reset = 0h]

SD_VENC_tvdetgp1 is shown in Figure 12-505 and described in Table 12-422.

TVDETGP Timing 1

**Figure 12-505. SD_VENC_tvdetgp1 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-422. SD_VENC_tvdetgp1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | TVDETGP_V_STP | R/W | 0h | TVDETGP output stop line. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | TVDETGP_V_STA | R/W | 0h | TVDETGP output start line. |



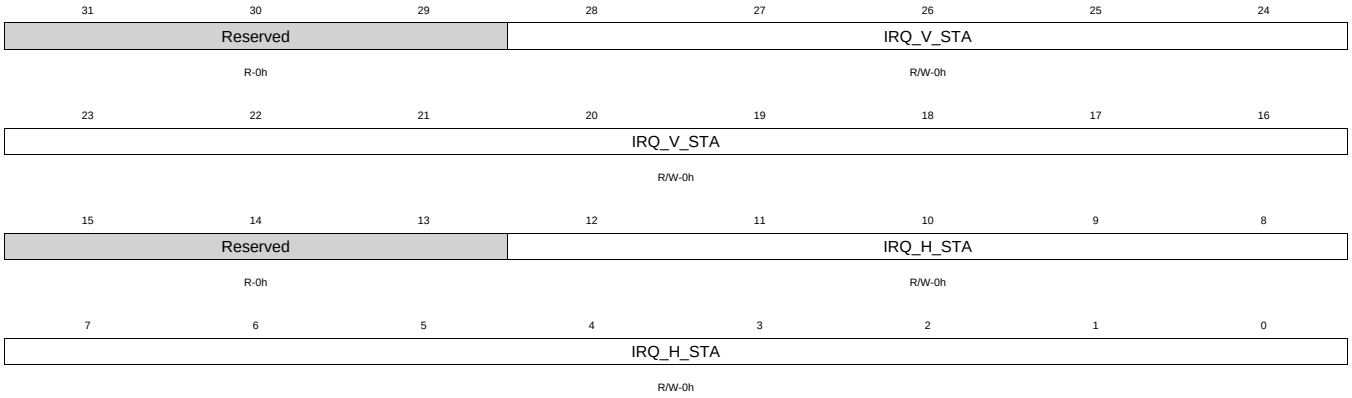
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.16 SD_VENC_irq0 Register (offset = 5Ch) [reset = 0h]

SD_VENC_irq0 is shown in Figure 12-506 and described in Table 12-423.

IRQ Timing

**Figure 12-506. SD_VENC_irq0 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-423. SD_VENC_irq0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | IRQ_V_STA | R/W | 0h | IRQ output start line |
| 15-13 | Reserved | R | 0h | |
| 12-0 | IRQ_H_STA | R/W | 0h | IRQ output start pixel. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.17 SD_VENC_estat Register (offset = 80h) [reset = 0h]

SD_VENC_estat is shown in Figure 12-507 and described in Table 12-424.

Encoder Status

**Figure 12-507. SD_VENC_estat Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | FIDST | Reserved | | CAEST | CAOST |
| R-0h | | | R-0h | R-0h | | R-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-424. SD_VENC_estat Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | Reserved | R | 0h | |
| 4 | FIDST | R | 0h | Field ID monitor. |
| 3-2 | Reserved | R | 0h | |
| 1 | CAEST | R | 0h | Closed caption status (even field). This bit shows 0 when caption data register (L21DE) is ready to be input.. and changes to 1 when data is written to L21DE. This bit is automatically cleared to 0 when a caption data transmission is completed on the line 284(NTSC) or 335(PAL) in even field. |
| 0 | CAOST | R | 0h | Closed caption status (odd field). This bit shows 0 when caption data register (L21DO) is ready to be input.. and changes to 1 when data is written to L21DO. This bit is automatically cleared to 0 when a caption data transmission is completed on the line 21(NTSC) or 22(PAL) in odd field. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.18 SD_VENC_ectl Register (offset = 84h) [reset = 0h]

SD_VENC_ectl is shown in Figure 12-508 and described in Table 12-425.

Encoder Control

**Figure 12-508. SD_VENC_ectl Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | FMT | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | PXLR | VBIEN | BLS | SVSW | DUPS | UPS | GAM |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-425. SD_VENC_ectl Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | Reserved | R | 0h | |
| 11-8 | FMT | R/W | 0h | TV scan format select. 0: 525i 1: 625i 2-15: Reserved |
| 7 | Reserved | R | 0h | |
| 6 | PXLR | R/W | 0h | Pixel rate. Set 0 when the pixel rate is half of the VENC clock. It is used to determine the internal pipeline delay alignment. 0: 1x 1: 2x |
| 5 | VBIEN | R/W | 0h | VBI enable. 0: Off 1: On |
| 4 | BLS | R/W | 0h | Blanking shape disable. When 1.. blanking shaping feature is disabled. 0: Enable 1: Disable |
| 3 | SVSW | R/W | 0h | SD vertical sync width 0: 3H 1: 2.5H |
| 2 | DUPS | R/W | 0h | DAC 2x oversampling enable. 0: Off 1: On |
| 1 | UPS | R/W | 0h | 2x up-sampling enable. 0: Off 1: On |
| 0 | GAM | R/W | 0h | Gamma correction 0: Off 1: On |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.19 SD_VENC_etmg0 Register (offset = 88h) [reset = 069400F4h]

SD_VENC_etmg0 is shown in Figure 12-509 and described in Table 12-426.

Encoder Timing 0

**Figure 12-509. SD_VENC_etmg0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | AV_H_STP | | | | |
| R-0h | | | R/W-694h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| AV_H_STP | | | | | | | |
| R/W-694h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | AV_H_STA | | | | |
| R-0h | | | R/W-F4h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AV_H_STA | | | | | | | |
| R/W-F4h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-426. SD_VENC_etmg0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | AV_H_STP | R/W | 694h | Active video horizontal stop position. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | AV_H_STA | R/W | F4h | Active video horizontal start position. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.20 SD_VENC_etmg1 Register (offset = 8Ch) [reset = 02070022h]

SD_VENC_etmg1 is shown in Figure 12-510 and described in Table 12-427.

Encoder Timing 1

**Figure 12-510. SD_VENC_etmg1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | AV_V_STP0 | | | | |
| R-0h | | | R/W-207h | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| AV_V_STP0 | | | | | | | |
| R/W-207h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | AV_V_STA0 | | | | |
| R-0h | | | R/W-22h | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| AV_V_STA0 | | | | | | | |
| R/W-22h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-427. SD_VENC_etmg1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | AV_V_STP0 | R/W | 207h | Active video vertical stop position for fid=0. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | AV_V_STA0 | R/W | 22h | Active video vertical start position for fid=0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.21 SD_VENC_etmg2 Register (offset = 90h) [reset = 02070022h]

SD_VENC_etmg2 is shown in Figure 12-511 and described in Table 12-428.

Encoder Timing 2

**Figure 12-511. SD_VENC_etmg2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | AV_V_STP1 | | | | |
| R-0h | | | R/W-207h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| AV_V_STP1 | | | | | | | |
| R/W-207h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | AV_V_STA1 | | | | |
| R-0h | | | R/W-22h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AV_V_STA1 | | | | | | | |
| R/W-22h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-428. SD_VENC_etmg2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | AV_V_STP1 | R/W | 207h | Active video vertical stop position for fid=1. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | AV_V_STA1 | R/W | 22h | Active video vertical start position for fid=1. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.22 SD_VENC_etmg3 Register (offset = 94h) [reset = 00D3008Fh]

SD_VENC_etmg3 is shown in Figure 12-512 and described in Table 12-429.

Encoder Timing 3

**Figure 12-512. SD_VENC_etmg3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | BST_H_STP |
| R-0h | | | | | | | R/W-D3h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| BST_H_STP | | | | | | | |
| R/W-D3h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | BST_H_STA |
| R-0h | | | | | | | R/W-8Fh |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BST_H_STA | | | | | | | |
| R/W-8Fh | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-429. SD_VENC_etmg3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | Reserved | R | 0h | |
| 24-16 | BST_H_STP | R/W | D3h | Color burst stop position. |
| 15-9 | Reserved | R | 0h | |
| 8-0 | BST_H_STA | R/W | 8Fh | Color burst start position. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.23 SD_VENC_etmg4 Register (offset = 98h) [reset = 0h]

SD_VENC_etmg4 is shown in Figure 12-513 and described in Table 12-430.

Encoder Timing 4

**Figure 12-513. SD_VENC_etmg4 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | VBI_H_STP | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| VBI_H_STP | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | VBI_H_STA | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VBI_H_STA | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-430. SD_VENC_etmg4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | VBI_H_STP | R/W | 0h | VBI request stop position. |
| 15-13 | Reserved | R | 0h | |
| 12-0 | VBI_H_STA | R/W | 0h | VBI request start position. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.24 SD_VENC_cvbs0 Register (offset = 9Ch) [reset = 034400BCh]

SD_VENC_cvbs0 is shown in Figure 12-514 and described in Table 12-431.

CVBS Control 0

**Figure 12-514. SD_VENC_cvbs0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CSLVL | | | |
| R-0h | | | | R/W-344h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CSLVL | | | | | | | |
| R/W-344h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CTLVL | | | |
| R-0h | | | | R/W-BCh | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CTLVL | | | | | | | |
| R/W-BCh | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-431. SD_VENC_cvbs0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | |
| 27-16 | CSLVL | R/W | 344h | CVBS sync amplitude. |
| 15-12 | Reserved | R | 0h | |
| 11-0 | CTLVL | R/W | BCh | CVBS sync-tip amplitude. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.25 SD_VENC_cvbs1 Register (offset = A0h) [reset = 01A20000h]

SD_VENC_cvbs1 is shown in Figure 12-515 and described in Table 12-432.

CVBS Control 1

**Figure 12-515. SD_VENC_cvbs1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CBLVL | | | |
| R-0h | | | | R/W-1A2h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CBLVL | | | | | | | |
| R/W-1A2h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CYDLY | | | |
| R-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | CCM | | Reserved | CLPF | YLPF | CPSR |
| R-0h | | R/W-0h | | R-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-432. SD_VENC_cvbs1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | |
| 27-16 | CBLVL | R/W | 1A2h | CVBS burst amplitude. |
| 15-12 | Reserved | R | 0h | |
| 11-8 | CYDLY | R/W | 0h | CVBS Y delay adjustment. s3.0 |
| 7-6 | Reserved | R | 0h | |
| 5-4 | CCM | R/W | 0h | CVBS color modulation mode 0: NTSC 1: PAL 2: SECAM 3: Reserved |
| 3 | Reserved | R | 0h | |
| 2 | CLPF | R/W | 0h | CVBS chroma LPF enable 0: Off 1: On |
| 1 | YLPF | R/W | 0h | CVBS luma LPF enable 0: Off 1: On |
| 0 | CPSR | R/W | 0h | CVBS picture sync ratio 0: 10:4 1: 7:3 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.26 SD_VENC_ccsc0 Register (offset = ACh) [reset = 0037011Dh]

SD_VENC_ccsc0 is shown in Figure 12-516 and described in Table 12-433.

CVBS Color Space Conversion 0

**Figure 12-516. SD_VENC_ccsc0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CCSCB0 | | | | |
| R-0h | | | R/W-37h | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CCSCB0 | | | | | | | |
| R/W-37h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CCSCA0 | | | | |
| R-0h | | | R/W-11Dh | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CCSCA0 | | | | | | | |
| R/W-11Dh | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-433. SD_VENC_ccsc0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | CCSCB0 | R/W | 37h | Coefficients of color space converter for CVBS. s6.6 |
| 15-13 | Reserved | R | 0h | |
| 12-0 | CCSCA0 | R/W | 11Dh | Coefficients of color space converter for CVBS. s6.6 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.27 SD_VENC_ccsc1 Register (offset = B0h) [reset = 00000091h]

SD_VENC_ccsc1 is shown in Figure 12-517 and described in Table 12-434.

CVBS Color Space Conversion 1

**Figure 12-517. SD_VENC_ccsc1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CCSCD0 | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CCSCD0 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | CCSCC0 | | | | |
| R-0h | | | R/W-91h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCSCC0 | | | | | | | |
| R/W-91h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-434. SD_VENC_ccsc1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | CCSCD0 | R/W | 0h | Coefficients of color space converter for CVBS. s8.4 |
| 15-13 | Reserved | R | 0h | |
| 12-0 | CCSCC0 | R/W | 91h | Coefficients of color space converter for CVBS. s6.6 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.28 SD_VENC_ccsc2 Register (offset = B4h) [reset = 0000009Dh]

SD_VENC_ccsc2 is shown in Figure 12-518 and described in Table 12-435.

CVBS Color Space Conversion 2

**Figure 12-518. SD_VENC_ccsc2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CCSCE0 | | | | |
| R-0h | | | R/W-9Dh | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CCSCE0 | | | | | | | |
| R/W-9Dh | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-435. SD_VENC_ccsc2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | Reserved | R | 0h | |
| 12-0 | CCSCE0 | R/W | 9Dh | Coefficients of color space converter for CVBS. s12.0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.29 SD_VENC_ccsc3 Register (offset = B8h) [reset = 00D31F74h]

SD_VENC_ccsc3 is shown in Figure 12-519 and described in Table 12-436.

CVBS Color Space Conversion 3

**Figure 12-519. SD_VENC_ccsc3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CCSCB1 | | | | |
| R-0h | | | R/W-D3h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CCSCB1 | | | | | | | |
| R/W-D3h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | CCSCA1 | | | | |
| R-7h | | | R/W-1F74h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCSCA1 | | | | | | | |
| R/W-1F74h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-436. SD_VENC_ccsc3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | CCSCB1 | R/W | D3h | Coefficients of color space converter for CVBS. s6.6 |
| 15-13 | Reserved | R | 7h | |
| 12-0 | CCSCA1 | R/W | 1F74h | Coefficients of color space converter for CVBS. s6.6 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.30 SD_VENC_ccsc4 Register (offset = BCh) [reset = 00001FB9h]

SD_VENC_ccsc4 is shown in Figure 12-520 and described in Table 12-437.

CVBS Color Space Conversion 4

**Figure 12-520. SD_VENC_ccsc4 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CCSCD1 | | | | |
| R-0h | | | R/W-0h | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CCSCD1 | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CCSCC1 | | | | |
| R-0h | | | R/W-1FB9h | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CCSCC1 | | | | | | | |
| R/W-1FB9h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-437. SD_VENC_ccsc4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | CCSCD1 | R/W | 0h | Coefficients of color space converter for CVBS. s8.4 |
| 15-13 | Reserved | R | 0h | |
| 12-0 | CCSCC1 | R/W | 1FB9h | Coefficients of color space converter for CVBS. s6.6 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.31 SD_VENC_ccsc5 Register (offset = C0h) [reset = 0h]

SD_VENC_ccsc5 is shown in Figure 12-521 and described in Table 12-438.

CVBS Color Space Conversion 5

**Figure 12-521. SD_VENC_ccsc5 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CCSCE1 | | | | |
| R-0h | | | R/W-0h | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CCSCE1 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-438. SD_VENC_ccsc5 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | Reserved | R | 0h | |
| 12-0 | CCSCE1 | R/W | 0h | Coefficients of color space converter for CVBS. s12.0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.32 SD_VENC_ccsc6 Register (offset = C4h) [reset = 1FD01F06h]

SD_VENC_ccsc6 is shown in Figure 12-522 and described in Table 12-439.

CVBS Color Space Conversion 6

**Figure 12-522. SD_VENC_ccsc6 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CCSCB2 | | | | |
| R-0h | | | R/W-1FD0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CCSCB2 | | | | | | | |
| R/W-1FD0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | CCSCA2 | | | | |
| R-0h | | | R/W-1F06h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCSCA2 | | | | | | | |
| R/W-1F06h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-439. SD_VENC_ccsc6 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | CCSCB2 | R/W | 1FD0h | Coefficients of color space converter for CVBS. s6.6 |
| 15-13 | Reserved | R | 0h | |
| 12-0 | CCSCA2 | R/W | 1F06h | Coefficients of color space converter for CVBS. s6.6 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.33 SD_VENC_ccsc7 Register (offset = C8h) [reset = 0000012Ah]

SD_VENC_ccsc7 is shown in Figure 12-523 and described in Table 12-440.

CVBS Color Space Conversion 7

**Figure 12-523. SD_VENC_ccsc7 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CCSCD2 | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CCSCD2 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | CCSCC2 | | | | |
| R-0h | | | R/W-12Ah | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCSCC2 | | | | | | | |
| R/W-12Ah | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-440. SD_VENC_ccsc7 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | CCSCD2 | R/W | 0h | Coefficients of color space converter for CVBS. s8.4 |
| 15-13 | Reserved | R | 0h | |
| 12-0 | CCSCC2 | R/W | 12Ah | Coefficients of color space converter for CVBS. s6.6 |



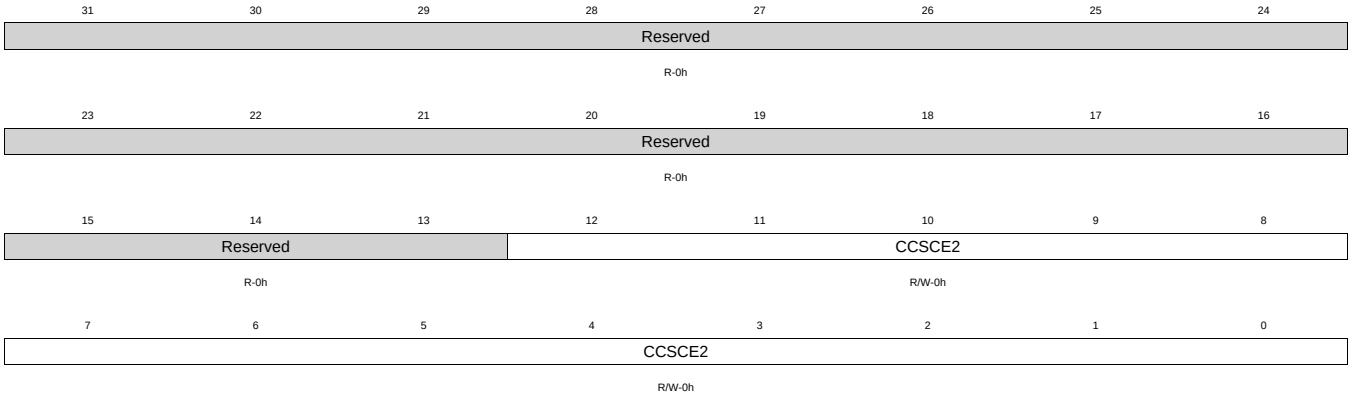
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.34 SD_VENC_ccsc8 Register (offset = CCh) [reset = 0h]

SD_VENC_ccsc8 is shown in Figure 12-524 and described in Table 12-441.

CVBS Color Space Conversion 8

**Figure 12-524. SD_VENC_ccsc8 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-441. SD_VENC_ccsc8 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | Reserved | R | 0h | |
| 12-0 | CCSCE2 | R/W | 0h | Coefficients of color space converter for CVBS. s12.0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.35 SD_VENC_cygclp Register (offset = F4h) [reset = 00000FFFh]

SD_VENC_cygclp is shown in Figure 12-525 and described in Table 12-442.

CVBS Y Clip

**Figure 12-525. SD_VENC_cygclp Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CYLCLP | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CYLCLP | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | CYUCLP | | | | |
| R-0h | | | R/W-FFFh | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CYUCLP | | | | | | | |
| R/W-FFFh | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-442. SD_VENC_cygclp Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | CYLCLP | R/W | 0h | CVBS Y Lower Limit. s12.0 |
| 15-13 | Reserved | R | 0h | |
| 12-0 | CYUCLP | R/W | FFFh | CVBS Y Upper Limit. s12.0 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.36 SD_VENC_cubclp Register (offset = F8h) [reset = 180007FFh]

SD_VENC_cubclp is shown in Figure 12-526 and described in Table 12-443.

CVBS U Clip

**Figure 12-526. SD_VENC_cubclp Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CULCLP | | | | |
| R-0h | | | R/W-1800h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CULCLP | | | | | | | |
| R/W-1800h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | CUUCLP | | | | |
| R-0h | | | R/W-7FFh | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CUUCLP | | | | | | | |
| R/W-7FFh | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-443. SD_VENC_cubclp Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | CULCLP | R/W | 1800h | CVBS U Lower Limit. s12.0 |
| 15-13 | Reserved | R | 0h | |
| 12-0 | CUUCLP | R/W | 7FFh | CVBS U Upper Limit. s12.0 |

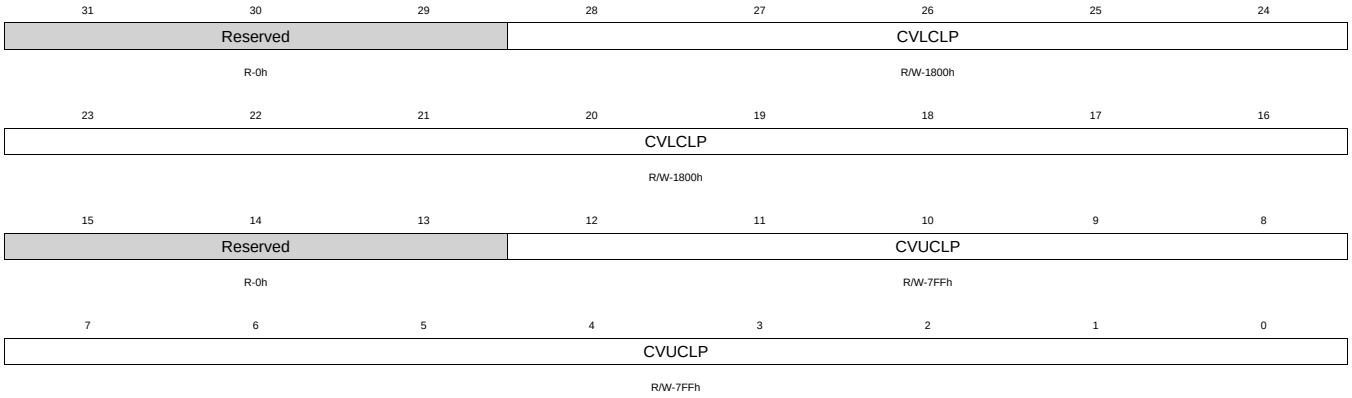

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.37 SD_VENC_cvrclp Register (offset = FCh) [reset = 180007FFh]

SD_VENC_cvrclp is shown in Figure 12-527 and described in Table 12-444.

CVBS V Clip

**Figure 12-527. SD_VENC_cvrclp Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-444. SD_VENC_cvrclp Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28-16 | CVLCLP | R/W | 1800h | CVBS V Lower Limit. s12.0 |
| 15-13 | Reserved | R | 0h | |
| 12-0 | CVUCLP | R/W | 7FFh | CVBS V Vpper Limit. s12.0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.38 SD_VENC_ylpf0 Register (offset = 10Ch) [reset = 0h]

SD_VENC_ylpf0 is shown in Figure 12-528 and described in Table 12-445.

CVBS Luma LPF Coefficient 0

**Figure 12-528. SD_VENC_ylpf0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| YLPFC3 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| YLPFC2 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| YLPFC1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| YLPFC0 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-445. SD_VENC_ylpf0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | YLPFC3 | R/W | 0h | Luma LPF coefficient 3. s0.7. |
| 23-16 | YLPFC2 | R/W | 0h | Luma LPF coefficient 2. s0.7. |
| 15-8 | YLPFC1 | R/W | 0h | Luma LPF coefficient 1. s0.7. |
| 7-0 | YLPFC0 | R/W | 0h | Luma LPF coefficient 0. s0.7. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.39 SD_VENC_ylpf1 Register (offset = 110h) [reset = 00004020h]

SD_VENC_ylpf1 is shown in Figure 12-529 and described in Table 12-446.

CVBS Luma LPF Coefficient 1

**Figure 12-529. SD_VENC_ylpf1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| YLPFC5 | | | | | | | |
| R/W-40h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| YLPFC4 | | | | | | | |
| R/W-20h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-446. SD_VENC_ylpf1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-8 | YLPFC5 | R/W | 40h | Luma LPF coefficient 5. s0.7. |
| 7-0 | YLPFC4 | R/W | 20h | Luma LPF coefficient 4. s0.7. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.40 SD_VENC_clpf0 Register (offset = 114h) [reset = 0C000000h]

SD_VENC_clpf0 is shown in Figure 12-530 and described in Table 12-447.

CVBS Chroma LPF Coefficient 0

**Figure 12-530. SD_VENC_clpf0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CLPFC3 | | | | | | | |
| R/W-Ch | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CLPFC2 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CLPFC1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLPFC0 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-447. SD_VENC_clpf0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | CLPFC3 | R/W | Ch | Chroma LPF coefficient 3. s0.7. |
| 23-16 | CLPFC2 | R/W | 0h | Chroma LPF coefficient 2. s0.7. |
| 15-8 | CLPFC1 | R/W | 0h | Chroma LPF coefficient 1. s0.7. |
| 7-0 | CLPFC0 | R/W | 0h | Chroma LPF coefficient 0. s0.7. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.41 SD_VENC_clpf1 Register (offset = 118h) [reset = 00002820h]

SD_VENC_clpf1 is shown in Figure 12-531 and described in Table 12-448.

CVBS Chroma LPF Coefficient 1

**Figure 12-531. SD_VENC_clpf1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CLPFC5 | | | | | | | |
| R/W-28h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CLPFC4 | | | | | | | |
| R/W-20h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-448. SD_VENC_clpf1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-8 | CLPFC5 | R/W | 28h | Chroma LPF coefficient 5. s0.7. |
| 7-0 | CLPFC4 | R/W | 20h | Chroma LPF coefficient 4. s0.7. |

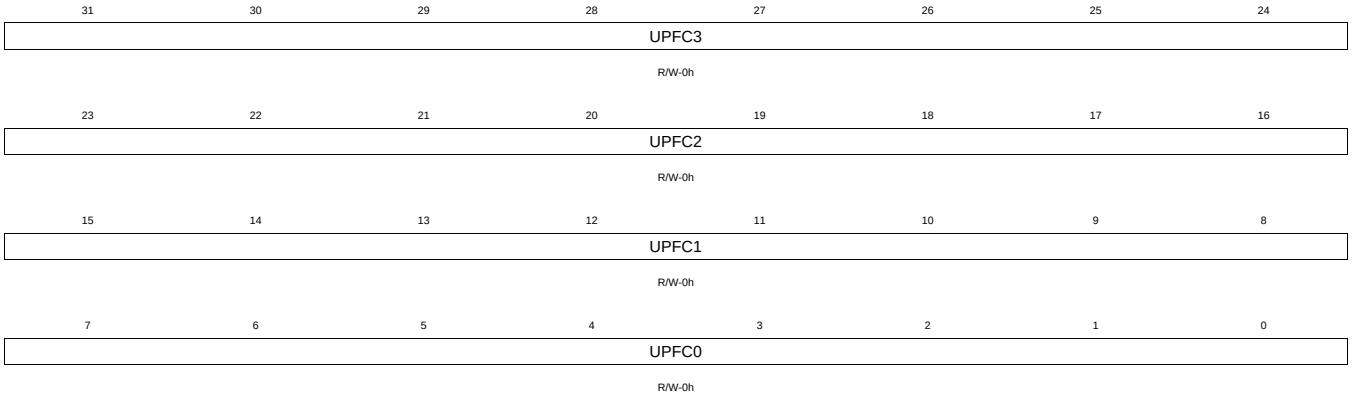

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.42 SD_VENC_upf0 Register (offset = 11Ch) [reset = 0h]

SD_VENC_upf0 is shown in Figure 12-532 and described in Table 12-449.

2x Upsampling Coefficient 0

**Figure 12-532. SD_VENC_upf0 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-449. SD_VENC_upf0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | UPFC3 | R/W | 0h | 2x up-sampling filter coefficient 3. s0.7. |
| 23-16 | UPFC2 | R/W | 0h | 2x up-sampling filter coefficient 2. s0.7. |
| 15-8 | UPFC1 | R/W | 0h | 2x up-sampling filter coefficient 1. s0.7. |
| 7-0 | UPFC0 | R/W | 0h | 2x up-sampling filter coefficient 0. s0.7. |

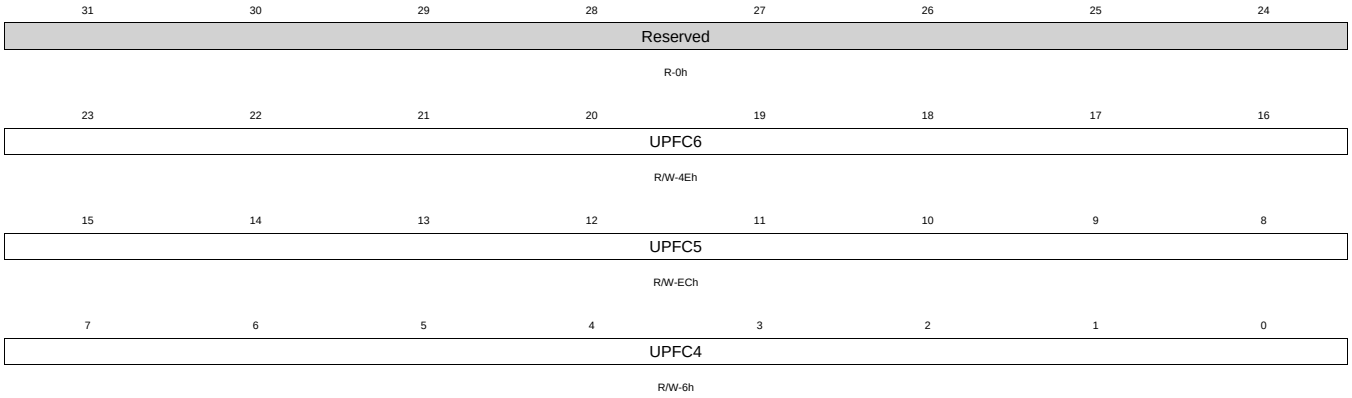

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.43 SD_VENC_upf1 Register (offset = 120h) [reset = 004EEC06h]

SD_VENC_upf1 is shown in Figure 12-533 and described in Table 12-450.

2x Upsampling Coefficient 1

**Figure 12-533. SD_VENC_upf1 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-450. SD_VENC_upf1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-16 | UPFC6 | R/W | 4Eh | 2x up-sampling filter coefficient 6. s0.7. |
| 15-8 | UPFC5 | R/W | ECh | 2x up-sampling filter coefficient 5. s0.7. |
| 7-0 | UPFC4 | R/W | 6h | 2x up-sampling filter coefficient 4. s0.7. |

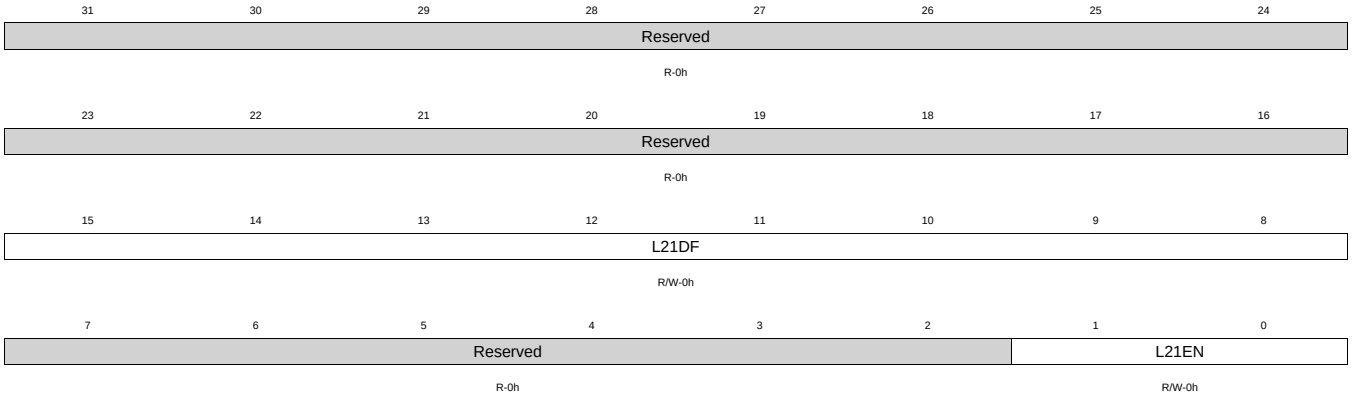

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.44 SD_VENC_l21ctl Register (offset = 124h) [reset = 0h]

SD_VENC_l21ctl is shown in Figure 12-534 and described in Table 12-451.

Line 21 Control

**Figure 12-534. SD_VENC_l21ctl Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-451. SD_VENC_l21ctl Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-8 | L21DF | R/W | 0h | Closed caption default data register. When the caption data register (L21DO or L21DE) is not updated before the caption data transmission timing for the corresponding field.. the ASCII code specified by this register is automatically transmitted for closed caption data. |
| 7-2 | Reserved | R | 0h | |
| 1-0 | L21EN | R/W | 0h | Closed caption field select. Specify the fields on which closed captioning is enabled. Closed caption data is transmitted on the line 21 for odd field and the line 284 for even field (line 22 and 335 for PAL). 0: No data output 1: Odd field 2: Even field 3: Both fields |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.45 SD_VENC_l21do Register (offset = 128h) [reset = 0h]

SD_VENC_l21do is shown in Figure 12-535 and described in Table 12-452.

Line 21 Data Even Field

**Figure 12-535. SD_VENC_l21do Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| L21DO0 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| L21DO1 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-452. SD_VENC_l21do Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-8 | L21DO0 | R/W | 0h | Closed caption data0 (odd field). Specify the ASCII code of the 1st byte to be transmitted in closed captioning for odd field. |
| 7-0 | L21DO1 | R/W | 0h | Closed caption data1 (odd field). Specify the ASCII code of the 2nd byte to be transmitted in closed captioning for odd field. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.46 SD_VENC_l21de Register (offset = 12Ch) [reset = 0h]

SD_VENC_l21de is shown in Figure 12-536 and described in Table 12-453.

Line 21 Data Odd Field

**Figure 12-536. SD_VENC_l21de Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| L21DE0 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| L21DE1 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-453. SD_VENC_l21de Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | Reserved | R | 0h | |
| 15-8 | L21DE0 | R/W | 0h | Closed caption data0 (even field). Specify the ASCII code of the 1st byte to be transmitted in closed captioning for even field. |
| 7-0 | L21DE1 | R/W | 0h | Closed caption data1 (even field). Specify the ASCII code of the 2nd byte to be transmitted in closed captioning for even field. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.47 SD_VENC_wss Register (offset = 130h) [reset = 0h]

SD_VENC_wss is shown in Figure 12-537 and described in Table 12-454.

WSS Control

**Figure 12-537. SD_VENC_wss Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | WSS_EN | Reserved | | | |
| R-0h | | | R/W-0h | R-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | WSS_DATA | | | |
| R-0h | | | | R/W-0h | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| WSS_DATA | | | | | | | |
| R/W-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WSS_DATA | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-454. SD_VENC_wss Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | Reserved | R | 0h | |
| 28 | WSS_EN | R/W | 0h | CGMS/WSS insertion enable 0: Disable 1: Enable |
| 27-20 | Reserved | R | 0h | |
| 19-0 | WSS_DATA | R/W | 0h | WSS data register (525i) bit1-0: WORD0 bit5-2: WORD1 bit13-6:WORD2 bit19-14: CRC (625i) bit3-0: GROUP1 bit7-4: GROUP2 bit10-8: GROUP3 bit13-11: GROUP4 bit19-14: unused |



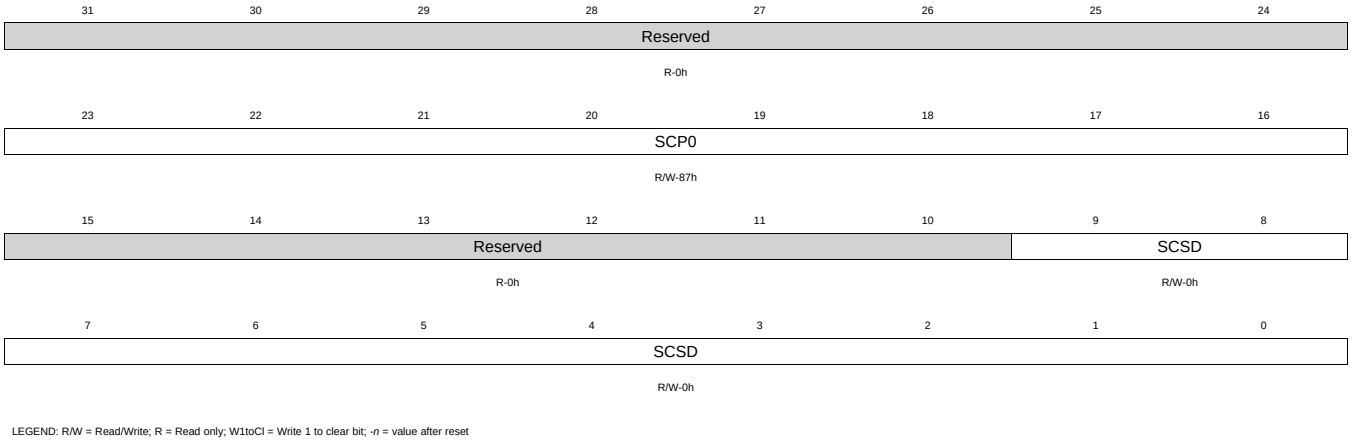
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.48 SD_VENC_scctl0 Register (offset = 148h) [reset = 00870000h]

SD_VENC_scctl0 is shown in Figure 12-538 and described in Table 12-455.

Sub-carrier Frequency Control 0

**Figure 12-538. SD_VENC_scctl0 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-455. SD_VENC_scctl0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-16 | SCP0 | R/W | 87h | Sub-carrier frequency parameter 0. Default is 135. |
| 15-10 | Reserved | R | 0h | |
| 9-0 | SCSD | R/W | 0h | Sub-carrier initial phase value. The degree can be specified by SCSD/1024*360. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.49 SD_VENC_scctl1 Register (offset = 14Ch) [reset = 00210019h]

SD_VENC_scctl1 is shown in Figure 12-539 and described in Table 12-456.

Sub-carrier Frequency Control 1

**Figure 12-539. SD_VENC_scctl1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SCP2 | | | | | | | |
| R/W-21h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SCP2 | | | | | | | |
| R/W-21h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SCP1 | | | | | | | |
| R/W-19h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SCP1 | | | | | | | |
| R/W-19h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-456. SD_VENC_scctl1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | SCP2 | R/W | 21h | Sub-carrier frequency parameter 2. Default is 33. Fsc = (SCP0 + SCP1/SCP2) * Fin / 1024 |
| 15-0 | SCP1 | R/W | 19h | Sub-carrier frequency parameter 1. Default is 25. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.50 SD_VENC_dacsel Register (offset = 150h) [reset = 0h]

SD_VENC_dacsel is shown in Figure 12-540 and described in Table 12-457.

DAC Output Signal Select

**Figure 12-540. SD_VENC_dacsel Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | DA1E | DA0E |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DA1S | | | | DA0S | | | |
| R/W-0h | | | | R/W-0h | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-457. SD_VENC_dacsel Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | Reserved | R | 0h | |
| 25 | DA1E | R/W | 0h | DAC1 enable |
| 24 | DA0E | R/W | 0h | DAC0 enable 0: DAC power down 1: DAC enable |
| 23-8 | Reserved | R | 0h | |
| 7-4 | DA1S | R/W | 0h | DAC1 output select. See DAC0S. |
| 3-0 | DA0S | R/W | 0h | DAC0 output select. 0: CVBS 1: S-Video Y 2: S-Video; 3-15: Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.51 SD_VENC_dupf0 Register (offset = 154h) [reset = 0h]

SD_VENC_dupf0 is shown in Figure 12-541 and described in Table 12-458.

DAC 2x Upsampling Coefficient 0

**Figure 12-541. SD_VENC_dupf0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DUPFC3 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DUPFC2 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DUPFC1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DUPFC0 | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-458. SD_VENC_dupf0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | DUPFC3 | R/W | 0h | DAC 2x oversampling filter coefficient 3. s0.7. (Default=0) |
| 23-16 | DUPFC2 | R/W | 0h | DAC 2x oversampling filter coefficient 2. s0.7. (Default=0) |
| 15-8 | DUPFC1 | R/W | 0h | DAC 2x oversampling filter coefficient 1. s0.7. (Default=0) |
| 7-0 | DUPFC0 | R/W | 0h | DAC 2x oversampling filter coefficient 0. s0.7. (Default=0) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.52 SD_VENC_dupf1 Register (offset = 158h) [reset = 004EEC06h]

SD_VENC_dupf1 is shown in Figure 12-542 and described in Table 12-459.

DAC 2x Upsampling Coefficient 1

**Figure 12-542. SD_VENC_dupf1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DUPFC6 | | | | | | | |
| R/W-4Eh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DUPFC5 | | | | | | | |
| R/W-ECh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DUPFC4 | | | | | | | |
| R/W-6h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-459. SD_VENC_dupf1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | Reserved | R | 0h | |
| 23-16 | DUPFC6 | R/W | 4Eh | DAC 2x oversampling filter coefficient 6. s0.7. (Default=78) |
| 15-8 | DUPFC5 | R/W | ECh | DAC 2x oversampling filter coefficient 5. s0.7. (Default=-20) |
| 7-0 | DUPFC4 | R/W | 6h | DAC 2x oversampling filter coefficient 4. s0.7. (Default=6) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.53 SD_VENC_dactst Register (offset = 15Ch) [reset = 0h]

SD_VENC_dactst is shown in Figure 12-543 and described in Table 12-460.

DAC Test Mode

**Figure 12-543. SD_VENC_dactst Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | DAIV | DADC |
| R-0h | | | | | | R/W-0h | R/W-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | DALVL | | | |
| R-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DALVL | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-460. SD_VENC_dactst Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | Reserved | R | 0h | |
| 17 | DAIV | R/W | 0h | DAC output invert mode. Setting 1 inverts the DAC output code. 0: Non-Inverse 1: Inverse |
| 16 | DADC | R/W | 0h | DAC DC output mode. Setting 1 converts the value written in the DALVL register to DAC and directly outputs from DAOUT. 0: Normal 1: DC output mode |
| 15-12 | Reserved | R | 0h | |
| 11-0 | DALVL | R/W | 0h | DAC DC level control |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.12.54 SD_VENC_vtest Register (offset = 1FCh) [reset = 0h]

SD_VENC_vtest is shown in Figure 12-544 and described in Table 12-461.

Internal Test Register

**Figure 12-544. SD_VENC_vtest Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VTST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-461. SD_VENC_vtest Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | VTST | R/W | 0h | Reserved for internal testing. Should be always zero for normal usage. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13 VCOMP Registers

Table 12-462 lists the memory-mapped registers for the VCOMP. All register offset addresses not listed in Table 12-462 should be considered as reserved locations and the register contents should not be modified.

**Table 12-462. VCOMP REGISTERS**

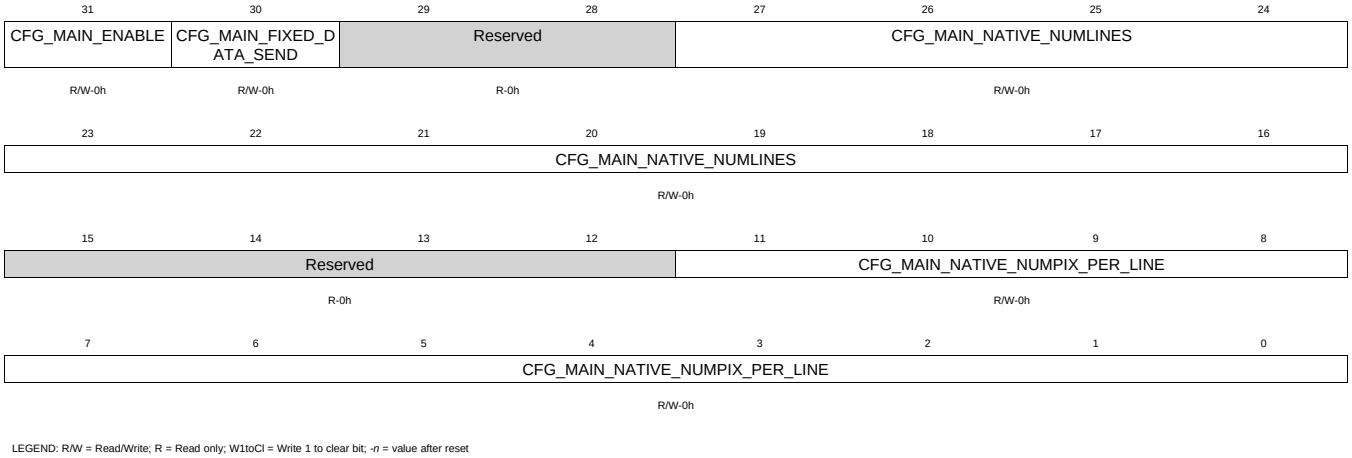
| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | VCOMP_reg0 | VCOMP Main Register 1 | Section 12.3.13.1 |
| 4h | VCOMP_reg1 | VCOMP Main Register 2 | Section 12.3.13.2 |
| 8h | VCOMP_reg2 | VCOMP Main Register 3 | Section 12.3.13.3 |
| Ch | VCOMP_reg3 | VCOMP Aux Register 1 | Section 12.3.13.4 |
| 10h | VCOMP_reg4 | VCOMP Aux Register 2 | Section 12.3.13.5 |
| 14h | VCOMP_reg5 | VCOMP Aux Register 3 | Section 12.3.13.6 |
| 18h | VCOMP_reg6 | VCOMP Display Register 1 | Section 12.3.13.7 |
| 1Ch | VCOMP_reg7 | VCOMP Display Register 2 | Section 12.3.13.8 |
| 20h | VCOMP_reg8 | VCOMP Display Register 3 | Section 12.3.13.9 |
| 24h | VCOMP_reg9 | VCOMP Display Register 4 | Section 12.3.13.10 |
| 28h | VCOMP_reg10 | VCOMP Display Register 5 | Section 12.3.13.11 |
| 2Ch | VCOMP_reg11 | VCOMP Display Register 6 | Section 12.3.13.12 |
| 30h | VCOMP_reg12 | VCOMP Display Register 7 | Section 12.3.13.13 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.1 VCOMP_reg0 Register (offset = 0h) [reset = 0h]

VCOMP_reg0 is shown in Figure 12-545 and described in Table 12-463.

**Figure 12-545. VCOMP_reg0 Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-463. VCOMP_reg0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CFG_MAIN_ENABLE | R/W | 0h | 0 = Disable Main Plane<br>1 = Enable Main Plane |
| 30 | CFG_MAIN_FIXED_DATA_SEND | R/W | 0h | 0 = Send out the main source data<br>1 = When the main source is enabled.. send the alternate main Y/Cb/Cr values instead of the true source picture. This bit allows the datapath to flush through without outputing the actual picture. The main window would be seen as a constant color. |
| 29-28 | Reserved | R | 0h | Reserved |
| 27-16 | CFG_MAIN_NATIVE_NUMLINES | R/W | 0h | Number of lines in a field or frame from the incoming main source |
| 15-12 | Reserved | R | 0h | Reserved |
| 11-0 | CFG_MAIN_NATIVE_NUMPIX_PER_LINE | R/W | 0h | Number of pixels per line from the incoming main source |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.2 VCOMP_reg1 Register (offset = 4h) [reset = 0h]

VCOMP_reg1 is shown in Figure 12-546 and described in Table 12-464.

**Figure 12-546. VCOMP_reg1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | CFG_MAIN_STARTUP_HANDLING | Reserved | | CFG_MAIN_SKIP_NUMPIX | | | |
| R-0h | R/W-0h | R-0h | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CFG_MAIN_SKIP_NUMPIX | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_MAIN_USE_NUMPIX | | | |
| R-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CFG_MAIN_USE_NUMPIX | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-464. VCOMP_reg1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Reserved | R | 0h | Reserved |
| 30 | CFG_MAIN_STARTUP_HANDLING | R/W | 0h | 0 = Default condition for video in/video out operation<br>1 = Set for video in/no video out<br>The Main source must be disabled when this bit is set. The NF and pixel data to the downstream module is not sent. This mode forces the VCOMP to request pictures from the upstream modules. Thus, these modules (like the DEI) can go ahead and load up with one or more input fields/frames before sending actual pictures to the output. Alternately, the config reg can be set to '0' and the fixed_data_send feature can send a color plane to downstream modules. cfg_main_enable should be '0' when this bit is set. If both the Main and the Aux are turned off, no nf or pixel data will propagate downstream of the dss_vcomp. |
| 29-28 | Reserved | R | 0h | Reserved |
| 27-16 | CFG_MAIN_SKIP_NUMPIX | R/W | 0h | Starting from the first pixel of each line. this field determines the number of incoming pixels to discard as part of the Compositor's picture clipping feature.<br>SKIP_NUMPIX should be even - For odd values, U and V will get swapped in output. As a workaround to support odd values, swap format may be used in the VPDMA descriptor to counteract this problem. |
| 15-12 | Reserved | R | 0h | Reserved |
| 11-0 | CFG_MAIN_USE_NUMPIX | R/W | 0h | After main_skip_numpix number of pixels, this field determiines the number of pixels to use from each incoming line. This field allows the input picture to be clipped. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.3 VCOMP_reg2 Register (offset = 8h) [reset = 0h]

VCOMP_reg2 is shown in Figure 12-547 and described in Table 12-465.

**Figure 12-547. VCOMP_reg2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_MAIN_SKIP_NUMLINES | | | |
| R-0h | | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CFG_MAIN_SKIP_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_MAIN_USE_NUMLINES | | | |
| R-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CFG_MAIN_USE_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-465. VCOMP_reg2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | Reserved |
| 27-16 | CFG_MAIN_SKIP_NUMLINES | R/W | 0h | Starting from the first line of each field or frame.. this field determines the number of incoming lines to discard as part of the Compositor's picture clipping feature. |
| 15-12 | Reserved | R | 0h | Reserved |
| 11-0 | CFG_MAIN_USE_NUMLINES | R/W | 0h | After main_skip_numlines number of lines.. this field determines the number of lines to use from each incoming field or frame. This field allows the input picture to be clipped. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.4 VCOMP_reg3 Register (offset = Ch) [reset = 0h]

VCOMP_reg3 is shown in Figure 12-548 and described in Table 12-466.

**Figure 12-548. VCOMP_reg3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CFG_AUX_ENABLE | CFG_AUX_FIXED_DATA_SEND | Reserved | | CFG_AUX_NATIVE_NUMLINES | | | |
| R/W-0h | R/W-0h | R-0h | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CFG_AUX_NATIVE_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | CFG_AUX_NATIVE_NUMPIX_PER_LINE | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFG_AUX_NATIVE_NUMPIX_PER_LINE | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-466. VCOMP_reg3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CFG_AUX_ENABLE | R/W | 0h | 0 = Disable Aux Plane<br>1 = Enable Aux Plane |
| 30 | CFG_AUX_FIXED_DATA_SEND | R/W | 0h | 0 = Send out the aux source data<br>1 = When the aux source is enabled.. send the alternate aux Y/Cb/Cr values instead of the true source picture. This bit allows the datapath to flush through without outputing the actual picture. The aux window would be seen as a constant color. |
| 29-28 | Reserved | R | 0h | Reserved |
| 27-16 | CFG_AUX_NATIVE_NUMLINES | R/W | 0h | Number of lines in a field or frame from the incoming aux source |
| 15-12 | Reserved | R | 0h | Reserved |
| 11-0 | CFG_AUX_NATIVE_NUMPIX_PER_LINE | R/W | 0h | Number of pixels per line from the incoming aux source |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.5 VCOMP_reg4 Register (offset = 10h) [reset = 0h]

VCOMP_reg4 is shown in Figure 12-549 and described in Table 12-467.

**Figure 12-549. VCOMP_reg4 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | CFG_AUX_STARTUP_HANDLING | Reserved | | CFG_AUX_SKIP_NUMPIX | | | |
| R-0h | R/W-0h | R-0h | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CFG_AUX_SKIP_NUMPIX | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | CFG_AUX_USE_NUMPIX | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFG_AUX_USE_NUMPIX | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-467. VCOMP_reg4 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | Reserved | R | 0h | Reserved |
| 30 | CFG_AUX_STARTUP_HANDLING | R/W | 0h | 0 = Default condition for video in/video out operation<br>1 = Set for video in/no video out<br>The Aux source must be disabled when this bit is set. The NF and pixel data to the downstream module is not sent. This mode forces the VCOMP to request pictures from the upstream modules. Thus, these modules (like the DEI) can go ahead and load up with one or more input fields/frames before sending actual pictures to the output. Alternately, the config reg can be set to '0' and the fixed_data_send feature can send a color plane to downstream modules. cfg_aux_enable should be '0' when this bit is set. If both the Main and the Aux are turned off, no nf or pixel data will propagate downstream of the dss_vcomp. |
| 29-28 | Reserved | R | 0h | Reserved |
| 27-16 | CFG_AUX_SKIP_NUMPIX | R/W | 0h | Starting from the first pixel of each line.. this field determines the number of incoming pixels to discard as part of the Compositor's picture clipping feature.<br>SKIP_NUMPIX should be even - For odd values, U and V will get swapped in output. As a workaround to support odd values, swap format may be used in the VPDMA descriptor to counteract this problem. |
| 15-12 | Reserved | R | 0h | Reserved |
| 11-0 | CFG_AUX_USE_NUMPIX | R/W | 0h | After aux_skip_numpix number of pixels, this field determiines the number of pixels to use from each incoming line. This field allows the input picture to be clipped. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.6 VCOMP_reg5 Register (offset = 14h) [reset = 0h]

VCOMP_reg5 is shown in Figure 12-550 and described in Table 12-468.

**Figure 12-550. VCOMP_reg5 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_AUX_SKIP_NUMLINES | | | |
| R-0h | | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CFG_AUX_SKIP_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_AUX_USE_NUMLINES | | | |
| R-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CFG_AUX_USE_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-468. VCOMP_reg5 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | Reserved |
| 27-16 | CFG_AUX_SKIP_NUMLINES | R/W | 0h | Starting from the first line of each field or frame.. this field determines the number of incoming lines to discard as part of the Compositor's picture clipping feature. |
| 15-12 | Reserved | R | 0h | Reserved |
| 11-0 | CFG_AUX_USE_NUMLINES | R/W | 0h | After aux_skip_numlines number of lines.. this field determines the number of lines to use from each incoming field or frame. This field allows the input picture to be clipped. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.7 VCOMP_reg6 Register (offset = 18h) [reset = 0h]

VCOMP_reg6 is shown in Figure 12-551 and described in Table 12-469.

**Figure 12-551. VCOMP_reg6 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_DSPLY_NUMLINES | | | |
| R-0h | | | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CFG_DSPLY_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | CFG_DSPLY_NUMPIX_PER_LINE | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFG_DSPLY_NUMPIX_PER_LINE | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-469. VCOMP_reg6 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | Reserved |
| 27-16 | CFG_DSPLY_NUMLINES | R/W | 0h | Number of lines in a field or frame for the output picture. |
| 15-12 | Reserved | R | 0h | Reserved |
| 11-0 | CFG_DSPLY_NUMPIX_PER_LINE | R/W | 0h | Number of pixels per line for the output picture. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.8 VCOMP_reg7 Register (offset = 1Ch) [reset = 0h]

VCOMP_reg7 is shown in Figure 12-552 and described in Table 12-470.

**Figure 12-552. VCOMP_reg7 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_DSPLY_MAIN_Y_ORIGIN | | | |
| R-0h | | | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CFG_DSPLY_MAIN_Y_ORIGIN | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_DSPLAY_MAIN_X_ORIGIN | | | |
| R-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CFG_DSPLAY_MAIN_X_ORIGIN | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-470. VCOMP_reg7 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | Reserved | R | 0h | Reserved |
| 27-16 | CFG_DSPLY_MAIN_Y_ORIGIN | R/W | 0h | Row origin index for the Main source picture . This source picture is the result of any cropping by cfg_main_skip_numlines. (x..y) = (0..0) is the top left corner of the display. |
| 15-12 | Reserved | R | 0h | Reserved |
| 11-0 | CFG_DSPLAY_MAIN_X_ORIGIN | R/W | 0h | Column origin index for the Main source picture. This source picture is the result of any cropping by cfg_main_skip_numpix. (x..y) = (0..0) is the top left corner of the display. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.9 VCOMP_reg8 Register (offset = 20h) [reset = 0h]

VCOMP_reg8 is shown in Figure 12-553 and described in Table 12-471.

**Figure 12-553. VCOMP_reg8 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CFG_DWNSTRM_SRC_FID_CTRL | | Reserved | | CFG_DSPLY_AUX_Y_ORIGIN | | | |
| R/W-0h | | R-0h | | R/W-0h | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CFG_DSPLY_AUX_Y_ORIGIN | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CFG_DSPLAY_AUX_X_ORIGIN | | | |
| R-0h | | | | R/W-0h | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CFG_DSPLAY_AUX_X_ORIGIN | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-471. VCOMP_reg8 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | CFG_DWNSTRM_SRC_FID_CTRL | R/W | 0h | 00 = Source FID (bit 0) out of the VCOMP is the same as the respective bit from the Upstream Module<br>01 = Source FID (bit 0) out of the VCOMP is the inverse of the respective bit from the Upstream Module<br>10 = Source FID (bit 0) out of the VCOMP is '0'.<br>11 = Source FID (bit 0) out of the VCOMP is '1'.<br>When both the Main and Aux sources are enabled.. NF and the Source/Encoder FID values are taken from the Main or Aux ports as defined in cfg_nf_handling. Otherwise.. NF and the Source/Encoder FID values are taken from the Active port. In no plane is active.. then no NF will be sent downstream. This register setting enables Bit 0.. representing the Source Picture FID.. to be the same as the Upstream Input.. the inverse of the Upstream Input.. programmably set to '0'.. or programmably set to '1'. |
| 29-28 | Reserved | R | 0h | Reserved |
| 27-16 | CFG_DSPLY_AUX_Y_ORIGIN | R/W | 0h | Row origin index for the Aux source picture. This source picture is the result of any cropping by cfg_aux_skip_numlines. (x..y) = (0..0) is the top left corner of the display. |
| 15-12 | Reserved | R | 0h | Reserved |
| 11-0 | CFG_DSPLAY_AUX_X_ORIGIN | R/W | 0h | Column origin index for the Aux source picture. This source picture is the result of any cropping by cfg_aux_skip_numpix. (x..y) = (0..0) is the top left corner of the display. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.10 VCOMP_reg9 Register (offset = 24h) [reset = 0h]

VCOMP_reg9 is shown in Figure 12-554 and described in Table 12-472.

**Figure 12-554. VCOMP_reg9 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CFG_MAIN_AUX_N_ONTOP | Reserved | CFG_DSPLY_BCKGRND_CR_VAL | | | | | |
| R/W-0h | R-0h | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CFG_DSPLY_BCKGRND_CR_VAL | | | | CFG_DSPLY_BCKGRND_CB_VAL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CFG_DSPLY_BCKGRND_CB_VAL | | | | | | CFG_DSPLY_BCKGRND_Y_VAL | |
| R/W-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFG_DSPLY_BCKGRND_Y_VAL | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-472. VCOMP_reg9 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CFG_MAIN_AUX_N_ONTOP | R/W | 0h | When '1..' the Main window takes precedence over the Aux window in display areas where they overlap.<br>When '0..' the Aux window takes precedence over the Main window in display areas where they overlap. |
| 30 | Reserved | R | 0h | Reserved |
| 29-20 | CFG_DSPLY_BCKGRND_CR_VAL | R/W | 0h | Background Cr/Chroma value for the display output. |
| 19-10 | CFG_DSPLY_BCKGRND_CB_VAL | R/W | 0h | Background Cb/Chroma value for the display output. |
| 9-0 | CFG_DSPLY_BCKGRND_Y_VAL | R/W | 0h | Background Y/Luma value for the display output. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.11 VCOMP_reg10 Register (offset = 28h) [reset = 0h]

VCOMP_reg10 is shown in Figure 12-555 and described in Table 12-473.

**Figure 12-555. VCOMP_reg10 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CFG_DSPLY_TIMEOUT_COUNT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-473. VCOMP_reg10 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CFG_DSPLY_TIMEOUT_COUNT | R/W | 0h | Timeout counter in DSS system clock cycles. An internal counter is reset to this value following a vpi_fp pulse to the VCOMP module. The VCOMP will start sending data to the downsteam module after the counter counts down to 0. The NF will be sent downstream based on the Main or Aux NF arrivals and the setting for cfg_nf_handling. The NF output will not be delayed. Likewise.. the Main and Aux request of data from the upstream modules are not delayed. Only the downstream output of pixel data is delayed. Generally.. this register should be set to 0x0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.12 VCOMP_reg11 Register (offset = 2Ch) [reset = 0h]

VCOMP_reg11 is shown in Figure 12-556 and described in Table 12-474.

**Figure 12-556. VCOMP_reg11 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CFG_NF_HANDLING | | CFG_DSPLY_ALT_MAIN_CR_VAL | | | | | |
| R/W-0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CFG_DSPLY_ALT_MAIN_CR_VAL | | | | CFG_DSPLY_ALT_MAIN_CB_VAL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CFG_DSPLY_ALT_MAIN_CB_VAL | | | | | | CFG_DSPLY_ALT_MAIN_Y_VAL | |
| R/W-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFG_DSPLY_ALT_MAIN_Y_VAL | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-474. VCOMP_reg11 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | CFG_NF_HANDLING | R/W | 0h | 00' = When both Main and Aux sources are enabled.. the first NF to enter the VCOMP is used<br>01 = Undefined<br>10 = When both Main and Aux sources are enabled.. the Main source's NF is used<br>11 = When both Main and Aux sources are enabled.. the Aux source's NF is used |
| 29-20 | CFG_DSPLY_ALT_MAIN_CR_VAL | R/W | 0h | Alternate Main Cr/Chroma value for the background display output. Used when main_fixed_data_send = 1. |
| 19-10 | CFG_DSPLY_ALT_MAIN_CB_VAL | R/W | 0h | Alternate Main Cb/Chroma value for the background display output. Used when main_fixed_data_send = 1. |
| 9-0 | CFG_DSPLY_ALT_MAIN_Y_VAL | R/W | 0h | Alternate Main Y/Luma value for the background display output. Used when main_fixed_data_send = 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.13.13 VCOMP_reg12 Register (offset = 30h) [reset = 0h]

VCOMP_reg12 is shown in Figure 12-557 and described in Table 12-475.

**Figure 12-557. VCOMP_reg12 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CFG_DWNSTRM_ENC_FID_CTRL | | CFG_DSPLY_ALT_AUX_CR_VAL | | | | | |
| R/W-0h | | R/W-0h | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CFG_DSPLY_ALT_AUX_CR_VAL | | | | CFG_DSPLY_ALT_AUX_CB_VAL | | | |
| R/W-0h | | | | R/W-0h | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CFG_DSPLY_ALT_AUX_CB_VAL | | | | | | CFG_DSPLY_ALT_AUX_Y_VAL | |
| R/W-0h | | | | | | R/W-0h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CFG_DSPLY_ALT_AUX_Y_VAL | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-475. VCOMP_reg12 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | CFG_DWNSTRM_ENC_FID_CTRL | R/W | 0h | 00 = Encoder FID (bit 1) out of the VCOMP is the same as the respective bit from the Upstream Module<br>01 = Encoder FID (bit 1) out of the VCOMP is the inverse of the respective bit from the Upstream Module<br>10 = Encoder FID (bit 1) out of the VCOMP is '0'.<br>11 = Encoder FID (bit 1) out of the VCOMP is '1'.<br>When both the Main and Aux sources are enabled.. NF and the Source/Encoder FID values are taken from the Main or Aux ports as defined in cfg_nf_handling. Otherwise.. NF and the Source/Encoder FID values are taken from the Active port. In no plane is active.. then no NF will be sent downstream. This register setting enables Bit 1.. representing the Encoder FID.. to be the same as the Upstream Input.. the inverse of the Upstream Input.. programmably set to '0'.. or programmably set to '1'. |
| 29-20 | CFG_DSPLY_ALT_AUX_CR_VAL | R/W | 0h | Alternate Aux Cr/Chroma value for the background display output. Used when aux_fixed_data_send = 1. |
| 19-10 | CFG_DSPLY_ALT_AUX_CB_VAL | R/W | 0h | Alternate Aux Cb/Chroma value for the background display output. Used when aux_fixed_data_send = 1. |
| 9-0 | CFG_DSPLY_ALT_AUX_Y_VAL | R/W | 0h | Alternate Aux Y/Luma value for the background display output. Used when aux_fixed_data_send = 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14 VIP_PARSER Registers

Table 12-476 lists the memory-mapped registers for the VIP_PARSER. All register offset addresses not listed in Table 12-476 should be considered as reserved locations and the register contents should not be modified.

**Table 12-476. VIP_PARSER REGISTERS**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | VIP_PARSER_main | Main Configuration for VIP Parser | Section 12.3.14.1 |
| 4h | VIP_PARSER_port_a | Configuration for Input Port A | Section 12.3.14.2 |
| 8h | VIP_PARSER_xtra_port_a | More Configuration for Input Port A | Section 12.3.14.3 |
| Ch | VIP_PARSER_port_b | Configuration for Input Port B | Section 12.3.14.4 |
| 10h | VIP_PARSER_xtra_port_b | More Configuration for Input Port B | Section 12.3.14.5 |
| 14h | VIP_PARSER_fiq_mask | Interrupt Mask Register | Section 12.3.14.6 |
| 18h | VIP_PARSER_fiq_clear | Interrupt Clear Register | Section 12.3.14.7 |
| 1Ch | VIP_PARSER_fiq_status | Interrupt Status Register | Section 12.3.14.8 |
| 20h | VIP_PARSER_output_port_a_src_fid | Current and Previous Output Port A Source FID values | Section 12.3.14.9 |
| 24h | VIP_PARSER_output_port_a_enc_fid | Current and Previous Output Port A Encoder FID values | Section 12.3.14.10 |
| 28h | VIP_PARSER_output_port_b_src_fid | Current and Previous Output Port B Source FID values | Section 12.3.14.11 |
| 2Ch | VIP_PARSER_output_port_b_enc_fid | Current and Previous Output Port B Encoder FID values | Section 12.3.14.12 |
| 30h | VIP_PARSER_output_port_a_src0_size | Width and Height for Source 0 | Section 12.3.14.13 |
| 34h | VIP_PARSER_output_port_a_src1_size | Width and Height for Source 1 | Section 12.3.14.14 |
| 38h | VIP_PARSER_output_port_a_src2_size | Width and Height for Source 2 | Section 12.3.14.15 |
| 3Ch | VIP_PARSER_output_port_a_src3_size | Width and Height for Source 3 | Section 12.3.14.16 |
| 40h | VIP_PARSER_output_port_a_src4_size | Width and Height for Source 4 | Section 12.3.14.17 |
| 44h | VIP_PARSER_output_port_a_src5_size | Width and Height for Source 5 | Section 12.3.14.18 |
| 48h | VIP_PARSER_output_port_a_src6_size | Width and Height for Source 6 | Section 12.3.14.19 |
| 4Ch | VIP_PARSER_output_port_a_src7_size | Width and Height for Source 7 | Section 12.3.14.20 |
| 50h | VIP_PARSER_output_port_a_src8_size | Width and Height for Source 8 | Section 12.3.14.21 |
| 54h | VIP_PARSER_output_port_a_src9_size | Width and Height for Source 9 | Section 12.3.14.22 |
| 58h | VIP_PARSER_output_port_a_src10_size | Width and Height for Source 10 | Section 12.3.14.23 |
| 5Ch | VIP_PARSER_output_port_a_src11_size | Width and Height for Source 11 | Section 12.3.14.24 |
| 60h | VIP_PARSER_output_port_a_src12_size | Width and Height for Source 12 | Section 12.3.14.25 |
| 64h | VIP_PARSER_output_port_a_src13_size | Width and Height for Source 13 | Section 12.3.14.26 |
| 68h | VIP_PARSER_output_port_a_src14_size | Width and Height for Source 14 | Section 12.3.14.27 |
| 6Ch | VIP_PARSER_output_port_a_src15_size | Width and Height for Source 15 | Section 12.3.14.28 |
| 70h | VIP_PARSER_output_port_b_src0_size | Width and Height for Source 0 | Section 12.3.14.29 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-476. VIP_PARSER REGISTERS (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 74h | VIP_PARSER_output_port_b_src1_size | Width and Height for Source 1 | Section 12.3.14.30 |
| 78h | VIP_PARSER_output_port_b_src2_size | Width and Height for Source 2 | Section 12.3.14.31 |
| 7Ch | VIP_PARSER_output_port_b_src3_size | Width and Height for Source 3 | Section 12.3.14.32 |
| 80h | VIP_PARSER_output_port_b_src4_size | Width and Height for Source 4 | Section 12.3.14.33 |
| 84h | VIP_PARSER_output_port_b_src5_size | Width and Height for Source 5 | Section 12.3.14.34 |
| 88h | VIP_PARSER_output_port_b_src6_size | Width and Height for Source 6 | Section 12.3.14.35 |
| 8Ch | VIP_PARSER_output_port_b_src7_size | Width and Height for Source 7 | Section 12.3.14.36 |
| 90h | VIP_PARSER_output_port_b_src8_size | Width and Height for Source 8 | Section 12.3.14.37 |
| 94h | VIP_PARSER_output_port_b_src9_size | Width and Height for Source 9 | Section 12.3.14.38 |
| 98h | VIP_PARSER_output_port_b_src10_size | Width and Height for Source 10 | Section 12.3.14.39 |
| 9Ch | VIP_PARSER_output_port_b_src11_size | Width and Height for Source 11 | Section 12.3.14.40 |
| A0h | VIP_PARSER_output_port_b_src12_size | Width and Height for Source 12 | Section 12.3.14.41 |
| A4h | VIP_PARSER_output_port_b_src13_size | Width and Height for Source 13 | Section 12.3.14.42 |
| A8h | VIP_PARSER_output_port_b_src14_size | Width and Height for Source 14 | Section 12.3.14.43 |
| ACh | VIP_PARSER_output_port_b_src15_size | Width and Height for Source 15 | Section 12.3.14.44 |
| B0h | VIP_PARSER_port_a_vdet_vec | VDET bit settings for Line Mux Mode for Port A | Section 12.3.14.45 |
| B4h | VIP_PARSER_port_b_vdet_vec | VDET bit setting sfor Line Mux Mode for Port B | Section 12.3.14.46 |
| B8h | VIP_PARSER_xtra2_port_a | Ancillary Cropping Configuration for Input Port A | Section 12.3.14.47 |
| BCh | VIP_PARSER_xtra3_port_a | Ancillary Cropping Configuration for Input Port A | Section 12.3.14.48 |
| C0h | VIP_PARSER_xtra4_port_a | Active Video Cropping Configuration for Input Port A | Section 12.3.14.49 |
| C4h | VIP_PARSER_xtra5_port_a | Active Video Cropping Configuration for Input Port A | Section 12.3.14.50 |
| C8h | VIP_PARSER_xtra2_port_b | Ancillary Cropping Configuration for Input Port B | Section 12.3.14.51 |
| CCh | VIP_PARSER_xtra3_port_b | Ancillary Cropping Configuration for Input Port B | Section 12.3.14.52 |
| D0h | VIP_PARSER_xtra4_port_b | Active Video Cropping Configuration for Input Port B | Section 12.3.14.53 |
| D4h | VIP_PARSER_xtra5_port_b | Active Video Cropping Configuration for Input Port B | Section 12.3.14.54 |
| D8h | VIP_PARSER_xtra6_port_a | Cfg Disable Active Srcnum Vector Input for Port A | Section 12.3.14.55 |
| DCh | VIP_PARSER_xtra7_port_b | Cfg Disable Active Srcnum Vector Input for Port B | Section 12.3.14.56 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.1 VIP_PARSER_main Register (offset = 0h) [reset = 0h]

VIP_PARSER_main is shown in Figure 12-558 and described in Table 12-477.

Main Configuration for VIP Parser

**Figure 12-558. VIP_PARSER_main Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | CLIP_ACTIVE | CLIP_BLNK | Reserved | | DATA_INTERFACE_MODE | |
| R-0h | | R/W-0h | R/W-0h | R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-477. VIP_PARSER_main Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0h | |
| 5 | CLIP_ACTIVE | R/W | 0h | Discrete Sync Only '0' = Do not clip active pixels '1' = Clip Active Pixels as follows: 0xFF -andgt; 0xFE.. 0x00 -andgt; 0x01 |
| 4 | CLIP_BLNK | R/W | 0h | Discrete Sync Only '0' = Do not clip Blanking Data '1' = Clip Blanking Data as follows: 0xFF -andgt; 0xFE.. 0x00 -andgt; 0x01 |
| 3-2 | Reserved | R | 0h | |
| 1-0 | DATA_INTERFACE_MODE | R/W | 0h | 00 = 24b data interface. Uses Port A settings 01 = 16b data interface. Uses Port A settings. 10 = Dual independent 8b data interfaces. Uses independent Port A and Port B settings. 11 = Undefined |

**NOTE:** Embedded sync input source is already clipped because the two 0x00 and 0xFF codes are illegal values in the active video data and they are reserved only for start codes.

If there is any need to down the line to pack this stream into an embedded sync format (input source contains the 0x00 and 0xFF codes), then the VIP Parser can be configured to clip the data on the way to being stored in memory. This prevents MPU or DSP to go through every byte to check if the data has to be clipped.



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.2 VIP_PARSER_port_a Register (offset = 4h) [reset = 0h]

VIP_PARSER_port_a is shown in Figure 12-559 and described in Table 12-478.

Configuration for Input Port A

**Figure 12-559. VIP_PARSER_port_a Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| analyzer_fvh_err_correction_enable | ANALYZER_2X4X_SRCNUM_POS | FID_SKEW_POSTCOUNT | | | | | |
| R-0h | R/W-0h | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SW_RESET | DISCRETE_BASIC_MODE | FID_SKEW_PRECOUNT | | | | | |
| R/W-0h | R/W-0h | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| USE_ACTVID_HSYNC_N | FID_DETECT_MODE | ACTVID_POLARITY | VSYNC_POLARITY | HSYNC_POLARITY | PIXCLK_EDGE_POLARITY | FID_POLARITY | ENABLE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLR_ASYNC_FIFO_RD | CLR_ASYNC_FIFO_WR | CTRL_CHAN_SEL | | SYNC_TYPE | | | |
| R/W-0h | R/W-0h | R/W-0h | | R/W-0h | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-478. VIP_PARSER_port_a Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | analyzer_fvh_err_correction_enable | R/W | 0h | Embedded Sync Only<br>0 = Ignore the protection bits in the XV (fvh) codeword header. This setting is typically desired.<br>1 = Use the protection bits in an attempt to do error correction for the fvh control bits. |
| 30 | ANALYZER_2X4X_SRCNUM_POS | R/W | 0h | Embedded Sync Only<br>0 = For 2x/4x mux mode.. srcnum is in the least significant nibble of the XV/fvh codeword (srcnum replaces the protection bits)<br>1 = For 2x/4x mux mode.. srcnum is in the least significant nibble of a horizontal blanking pixel value |
| 29-24 | FID_SKEW_POSTCOUNT | R/W | 0h | Discrete Sync Only post count value when using vsync skew in FID determination |
| 23 | SW_RESET | R/W | 0h | 0 = Normal<br>1 = Reset Port A logic. Must be set to '0' again by the software for the module to function. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-478. VIP_PARSER_port_a Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 22 | DISCRETE_BASIC_MODE | R/W | 0h | This register is valid for Discrete Sync mode only.<br>0 = Normal Discrete Mode.<br>Hsync Style Capture operates as follows:<br>- Captures line starting from HSYNC inactive to active condition.<br>- VSYNC determined by sync window.<br>- FID can be determined by VSYNC skew or captured from pin at first pixel in first line.<br>ACTVID style capture works as follows:<br>- Captures line during contiguous ACTVID envelope.<br>- VSYNC is captured at the first pixel in each line.<br>- FID is captured on first pixel of ACTVID window.<br>1 = Basic Discrete Mode.<br>When using hsync with Hsync Style Capture operates as follows:<br>- The last line of active video ends on the pixel clock cycle where VSYNC transitions from inactive to active.<br>- FID pin value is captured on this cycle and is used for the next field.<br>- FID detection by VSYNC skew is not allowed.<br>ACTVID style capture works as follows:<br>- VSYNC is expected to transition from inactive to active between ACTVID window.<br>- This VSYNC transition allows the next line in an ACTVID envelope to be sent to a new VPDMA buffer.<br>- FID value is determined by the FID pin value on the cycle where VSYNC transitions from inactive to active.<br>In basic discrete mode, there is no Vertical Ancillary Data. Therefore, VPDMA descriptors should not use Ancillary Data channels. |
| 21-16 | FID_SKEW_PRECOUNT | R/W | 0h | Discrete Sync Only pre count value when using vsync skew in FID determination |
| 15 | USE_ACTVID_HSYNC_N | R/W | 0h | Discrete Sync Only<br>0 = Use HSYNC style line capture<br>1 = Use ACTVID style line capture |
| 14 | FID_DETECT_MODE | R/W | 0h | Discrete Sync Only<br>0 = Take FID from pin<br>1 = FID is determined by VSYNC skew |
| 13 | ACTVID_POLARITY | R/W | 0h | Discrete Sync Only<br>0 = ACTVID is active low<br>1 = ACTVID is active high |
| 12 | VSYNC_POLARITY | R/W | 0h | Discrete Sync Only<br>0 = VSYNC is active low<br>1 = VSYNC is active high |
| 11 | HSYNC_POLARITY | R/W | 0h | Discrete Sync Only<br>0 = HSYNC is active low<br>1 = HSYNC is active high |
| 10 | PIXCLK_EDGE_POLARITY | R/W | 0h | 0 = Rising Edge is active PIXCLK edge<br>1 = Falling Edge is active PIXCLK edge |
| 9 | FID_POLARITY | R/W | 0h | 0 = Keep FID as found<br>1 = Invert Determined Value of FID<br>Value on FID pin in discrete sync or value of "F-bit" in embedded sync indicates whether a field is EVEN or ODD (time-domain). However modules in HDVPSS follow convention (top/bottom) as specified in the table of HDVPSS Acronyms. As a result, fid_polarity needs to be set appropriately. Typically, this will result in different configuration for fid_polarity between PAL and NTSC modes. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-478. VIP_PARSER_port_a Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | ENABLE | R/W | 0h | 0 = Disable Port<br>1 = Enable Port |
| 7 | CLR_ASYNC_FIFO_RD | R/W | 0h | 0 = Normal<br>1 = Clear Async FIFO Read Logic |
| 6 | CLR_ASYNC_FIFO_WR | R/W | 0h | 0 = Normal<br>1 = Clear Async FIFO Write Logic |
| 5-4 | CTRL_CHAN_SEL | R/W | 0h | Embedded Sync Only In 8b mode.. there is only one channel on data[7:0]. In 16b mode.. there are two channels. The Luma Channel is on data[15:8]. The Chroma Channel is on data[7:0]. In 24b mode.. there are three channels. The R channel is on data[23:16].. the G channel is on [15:8].. and the B channel is on data[7:0].<br>00 = Use data[7:0] to extract control codes.<br>01 = Use data[15:8] to extract control codes.<br>10 = Use data[23:16] to extract control codes.<br>11 = Undefined<br>In 16b and 24b modes.. this register is also used to select the channel from which Ancillary Data is extracted. The Ancillary Data channel must be the same as the control code channel. For 8b mode.. the anc_chan_sel_8b register is used to select the Luma or Chroma channel from which Ancillary Data is taken. |
| 3-0 | SYNC_TYPE | R/W | 0h | 0000 = embedded sync single 4:2:2 YUV stream<br>0001 = embedded sync 2x multiplexed 4:2:2 YUV stream<br>0010 = embedded sync 4x multiplexed 4:2:2 YUV stream<br>0011 = embedded sync line multiplexed 4:2:2 YUV stream<br>0100 = discrete sync single 4:2:2 YUV stream<br>0101 = embedded sync single RGB stream or single 444 YUV stream<br>0110 = reserved<br>0111 = reserved<br>1000 = reserved<br>1001 = reserved<br>1010 = discrete sync single 24b RGB stream |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.3 VIP_PARSER_xtra_port_a Register (offset = 8h) [reset = 0h]

VIP_PARSER_xtra_port_a is shown in Figure 12-560 and described in Table 12-479.

More Configuration for Input Port A

**Figure 12-560. VIP_PARSER_xtra_port_a Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | SRC0_NUMPIX | | |
| R-0h | | | | | R/W-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| SRC0_NUMPIX | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | ANC_CHAN_SEL_8B | | Reserved | | SRC0_NUMLINES | | |
| R-0h | R/W-0h | | R-0h | | R/W-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SRC0_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-479. VIP_PARSER_xtra_port_a Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | SRC0_NUMPIX | R/W | 0h | Number of expected pixels on Source Number 0. The Port_a_src0_size interrupt will trigger if a line is encountered that differs from this pixelcount. |
| 15 | Reserved | R | 0h | |
| 14-13 | ANC_CHAN_SEL_8B | R/W | 0h | In 8b mode, Vertically Ancillary Data typically resides in the Luma sites. This bit allows vertical ancillary data to be extracted from the chroma sites instead.<br>00 = Extract 8b Mode Vertical Ancillary Data from Luma Sites<br>01 = Extract 8b Mode Vertical Ancillary Data from Chroma Sites<br>10, 11 = Extract every single sample of vertical ancillary data<br>The output line is twice as wide as the other modes. For 16b and 24b inputs, ctrl_chan_sel is used to select which channel is used as a source for vertical ancillary data. |
| 12-11 | Reserved | R | 0h | |
| 10-0 | SRC0_NUMLINES | R/W | 0h | Number of expected lines on Source Number 0. The Port_a_src0_size interrupt will trigger if a field/frame is encountered that differs from this linecount. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.4 VIP_PARSER_port_b Register (offset = Ch) [reset = 0h]

VIP_PARSER_port_b is shown in Figure 12-561 and described in Table 12-480.

Configuration for Input Port B

**Figure 12-561. VIP_PARSER_port_b Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| analyzer_fvh_err_correction_enable | ANALYZER_2X4X_SRCNUM_POS | FID_SKEW_POSTCOUNT | | | | | |
| R-0h | R/W-0h | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SW_RESET | DISCRETE_BASIC_MODE | FID_SKEW_PRECOUNT | | | | | |
| R/W-0h | R/W-0h | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| USE_ACTVID_HSYNC_N | FID_DETECT_MODE | ACTVID_POLARITY | VSYNC_POLARITY | HSYNC_POLARITY | PIXCLK_EDGE_POLARITY | FID_POLARITY | ENABLE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLR_ASYNC_FIFO_RD | CLR_ASYNC_FIFO_WR | CTRL_CHAN_SEL | | SYNC_TYPE | | | |
| R/W-0h | R/W-0h | R/W-0h | | R/W-0h | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-480. VIP_PARSER_port_b Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | analyzer_fvh_err_correction_enable | R/W | 0h | Embedded Sync Only<br>0 = Ignore the protection bits in the XV (fvh) codeword header. This setting is typically desired.<br>1 = Use the protection bits in an attempt to do error correction for the fvh control bits. |
| 30 | ANALYZER_2X4X_SRCNUM_POS | R/W | 0h | Embedded Sync Only<br>0 = For 2x/4x mux mode.. srcnum is in the least significant nibble of the XV/fvh codeword (srcnum replaces the protection bits)<br>1 = For 2x/4x mux mode.. srcnum is in the least significant nibble of a horizontal blanking pixel value |
| 29-24 | FID_SKEW_POSTCOUNT | R/W | 0h | Discrete Sync Only post count value when using vsync skew in FID determination |
| 23 | SW_RESET | R/W | 0h | 0 = Normal<br>1 = Reset Port B logic. Must be set to '0' again by the software for the module to function. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-480. VIP_PARSER_port_b Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 22 | DISCRETE_BASIC_MODE | R/W | 0h | This register is valid for Discrete Sync mode only.<br>0 = Normal Discrete Mode.<br>Hsync Style Capture operates as follows:<br>- Captures line starting from HSYNC inactive to active condition.<br>- VSYNC determined by sync window.<br>- FID can be determined by VSYNC skew or captured from pin at first pixel in first line.<br>ACTVID style capture works as follows:<br>- Captures line during contiguous ACTVID envelope.<br>- VSYNC is captured at the first pixel in each line.<br>- FID is captured on first pixel of ACTVID window.<br>1 = Basic Discrete Mode.<br>When using hsync with Hsync Style Capture operates as follows:<br>- The last line of active video ends on the pixel clock cycle where VSYNC transitions from inactive to active.<br>- FID pin value is captured on this cycle and is used for the next field.<br>- FID detection by VSYNC skew is not allowed.<br>ACTVID style capture works as follows:<br>- VSYNC is expected to transition from inactive to active between ACTVID window.<br>- This VSYNC transition allows the next line in an ACTVID envelope to be sent to a new VPDMA buffer.<br>- FID value is determined by the FID pin value on the cycle where VSYNC transitions from inactive to active.<br>In basic discrete mode, there is no Vertical Ancillary Data. Therefore, VPDMA descriptors should not use Ancillary Data channels. |
| 21-16 | FID_SKEW_PRECOUNT | R/W | 0h | Discrete Sync Only pre count value when using vsync skew in FID determination |
| 15 | USE_ACTVID_HSYNC_N | R/W | 0h | Discrete Sync Only<br>0 = Use HSYNC style line capture<br>1 = Use ACTVID style line capture |
| 14 | FID_DETECT_MODE | R/W | 0h | Discrete Sync Only<br>0 = Take FID from pin<br>1 = FID is determined by VSYNC skew |
| 13 | ACTVID_POLARITY | R/W | 0h | Discrete Sync Only<br>0 = ACTVID is active low<br>1 = ACTVID is active high |
| 12 | VSYNC_POLARITY | R/W | 0h | Discrete Sync Only<br>0 = VSYNC is active low<br>1 = VSYNC is active high |
| 11 | HSYNC_POLARITY | R/W | 0h | Discrete Sync Only<br>0 = HSYNC is active low<br>1 = HSYNC is active high |
| 10 | PIXCLK_EDGE_POLARITY | R/W | 0h | 0 = Rising Edge is active PIXCLK edge<br>1 = Falling Edge is active PIXCLK edge |
| 9 | FID_POLARITY | R/W | 0h | 0 = Keep FID as found<br>1 = Invert Determined Value of FID<br>Value on FID pin in discrete sync or value of "F-bit" in embedded sync indicates whether a field is EVEN or ODD (time-domain). However modules in HDVPSS follow convention (top/bottom) as specified in the table of HDVPSS Acronyms. As a result, fid_polarity needs to be set appropriately. Typically, this will result in different configuration for fid_polarity between PAL and NTSC modes. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-480. VIP_PARSER_port_b Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | ENABLE | R/W | 0h | 0 = Disable<br>1 = Enable |
| 7 | CLR_ASYNC_FIFO_RD | R/W | 0h | 0 = Normal<br>1 = Clear Async FIFO Read Logic |
| 6 | CLR_ASYNC_FIFO_WR | R/W | 0h | 0 = Normal<br>1 = Clear Async FIFO Write Logic |
| 5-4 | CTRL_CHAN_SEL | R/W | 0h | PORT B supports on 8b mode. Always write 0 to this field.<br>The anc_chan_sel_8b register is used to select the Luma or Chroma channel from which Ancillary Data is taken. |
| 3-0 | SYNC_TYPE | R/W | 0h | 0000 = embedded sync single YUV stream<br>0001 = embedded sync 2x multiplexed YUV stream<br>0010 = embedded sync 4x multiplexed YUV stream<br>0011 = embedded sync line multiplexed YUV stream<br>0100 = discrete sync single YUV stream<br>0101 = embedded sync single RGB stream<br>0110 = reserved<br>0111 = reserved<br>1000 = reserved<br>1001 = reserved<br>1010 = discrete sync single 24b RGB stream |

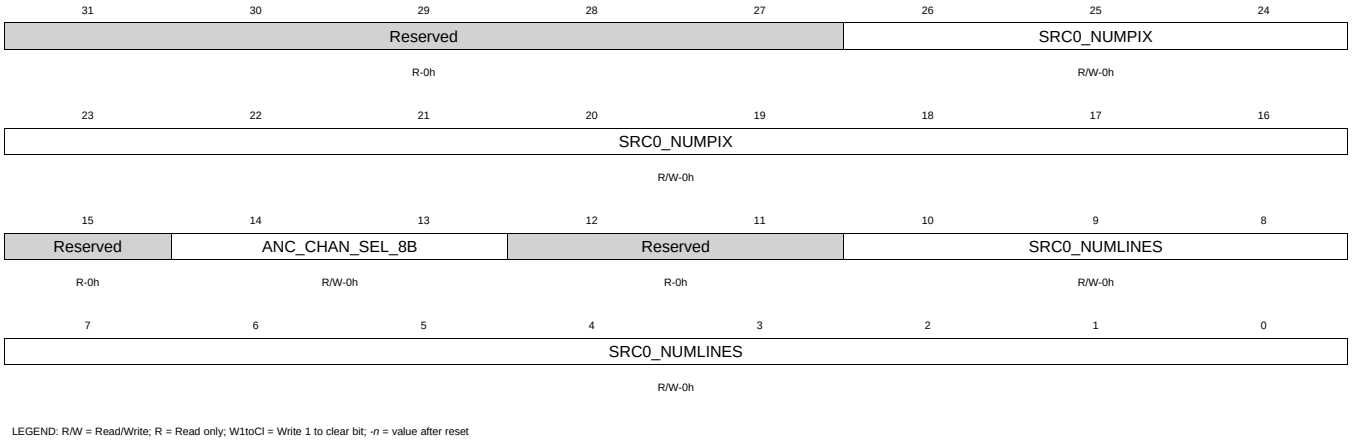

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.5 VIP_PARSER_xtra_port_b Register (offset = 10h) [reset = 0h]

VIP_PARSER_xtra_port_b is shown in Figure 12-562 and described in Table 12-481.

More Configuration for Input Port B

**Figure 12-562. VIP_PARSER_xtra_port_b Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-481. VIP_PARSER_xtra_port_b Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | SRC0_NUMPIX | R/W | 0h | Number of expected pixels on Source Number 0. The Port_b_src0_size interrupt will trigger if a line is encountered that differs from this pixelcount. |
| 15 | Reserved | R | 0h | |
| 14-13 | ANC_CHAN_SEL_8B | R/W | 0h | In 8b mode, Vertically Ancillary Data typically resides in the Luma sites. This bit allows vertical ancillary data to be extracted from the chroma sites instead.<br>00 = Extract 8b Mode Vertical Ancillary Data from Luma Sites<br>01 = Extract 8b Mode Vertical Ancillary Data from Chroma Sites<br>10, 11 = Extract every single sample of vertical ancillary data. The output line is twice as wide as the other modes. |
| 12-11 | Reserved | R | 0h | |
| 10-0 | SRC0_NUMLINES | R/W | 0h | Number of expected lines on Source Number 0. The Port_b_src0_size interrupt will trigger if a field/frame is encountered that differs from this linecount. |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.6 VIP_PARSER_fiq_mask Register (offset = 14h) [reset = 0h]

VIP_PARSER_fiq_mask is shown in Figure 12-563 and described in Table 12-482.

Mask Bits for ARM FIQs. A 0 means the Host ARM will get an interrupt if the hardware sets one. A 1 in the mask bitfield means that the hardware interrupt is masked and the ARM will not be interrupted.

**Figure 12-563. VIP_PARSER_fiq_mask Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | PORT_B_CFG_DISABLE_COMPLETE_MASK | PORT_A_CFG_DISABLE_COMPLETE_MASK | PORT_B_ANC_PROTOCOL_VIOLATION_MASK | PORT_B_YUV_PROTOCOL_VIOLATION_MASK | PORT_A_ANC_PROTOCOL_VIOLATION_MASK | PORT_A_YUV_PROTOCOL_VIOLATION_MASK |
| R-0h | | R/W-0h | R/W-0h | | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PORT_B_SRC0_SIZE | PORT_A_SRC0_SIZE | PORT_B_DISCONN | PORT_B_CONN | PORT_A_DISCONN | PORT_A_CONN | OUTPUT_FIFO_PRTB_ANC_OF | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUTPUT_FIFO_PRTB_YUV_OF | OUTPUT_FIFO_PRTA_ANC_OF | Reserved | OUTPUT_FIFO_PRTA_YUV_OF | ASYNC_FIFO_PRTB_OF | ASYNC_FIFO_PRTA_OF | PRTB_VDET_MASK | PRTA_VDET_MASK |
| R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-482. VIP_PARSER_fiq_mask Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | Reserved | R | 0h | |
| 21 | PORT_B_CFG_DISABLE_COMPLETE_MASK | R/W | 0h | Port B Cfg Disable Complete Mask |
| 20 | PORT_A_CFG_DISABLE_COMPLETE_MASK | R/W | 0h | Port A Cfg Disable Complete Mask |
| 19 | PORT_B_ANC_PROTOCOL_VIOLATION_MASK | R/W | 0h | Port B ANC VPI Protocol Violation Mask |
| 18 | PORT_B_YUV_PROTOCOL_VIOLATION_MASK | R/W | 0h | Port B YUV VPI Protocol Violation Mask |
| 17 | PORT_A_ANC_PROTOCOL_VIOLATION_MASK | R/W | 0h | Port A ANC VPI Protocol Violation Mask |
| 16 | PORT_A_YUV_PROTOCOL_VIOLATION_MASK | R/W | 0h | Port A YUV VPI Protocol Violation Mask |
| 15 | PORT_B_SRC0_SIZE | R/W | 0h | Video size detected on Port B does not match size programmed in xtra_port_b register |
| 14 | PORT_A_SRC0_SIZE | R/W | 0h | Video size detected on Port A does not match size programmed in xtra_port_a register |
| 13 | PORT_B_DISCONN | R/W | 0h | Port B Link Disconnect Srcnum 0 Mask |
| 12 | PORT_B_CONN | R/W | 0h | Port B Link Connect Srcnum 0 Mask |
| 11 | PORT_A_DISCONN | R/W | 0h | Port A Link Disconnect Scrnum 0 Mask |
| 10 | PORT_A_CONN | R/W | 0h | Port A Link Connect Srcnum 0 Mask |
| 9 | OUTPUT_FIFO_PRTB_ANC_OF | R/W | 0h | Output FIFO Port B Ancillary Overflow Mask |
| 8 | Reserved | R | 0h | |
| 7 | OUTPUT_FIFO_PRTB_YUV_OF | R/W | 0h | Output FIFO Port B Luma Overflow Mask |
| 6 | OUTPUT_FIFO_PRTA_ANC_OF | R/W | 0h | Output FIFO Port A Ancillary Overflow Mask |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-482. VIP_PARSER_fiq_mask Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | Reserved | R | 0h | |
| 4 | OUTPUT_FIFO_PRTA_YUV_OF | R/W | 0h | Output FIFO Port A Luma Overflow Mask |
| 3 | ASYNC_FIFO_PRTB_OF | R/W | 0h | Port B Async FIFO Overflow FIQ Mask |
| 2 | ASYNC_FIFO_PRTA_OF | R/W | 0h | Port A Async FIFO Overflow FIQ Mask |
| 1 | PRTB_VDET_MASK | R/W | 0h | Port B Video Detect FIQ Mask |
| 0 | PRTA_VDET_MASK | R/W | 0h | Port A Video Detect FIQ Mask |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.7 VIP_PARSER_fiq_clear Register (offset = 18h) [reset = 0h]

VIP_PARSER_fiq_clear is shown in Figure 12-564 and described in Table 12-483.

Clears bits in the FIQ Status

**Figure 12-564. VIP_PARSER_fiq_clear Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | PORT_B_CFG_DISABLE_COMPLETE_CLR | PORT_A_CFG_DISABLE_COMPLETE_CLR | PORT_B_ANC_PROTOCOL_VIOLATION_CLR | PORT_B_YUV_PROTOCOL_VIOLATION_CLR | PORT_A_ANC_PROTOCOL_VIOLATION_CLR | PORT_A_YUV_PROTOCOL_VIOLATION_CLR |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PORT_B_SRC0_SIZE_CLR | PORT_A_SRC0_SIZE_CLR | PORT_B_DISCONN_CLR | PORT_B_CONN_CLR | PORT_A_DISCONN_CLR | PORT_A_CONN_CLR | OUTPUT_FIFO_PRTB_ANC_CLR | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUTPUT_FIFO_PRTB_YUV_CLR | OUTPUT_FIFO_PRTA_ANC_CLR | Reserved | OUTPUT_FIFO_PRTA_YUV_CLR | ASYNC_FIFO_PRTB_CLR | ASYNC_FIFO_PRTA_CLR | PRTB_VDET_CLR | PRTA_VDET_CLR |
| R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-483. VIP_PARSER_fiq_clear Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | Reserved | R | 0h | |
| 21 | PORT_B_CFG_DISABLE_COMPLETE_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port B Cfg Disable Complete FIQ |
| 20 | PORT_A_CFG_DISABLE_COMPLETE_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port A Cfg Disable Complete FIQ |
| 19 | PORT_B_ANC_PROTOCOL_VIOLATION_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port B ANC VPI Protocol Violation FIQ |
| 18 | PORT_B_YUV_PROTOCOL_VIOLATION_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port B YUV VPI Protocol Violation FIQ |
| 17 | PORT_A_ANC_PROTOCOL_VIOLATION_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port A ANC VPI Protocol Violation FIQ |
| 16 | PORT_A_YUV_PROTOCOL_VIOLATION_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port A YUV VPI Protocol Violation FIQ |
| 15 | PORT_B_SRC0_SIZE_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port B Src0 Size FIQ |
| 14 | PORT_A_SRC0_SIZE_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port A Src0 Size FIQ |
| 13 | PORT_B_DISCONN_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port B Link Disconnect FIQ |
| 12 | PORT_B_CONN_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port B Link Connect FIQ |
| 11 | PORT_A_DISCONN_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port A Link Disconnect FIQ |
| 10 | PORT_A_CONN_CLR | R/W | 0h | Write '1' followed by '0' to Clear Port A Link Connect FIQ |
| 9 | OUTPUT_FIFO_PRTB_ANC_CLR | R/W | 0h | Write '1' followed by '0' to Clear Output FIFO Port B Ancillary Overflow FIQ |
| 8 | Reserved | R | 0h | |
| 7 | OUTPUT_FIFO_PRTB_YUV_CLR | R/W | 0h | Write '1' followed by '0' to Clear Output FIFO Port B Luma Overflow FIQ |
| 6 | OUTPUT_FIFO_PRTA_ANC_CLR | R/W | 0h | Write '1' followed by '0' to Clear Output FIFO Port A Ancillary Overflow FIQ |
| 5 | Reserved | R | 0h | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-483. VIP_PARSER_fiq_clear Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 4 | OUTPUT_FIFO_PRTA_YUV_CLR | R/W | 0h | Write '1' followed by '0' to Clear Output FIFO Port A Luma Overflow FIQ |
| 3 | ASYNC_FIFO_PRTB_CLR | R/W | 0h | Write '1' followed by '0' to Clear Async FIFO Port B Overflow FIQ |
| 2 | ASYNC_FIFO_PRTA_CLR | R/W | 0h | Write '1' followed by '0' to Clear Async FIFO Port A Overflow FIQ |
| 1 | PRTB_VDET_CLR | R/W | 0h | Write '1' followed by '0' to Clear Video Detect FIQ for Port B |
| 0 | PRTA_VDET_CLR | R/W | 0h | Write '1' followed by '0' to Clear Video Detect FIQ for Port A |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.8 VIP_PARSER_fiq_status Register (offset = 1Ch) [reset = 0h]

VIP_PARSER_fiq_status is shown in Figure 12-565 and described in Table 12-484.

FIQ Status values

**Figure 12-565. VIP_PARSER_fiq_status Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | PORT_B_CFG_DISABLE_COMPLETE_CLR | PORT_A_CFG_DISABLE_COMPLETE | PORT_B_ANC_PROTOCOL_VIOLATION | PORT_B_YUV_PROTOCOL_VIOLATION | PORT_A_ANC_PROTOCOL_VIOLATION | PORT_A_YUV_PROTOCOL_VIOLATION |
| R-0h | | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| PORT_B_SRC0_SIZE_STATUS | PORT_A_SRC0_SIZE_STATUS | PORT_B_DISCONN_STATUS | PORT_B_CONN_STATUS | PORT_A_DISCONN_STATUS | PORT_A_CONN_STATUS | OUTPUT_FIFO_PRTB_ANC_STATUS | OUTPUT_FIFO_PRTB_CHROMA_STATUS |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| OUTPUT_FIFO_PRTB_LUMA_STATUS | OUTPUT_FIFO_PRTA_ANC_STATUS | OUTPUT_FIFO_PRTA_CHROMA_STATUS | OUTPUT_FIFO_PRTA_LUMA_STATUS | ASYNC_FIFO_PRTB_STATUS | ASYNC_FIFO_PRTA_STATUS | PRTB_VDET_STATUS | PRTA_VDET_STATUS |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-484. VIP_PARSER_fiq_status Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | Reserved | R | 0h | |
| 21 | PORT_B_CFG_DISABLE_COMPLETE_CLR | R | 0h | Port B Cfg Disable Complete FIQ |
| 20 | PORT_A_CFG_DISABLE_COMPLETE | R | 0h | Port A Cfg Disable Complete FIQ |
| 19 | PORT_B_ANC_PROTOCOL_VIOLATION | R | 0h | Port B ANC VPI Protocol Violation FIQ |
| 18 | PORT_B_YUV_PROTOCOL_VIOLATION | R | 0h | Port B YUV VPI Protocol Violation FIQ |
| 17 | PORT_A_ANC_PROTOCOL_VIOLATION | R | 0h | Port A ANC VPI Protocol Violation FIQ |
| 16 | PORT_A_YUV_PROTOCOL_VIOLATION | R | 0h | Port A YUV VPI Protocol Violation FIQ |
| 15 | PORT_B_SRC0_SIZE_STATUS | R | 0h | Port B Source 0 Size FIQ |
| 14 | PORT_A_SRC0_SIZE_STATUS | R | 0h | Port A Source 0 Size FIQ |
| 13 | PORT_B_DISCONN_STATUS | R | 0h | Port B Disconnect FIQ |
| 12 | PORT_B_CONN_STATUS | R | 0h | Port B Connect FIQ |
| 11 | PORT_A_DISCONN_STATUS | R | 0h | Port A Disconnect FIQ |
| 10 | PORT_A_CONN_STATUS | R | 0h | Port A Connect FIQ |
| 9 | OUTPUT_FIFO_PRTB_ANC_STATUS | R | 0h | Output FIFO Port B Ancillary Overflow Status |
| 8 | OUTPUT_FIFO_PRTB_CHROMA_STATUS | R | 0h | Output FIFO Port B Chroma Overflow Status |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-484. VIP_PARSER_fiq_status Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | OUTPUT_FIFO_PRTB_LUMA_STATUS | R | 0h | Output FIFO Port B Luma Overflow Status |
| 6 | OUTPUT_FIFO_PRTA_ANC_STATUS | R | 0h | Output FIFO Port A Ancillary Overflow Status |
| 5 | OUTPUT_FIFO_PRTA_CHROMA_STATUS | R | 0h | Output FIFO Port A Chroma Overflow Status |
| 4 | OUTPUT_FIFO_PRTA_LUMA_STATUS | R | 0h | Output FIFO Port A Luma Overflow Status |
| 3 | ASYNC_FIFO_PRTB_STATUS | R | 0h | Async FIFO Port B Overflow Status |
| 2 | ASYNC_FIFO_PRTA_STATUS | R | 0h | Async FIFO Port A Overflow Status |
| 1 | PRTB_VDET_STATUS | R | 0h | VDET Status for Port B |
| 0 | PRTA_VDET_STATUS | R | 0h | VDET Status for Port A |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.9 VIP_PARSER_output_port_a_src_fid Register (offset = 20h) [reset = FFFFFFFFh]

VIP_PARSER_output_port_a_src_fid is shown in Figure 12-566 and described in Table 12-485.

Current and Previous Output Port A Source FID values

**Figure 12-566. VIP_PARSER_output_port_a_src_fid Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC15_CURR_SOURCE_FID | PRTA_SRC15_PREV_SOURCE_FID | PRTA_SRC14_CURR_SOURCE_FID | PRTA_SRC14_PREV_SOURCE_FID | PRTA_SRC13_CURR_SOURCE_FID | PRTA_SRC13_PREV_SOURCE_FID | PRTA_SRC12_CURR_SOURCE_FID | PRTA_SRC12_PREV_SOURCE_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PRTA_SRC11_CURR_SOURCE_FID | PRTA_SRC11_PREV_SOURCE_FID | PRTA_SRC10_CURR_SOURCE_FID | PRTA_SRC10_PREV_SOURCE_FID | PRTA_SRC9_CURR_SOURCE_FID | PRTA_SRC9_PREV_SOURCE_FID | PRTA_SRC8_CURR_SOURCE_FID | PRTA_SRC8_PREV_SOURCE_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PRTA_SRC7_CURR_SOURCE_FID | PRTA_SRC7_PREV_SOURCE_FID | PRTA_SRC6_CURR_SOURCE_FID | PRTA_SRC6_PREV_SOURCE_FID | PRTA_SRC5_CURR_SOURCE_FID | PRTA_SRC5_PREV_SOURCE_FID | PRTA_SRC4_CURR_SOURCE_FID | PRTA_SRC4_PREV_SOURCE_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRTA_SRC3_CURR_SOURCE_FID | PRTA_SRC3_PREV_SOURCE_FID | PRTA_SRC2_CURR_SOURCE_FID | PRTA_SRC2_PREV_SOURCE_FID | PRTA_SRC1_CURR_SOURCE_FID | PRTA_SRC1_PREV_SOURCE_FID | PRTA_SRC0_CURR_SOURCE_FID | PRTA_SRC0_PREV_SOURCE_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-485. VIP_PARSER_output_port_a_src_fid Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PRTA_SRC15_CURR_SOURCE_FID | R | 1h | For Source ID 15 from Port A.. Source Field ID for Current Field |
| 30 | PRTA_SRC15_PREV_SOURCE_FID | R | 1h | For Source ID 15 from Port A.. Source Field ID for Previous Field |
| 29 | PRTA_SRC14_CURR_SOURCE_FID | R | 1h | For Source ID 14 from Port A.. Source Field ID for Current Field |
| 28 | PRTA_SRC14_PREV_SOURCE_FID | R | 1h | For Source ID 14 from Port A.. Source Field ID for Previous Field |
| 27 | PRTA_SRC13_CURR_SOURCE_FID | R | 1h | For Source ID 13 from Port A.. Source Field ID for Current Field |
| 26 | PRTA_SRC13_PREV_SOURCE_FID | R | 1h | For Source ID 13 from Port A.. Source Field ID for Previous Field |
| 25 | PRTA_SRC12_CURR_SOURCE_FID | R | 1h | For Source ID 12 from Port A.. Source Field ID for Current Field |
| 24 | PRTA_SRC12_PREV_SOURCE_FID | R | 1h | For Source ID 12 from Port A.. Source Field ID for Previous Field |
| 23 | PRTA_SRC11_CURR_SOURCE_FID | R | 1h | For Source ID 11 from Port A.. Source Field ID for Current Field |
| 22 | PRTA_SRC11_PREV_SOURCE_FID | R | 1h | For Source ID 11 from Port A.. Source Field ID for Previous Field |
| 21 | PRTA_SRC10_CURR_SOURCE_FID | R | 1h | For Source ID 10 from Port A.. Source Field ID for Current Field |
| 20 | PRTA_SRC10_PREV_SOURCE_FID | R | 1h | For Source ID 10 from Port A.. Source Field ID for Previous Field |
| 19 | PRTA_SRC9_CURR_SOURCE_FID | R | 1h | For Source ID 9 from Port A.. Source Field ID for Current Field |
| 18 | PRTA_SRC9_PREV_SOURCE_FID | R | 1h | For Source ID 9 from Port A.. Source Field ID for Previous Field |
| 17 | PRTA_SRC8_CURR_SOURCE_FID | R | 1h | For Source ID 8 from Port A.. Source Field ID for Current Field |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-485. VIP_PARSER_output_port_a_src_fid Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | PRTA_SRC8_PREV_SOURCE_FID | R | 1h | For Source ID 8 from Port A.. Source Field ID for Previous Field |
| 15 | PRTA_SRC7_CURR_SOURCE_FID | R | 1h | For Source ID 7 from Port A.. Source Field ID for Current Field |
| 14 | PRTA_SRC7_PREV_SOURCE_FID | R | 1h | For Source ID 7 from Port A.. Source Field ID for Previous Field |
| 13 | PRTA_SRC6_CURR_SOURCE_FID | R | 1h | For Source ID 6 from Port A.. Source Field ID for Current Field |
| 12 | PRTA_SRC6_PREV_SOURCE_FID | R | 1h | For Source ID 6 from Port A.. Source Field ID for Previous Field |
| 11 | PRTA_SRC5_CURR_SOURCE_FID | R | 1h | For Source ID 5 from Port A.. Source Field ID for Current Field |
| 10 | PRTA_SRC5_PREV_SOURCE_FID | R | 1h | For Source ID 5 from Port A.. Source Field ID for Previous Field |
| 9 | PRTA_SRC4_CURR_SOURCE_FID | R | 1h | For Source ID 4 from Port A.. Source Field ID for Current Field |
| 8 | PRTA_SRC4_PREV_SOURCE_FID | R | 1h | For Source ID 4 from Port A.. Source Field ID for Previous Field |
| 7 | PRTA_SRC3_CURR_SOURCE_FID | R | 1h | For Source ID 3 from Port A.. Source Field ID for Current Field |
| 6 | PRTA_SRC3_PREV_SOURCE_FID | R | 1h | For Source ID 3 from Port A.. Source Field ID for Previous Field |
| 5 | PRTA_SRC2_CURR_SOURCE_FID | R | 1h | For Source ID 2 from Port A.. Source Field ID for Current Field |
| 4 | PRTA_SRC2_PREV_SOURCE_FID | R | 1h | For Source ID 2 from Port A.. Source Field ID for Previous Field |
| 3 | PRTA_SRC1_CURR_SOURCE_FID | R | 1h | For Source ID 1 from Port A.. Source Field ID for Current Field |
| 2 | PRTA_SRC1_PREV_SOURCE_FID | R | 1h | For Source ID 1 from Port A.. Source Field ID for Previous Field |
| 1 | PRTA_SRC0_CURR_SOURCE_FID | R | 1h | For Source ID 0 from Port A.. Source Field ID for Current Field |
| 0 | PRTA_SRC0_PREV_SOURCE_FID | R | 1h | For Source ID 0 from Port A.. Source Field ID for Previous Field |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.10 VIP_PARSER_output_port_a_enc_fid Register (offset = 24h) [reset = FFFFFFFFh]

VIP_PARSER_output_port_a_enc_fid is shown in Figure 12-567 and described in Table 12-486.

Current and Previous Output Port A Encoder FID values

**Figure 12-567. VIP_PARSER_output_port_a_enc_fid Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC15_CURR_ENC_FID | PRTA_SRC15_PREV_ENC_FID | PRTA_SRC14_CURR_ENC_FID | PRTA_SRC14_PREV_ENC_FID | PRTA_SRC13_CURR_ENC_FID | PRTA_SRC13_PREV_ENC_FID | PRTA_SRC12_CURR_ENC_FID | PRTA_SRC12_PREV_ENC_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PRTA_SRC11_CURR_ENC_FID | PRTA_SRC11_PREV_ENC_FID | PRTA_SRC10_CURR_ENC_FID | PRTA_SRC10_PREV_ENC_FID | PRTA_SRC9_CURR_ENC_FID | PRTA_SRC9_PREV_ENC_FID | PRTA_SRC8_CURR_ENC_FID | PRTA_SRC8_PREV_ENC_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PRTA_SRC7_CURR_ENC_FID | PRTA_SRC7_PREV_ENC_FID | PRTA_SRC6_CURR_ENC_FID | PRTA_SRC6_PREV_ENC_FID | PRTA_SRC5_CURR_ENC_FID | PRTA_SRC5_PREV_ENC_FID | PRTA_SRC4_CURR_ENC_FID | PRTA_SRC4_PREV_ENC_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRTA_SRC3_CURR_ENC_FID | PRTA_SRC3_PREV_ENC_FID | PRTA_SRC2_CURR_ENC_FID | PRTA_SRC2_PREV_ENC_FID | PRTA_SRC1_CURR_ENC_FID | PRTA_SRC1_PREV_ENC_FID | PRTA_SRC0_CURR_ENC_FID | PRTA_SRC0_PREV_ENC_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-486. VIP_PARSER_output_port_a_enc_fid Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PRTA_SRC15_CURR_ENC_FID | R | 1h | For Source ID 15 from Port A.. Encoder Field ID for Current Field |
| 30 | PRTA_SRC15_PREV_ENC_FID | R | 1h | For Source ID 15 from Port A.. Encoder Field ID for Previous Field |
| 29 | PRTA_SRC14_CURR_ENC_FID | R | 1h | For Source ID 14 from Port A.. Encoder Field ID for Current Field |
| 28 | PRTA_SRC14_PREV_ENC_FID | R | 1h | For Source ID 14 from Port A.. Encoder Field ID for Previous Field |
| 27 | PRTA_SRC13_CURR_ENC_FID | R | 1h | For Source ID 13 from Port A.. Encoder Field ID for Current Field |
| 26 | PRTA_SRC13_PREV_ENC_FID | R | 1h | For Source ID 13 from Port A.. Encoder Field ID for Previous Field |
| 25 | PRTA_SRC12_CURR_ENC_FID | R | 1h | For Source ID 12 from Port A.. Encoder Field ID for Current Field |
| 24 | PRTA_SRC12_PREV_ENC_FID | R | 1h | For Source ID 12 from Port A.. Encoder Field ID for Previous Field |
| 23 | PRTA_SRC11_CURR_ENC_FID | R | 1h | For Source ID 11 from Port A.. Encoder Field ID for Current Field |
| 22 | PRTA_SRC11_PREV_ENC_FID | R | 1h | For Source ID 11 from Port A.. Encoder Field ID for Previous Field |
| 21 | PRTA_SRC10_CURR_ENC_FID | R | 1h | For Source ID 10 from Port A.. Encoder Field ID for Current Field |
| 20 | PRTA_SRC10_PREV_ENC_FID | R | 1h | For Source ID 10 from Port A.. Encoder Field ID for Previous Field |
| 19 | PRTA_SRC9_CURR_ENC_FID | R | 1h | For Source ID 9 from Port A.. Encoder Field ID for Current Field |
| 18 | PRTA_SRC9_PREV_ENC_FID | R | 1h | For Source ID 9 from Port A.. Encoder Field ID for Previous Field |
| 17 | PRTA_SRC8_CURR_ENC_FID | R | 1h | For Source ID 8 from Port A.. Encoder Field ID for Current Field |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-486. VIP_PARSER_output_port_a_enc_fid Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | PRTA_SRC8_PREV_ENC_FID | R | 1h | For Source ID 8 from Port A.. Encoder Field ID for Previous Field |
| 15 | PRTA_SRC7_CURR_ENC_FID | R | 1h | For Source ID 7 from Port A.. Encoder Field ID for Current Field |
| 14 | PRTA_SRC7_PREV_ENC_FID | R | 1h | For Source ID 7 from Port A.. Encoder Field ID for Previous Field |
| 13 | PRTA_SRC6_CURR_ENC_FID | R | 1h | For Source ID 6 from Port A.. Encoder Field ID for Current Field |
| 12 | PRTA_SRC6_PREV_ENC_FID | R | 1h | For Source ID 6 from Port A.. Encoder Field ID for Previous Field |
| 11 | PRTA_SRC5_CURR_ENC_FID | R | 1h | For Source ID 5 from Port A.. Encoder Field ID for Current Field |
| 10 | PRTA_SRC5_PREV_ENC_FID | R | 1h | For Source ID 5 from Port A.. Encoder Field ID for Previous Field |
| 9 | PRTA_SRC4_CURR_ENC_FID | R | 1h | For Source ID 4 from Port A.. Encoder Field ID for Current Field |
| 8 | PRTA_SRC4_PREV_ENC_FID | R | 1h | For Source ID 4 from Port A.. Encoder Field ID for Previous Field |
| 7 | PRTA_SRC3_CURR_ENC_FID | R | 1h | For Source ID 3 from Port A.. Encoder Field ID for Current Field |
| 6 | PRTA_SRC3_PREV_ENC_FID | R | 1h | For Source ID 3 from Port A.. Encoder Field ID for Previous Field |
| 5 | PRTA_SRC2_CURR_ENC_FID | R | 1h | For Source ID 2 from Port A.. Encoder Field ID for Current Field |
| 4 | PRTA_SRC2_PREV_ENC_FID | R | 1h | For Source ID 2 from Port A.. Encoder Field ID for Previous Field |
| 3 | PRTA_SRC1_CURR_ENC_FID | R | 1h | For Source ID 1 from Port A.. Encoder Field ID for Current Field |
| 2 | PRTA_SRC1_PREV_ENC_FID | R | 1h | For Source ID 1 from Port A.. Encoder Field ID for Previous Field |
| 1 | PRTA_SRC0_CURR_ENC_FID | R | 1h | For Source ID 0 from Port A.. Encoder Field ID for Current Field |
| 0 | PRTA_SRC0_PREV_ENC_FID | R | 1h | For Source ID 0 from Port A.. Encoder Field ID for Previous Field |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.11 VIP_PARSER_output_port_b_src_fid Register (offset = 28h) [reset = FFFFFFFFh]

VIP_PARSER_output_port_b_src_fid is shown in Figure 12-568 and described in Table 12-487.

Current and Previous Output Port B Source FID values

**Figure 12-568. VIP_PARSER_output_port_b_src_fid Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC15_CURR_SOURCE_FID | PRTB_SRC15_PREV_SOURCE_FID | PRTB_SRC14_CURR_SOURCE_FID | PRTB_SRC14_PREV_SOURCE_FID | PRTB_SRC13_CURR_SOURCE_FID | PRTB_SRC13_PREV_SOURCE_FID | PRTB_SRC12_CURR_SOURCE_FID | PRTB_SRC12_PREV_SOURCE_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PRTB_SRC11_CURR_SOURCE_FID | PRTB_SRC11_PREV_SOURCE_FID | PRTB_SRC10_CURR_SOURCE_FID | PRTB_SRC10_PREV_SOURCE_FID | PRTB_SRC9_CURR_SOURCE_FID | PRTB_SRC9_PREV_SOURCE_FID | PRTB_SRC8_CURR_SOURCE_FID | PRTB_SRC8_PREV_SOURCE_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PRTB_SRC7_CURR_SOURCE_FID | PRTB_SRC7_PREV_SOURCE_FID | PRTB_SRC6_CURR_SOURCE_FID | PRTB_SRC6_PREV_SOURCE_FID | PRTB_SRC5_CURR_SOURCE_FID | PRTB_SRC5_PREV_SOURCE_FID | PRTB_SRC4_CURR_SOURCE_FID | PRTB_SRC4_PREV_SOURCE_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRTB_SRC3_CURR_SOURCE_FID | PRTB_SRC3_PREV_SOURCE_FID | PRTB_SRC2_CURR_SOURCE_FID | PRTB_SRC2_PREV_SOURCE_FID | PRTB_SRC1_CURR_SOURCE_FID | PRTB_SRC1_PREV_SOURCE_FID | PRTB_SRC0_CURR_SOURCE_FID | PRTB_SRC0_PREV_SOURCE_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-487. VIP_PARSER_output_port_b_src_fid Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PRTB_SRC15_CURR_SOURCE_FID | R | 1h | For Source ID 15 from Port B.. Source Field ID for Current Field |
| 30 | PRTB_SRC15_PREV_SOURCE_FID | R | 1h | For Source ID 15 from Port B.. Source Field ID for Previous Field |
| 29 | PRTB_SRC14_CURR_SOURCE_FID | R | 1h | For Source ID 14 from Port B.. Source Field ID for Current Field |
| 28 | PRTB_SRC14_PREV_SOURCE_FID | R | 1h | For Source ID 14 from Port B.. Source Field ID for Previous Field |
| 27 | PRTB_SRC13_CURR_SOURCE_FID | R | 1h | For Source ID 13 from Port B.. Source Field ID for Current Field |
| 26 | PRTB_SRC13_PREV_SOURCE_FID | R | 1h | For Source ID 13 from Port B.. Source Field ID for Previous Field |
| 25 | PRTB_SRC12_CURR_SOURCE_FID | R | 1h | For Source ID 12 from Port B.. Source Field ID for Current Field |
| 24 | PRTB_SRC12_PREV_SOURCE_FID | R | 1h | For Source ID 12 from Port B.. Source Field ID for Previous Field |
| 23 | PRTB_SRC11_CURR_SOURCE_FID | R | 1h | For Source ID 11 from Port B.. Source Field ID for Current Field |
| 22 | PRTB_SRC11_PREV_SOURCE_FID | R | 1h | For Source ID 11.. from Port B Source Field ID for Previous Field |
| 21 | PRTB_SRC10_CURR_SOURCE_FID | R | 1h | For Source ID 10 from Port B.. Source Field ID for Current Field |
| 20 | PRTB_SRC10_PREV_SOURCE_FID | R | 1h | For Source ID 10 from Port B.. Source Field ID for Previous Field |
| 19 | PRTB_SRC9_CURR_SOURCE_FID | R | 1h | For Source ID 9 from Port B.. Source Field ID for Current Field |
| 18 | PRTB_SRC9_PREV_SOURCE_FID | R | 1h | For Source ID 9 from Port B.. Source Field ID for Previous Field |
| 17 | PRTB_SRC8_CURR_SOURCE_FID | R | 1h | For Source ID 8 from Port B.. Source Field ID for Current Field |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-487. VIP_PARSER_output_port_b_src_fid Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | PRTB_SRC8_PREV_SOURCE_FID | R | 1h | For Source ID 8 from Port B.. Source Field ID for Previous Field |
| 15 | PRTB_SRC7_CURR_SOURCE_FID | R | 1h | For Source ID 7 from Port B.. Source Field ID for Current Field |
| 14 | PRTB_SRC7_PREV_SOURCE_FID | R | 1h | For Source ID 7 from Port B.. Source Field ID for Previous Field |
| 13 | PRTB_SRC6_CURR_SOURCE_FID | R | 1h | For Source ID 6 from Port B.. Source Field ID for Current Field |
| 12 | PRTB_SRC6_PREV_SOURCE_FID | R | 1h | For Source ID 6 from Port B.. Source Field ID for Previous Field |
| 11 | PRTB_SRC5_CURR_SOURCE_FID | R | 1h | For Source ID 5 from Port B.. Source Field ID for Current Field |
| 10 | PRTB_SRC5_PREV_SOURCE_FID | R | 1h | For Source ID 5 from Port B.. Source Field ID for Previous Field |
| 9 | PRTB_SRC4_CURR_SOURCE_FID | R | 1h | For Source ID 4 from Port B.. Source Field ID for Current Field |
| 8 | PRTB_SRC4_PREV_SOURCE_FID | R | 1h | For Source ID 4 from Port B.. Source Field ID for Previous Field |
| 7 | PRTB_SRC3_CURR_SOURCE_FID | R | 1h | For Source ID 3 from Port B.. Source Field ID for Current Field |
| 6 | PRTB_SRC3_PREV_SOURCE_FID | R | 1h | For Source ID 3 from Port B.. Source Field ID for Previous Field |
| 5 | PRTB_SRC2_CURR_SOURCE_FID | R | 1h | For Source ID 2 from Port B.. Source Field ID for Current Field |
| 4 | PRTB_SRC2_PREV_SOURCE_FID | R | 1h | For Source ID 2 from Port B.. Source Field ID for Previous Field |
| 3 | PRTB_SRC1_CURR_SOURCE_FID | R | 1h | For Source ID 1 from Port B.. Source Field ID for Current Field |
| 2 | PRTB_SRC1_PREV_SOURCE_FID | R | 1h | For Source ID 1 from Port B.. Source Field ID for Previous Field |
| 1 | PRTB_SRC0_CURR_SOURCE_FID | R | 1h | For Source ID 0 from Port B.. Source Field ID for Current Field |
| 0 | PRTB_SRC0_PREV_SOURCE_FID | R | 1h | For Source ID 0 from Port B.. Source Field ID for Previous Field |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.12 VIP_PARSER_output_port_b_enc_fid Register (offset = 2Ch) [reset = FFFFFFFFh]

VIP_PARSER_output_port_b_enc_fid is shown in Figure 12-569 and described in Table 12-488.

Current and Previous Output Port B Encoder FID values

**Figure 12-569. VIP_PARSER_output_port_b_enc_fid Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC15_CURR_ENC_FID | PRTB_SRC15_PREV_ENC_FID | PRTB_SRC14_CURR_ENC_FID | PRTB_SRC14_PREV_ENC_FID | PRTB_SRC13_CURR_ENC_FID | PRTB_SRC13_PREV_ENC_FID | PRTB_SRC12_CURR_ENC_FID | PRTB_SRC12_PREV_ENC_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PRTB_SRC11_CURR_ENC_FID | PRTB_SRC11_PREV_ENC_FID | PRTB_SRC10_CURR_ENC_FID | PRTB_SRC10_PREV_ENC_FID | PRTB_SRC9_CURR_ENC_FID | PRTB_SRC9_PREV_ENC_FID | PRTB_SRC8_CURR_ENC_FID | PRTB_SRC8_PREV_ENC_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PRTB_SRC7_CURR_ENC_FID | PRTB_SRC7_PREV_ENC_FID | PRTB_SRC6_CURR_ENC_FID | PRTB_SRC6_PREV_ENC_FID | PRTB_SRC5_CURR_ENC_FID | PRTB_SRC5_PREV_ENC_FID | PRTB_SRC4_CURR_ENC_FID | PRTB_SRC4_PREV_ENC_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRTB_SRC3_CURR_ENC_FID | PRTB_SRC3_PREV_ENC_FID | PRTB_SRC2_CURR_ENC_FID | PRTB_SRC2_PREV_ENC_FID | PRTB_SRC1_CURR_ENC_FID | PRTB_SRC1_PREV_ENC_FID | PRTB_SRC0_CURR_ENC_FID | PRTB_SRC0_PREV_ENC_FID |
| R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-488. VIP_PARSER_output_port_b_enc_fid Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | PRTB_SRC15_CURR_ENC_FID | R | 1h | For Source ID 15 from Port B.. Encoder Field ID for Current Field |
| 30 | PRTB_SRC15_PREV_ENC_FID | R | 1h | For Source ID 15 from Port B.. Encoder Field ID for Previous Field |
| 29 | PRTB_SRC14_CURR_ENC_FID | R | 1h | For Source ID 14 from Port B.. Encoder Field ID for Current Field |
| 28 | PRTB_SRC14_PREV_ENC_FID | R | 1h | For Source ID 14 from Port B.. Encoder Field ID for Previous Field |
| 27 | PRTB_SRC13_CURR_ENC_FID | R | 1h | For Source ID 13 from Port B.. Encoder Field ID for Current Field |
| 26 | PRTB_SRC13_PREV_ENC_FID | R | 1h | For Source ID 13 from Port B.. Encoder Field ID for Previous Field |
| 25 | PRTB_SRC12_CURR_ENC_FID | R | 1h | For Source ID 12 from Port B.. Encoder Field ID for Current Field |
| 24 | PRTB_SRC12_PREV_ENC_FID | R | 1h | For Source ID 12 from Port B.. Encoder Field ID for Previous Field |
| 23 | PRTB_SRC11_CURR_ENC_FID | R | 1h | For Source ID 11 from Port B.. Encoder Field ID for Current Field |
| 22 | PRTB_SRC11_PREV_ENC_FID | R | 1h | For Source ID 11 from Port B.. Encoder Field ID for Previous Field |
| 21 | PRTB_SRC10_CURR_ENC_FID | R | 1h | For Source ID 10 from Port B.. Encoder Field ID for Current Field |
| 20 | PRTB_SRC10_PREV_ENC_FID | R | 1h | For Source ID 10 from Port B.. Encoder Field ID for Previous Field |
| 19 | PRTB_SRC9_CURR_ENC_FID | R | 1h | For Source ID 9 from Port B.. Encoder Field ID for Current Field |
| 18 | PRTB_SRC9_PREV_ENC_FID | R | 1h | For Source ID 9 from Port B.. Encoder Field ID for Previous Field |
| 17 | PRTB_SRC8_CURR_ENC_FID | R | 1h | For Source ID 8 from Port B.. Encoder Field ID for Current Field |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 12-488. VIP_PARSER_output_port_b_enc_fid Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | PRTB_SRC8_PREV_ENC_FID | R | 1h | For Source ID 8 from Port B.. Encoder Field ID for Previous Field |
| 15 | PRTB_SRC7_CURR_ENC_FID | R | 1h | For Source ID 7 from Port B.. Encoder Field ID for Current Field |
| 14 | PRTB_SRC7_PREV_ENC_FID | R | 1h | For Source ID 7 from Port B.. Encoder Field ID for Previous Field |
| 13 | PRTB_SRC6_CURR_ENC_FID | R | 1h | For Source ID 6 from Port B.. Encoder Field ID for Current Field |
| 12 | PRTB_SRC6_PREV_ENC_FID | R | 1h | For Source ID 6 from Port B.. Encoder Field ID for Previous Field |
| 11 | PRTB_SRC5_CURR_ENC_FID | R | 1h | For Source ID 5 from Port B.. Encoder Field ID for Current Field |
| 10 | PRTB_SRC5_PREV_ENC_FID | R | 1h | For Source ID 5 from Port B.. Encoder Field ID for Previous Field |
| 9 | PRTB_SRC4_CURR_ENC_FID | R | 1h | For Source ID 4 from Port B.. Encoder Field ID for Current Field |
| 8 | PRTB_SRC4_PREV_ENC_FID | R | 1h | For Source ID 4 from Port B.. Encoder Field ID for Previous Field |
| 7 | PRTB_SRC3_CURR_ENC_FID | R | 1h | For Source ID 3 from Port B.. Encoder Field ID for Current Field |
| 6 | PRTB_SRC3_PREV_ENC_FID | R | 1h | For Source ID 3 from Port B.. Encoder Field ID for Previous Field |
| 5 | PRTB_SRC2_CURR_ENC_FID | R | 1h | For Source ID 2 from Port B.. Encoder Field ID for Current Field |
| 4 | PRTB_SRC2_PREV_ENC_FID | R | 1h | For Source ID 2 from Port B.. Encoder Field ID for Previous Field |
| 3 | PRTB_SRC1_CURR_ENC_FID | R | 1h | For Source ID 1 from Port B.. Encoder Field ID for Current Field |
| 2 | PRTB_SRC1_PREV_ENC_FID | R | 1h | For Source ID 1 from Port B.. Encoder Field ID for Previous Field |
| 1 | PRTB_SRC0_CURR_ENC_FID | R | 1h | For Source ID 0 from Port B.. Encoder Field ID for Current Field |
| 0 | PRTB_SRC0_PREV_ENC_FID | R | 1h | For Source ID 0 from Port B.. Encoder Field ID for Previous Field |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.13 VIP_PARSER_output_port_a_src0_size Register (offset = 30h) [reset = 0h]

VIP_PARSER_output_port_a_src0_size is shown in Figure 12-570 and described in Table 12-489.

Width and Height for Source 0

**Figure 12-570. VIP_PARSER_output_port_a_src0_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC0_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC0_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC0_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC0_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-489. VIP_PARSER_output_port_a_src0_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC0_WIDTH | R | 0h | On Port A.. Width of Source ID 0 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC0_HEIGHT | R | 0h | On Port A.. Height of Source ID 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.14 VIP_PARSER_output_port_a_src1_size Register (offset = 34h) [reset = 0h]

VIP_PARSER_output_port_a_src1_size is shown in Figure 12-571 and described in Table 12-490.

Width and Height for Source 1

**Figure 12-571. VIP_PARSER_output_port_a_src1_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC1_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC1_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC1_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC1_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-490. VIP_PARSER_output_port_a_src1_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC1_WIDTH | R | 0h | On Port A.. Width of Source ID 1 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC1_HEIGHT | R | 0h | On Port A.. Height of Source ID 1 |

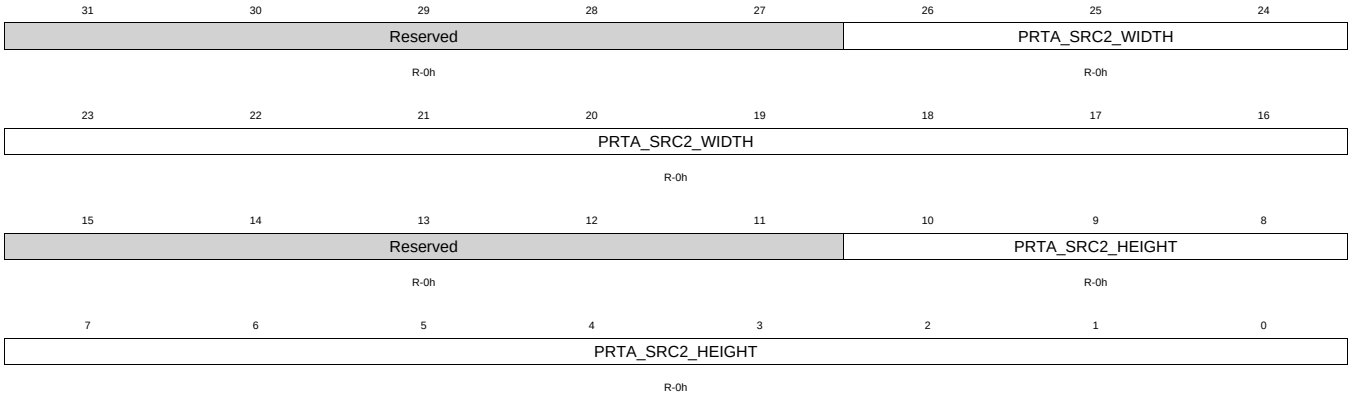

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.15 VIP_PARSER_output_port_a_src2_size Register (offset = 38h) [reset = 0h]

VIP_PARSER_output_port_a_src2_size is shown in Figure 12-572 and described in Table 12-491.

Width and Height for Source 2

**Figure 12-572. VIP_PARSER_output_port_a_src2_size Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-491. VIP_PARSER_output_port_a_src2_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC2_WIDTH | R | 0h | On Port A.. Width of Source ID 2 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC2_HEIGHT | R | 0h | On Port A.. Height of Source ID 2 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.16 VIP_PARSER_output_port_a_src3_size Register (offset = 3Ch) [reset = 0h]

VIP_PARSER_output_port_a_src3_size is shown in Figure 12-573 and described in Table 12-492.

Width and Height for Source 3

**Figure 12-573. VIP_PARSER_output_port_a_src3_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC3_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC3_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC3_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC3_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-492. VIP_PARSER_output_port_a_src3_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC3_WIDTH | R | 0h | On Port A.. Width of Source ID 3 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC3_HEIGHT | R | 0h | On Port A.. Height of Source ID 3 |



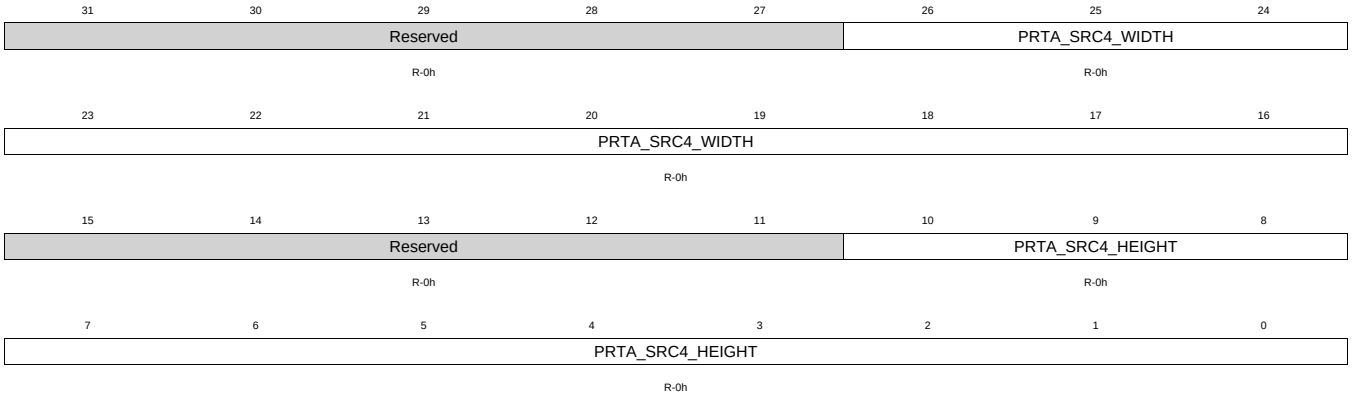
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.17 VIP_PARSER_output_port_a_src4_size Register (offset = 40h) [reset = 0h]

VIP_PARSER_output_port_a_src4_size is shown in Figure 12-574 and described in Table 12-493.

Width and Height for Source 4

**Figure 12-574. VIP_PARSER_output_port_a_src4_size Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-493. VIP_PARSER_output_port_a_src4_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC4_WIDTH | R | 0h | On Port A.. Width of Source ID 4 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC4_HEIGHT | R | 0h | On Port A.. Height of Source ID 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.18 VIP_PARSER_output_port_a_src5_size Register (offset = 44h) [reset = 0h]

VIP_PARSER_output_port_a_src5_size is shown in Figure 12-575 and described in Table 12-494.

Width and Height for Source 5

**Figure 12-575. VIP_PARSER_output_port_a_src5_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC5_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC5_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC5_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC5_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-494. VIP_PARSER_output_port_a_src5_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC5_WIDTH | R | 0h | On Port A.. Width of Source ID 5 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC5_HEIGHT | R | 0h | On Port A.. Height of Source ID 5 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.19 VIP_PARSER_output_port_a_src6_size Register (offset = 48h) [reset = 0h]

VIP_PARSER_output_port_a_src6_size is shown in Figure 12-576 and described in Table 12-495.

Width and Height for Source 6

**Figure 12-576. VIP_PARSER_output_port_a_src6_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC6_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC6_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC6_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC6_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-495. VIP_PARSER_output_port_a_src6_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC6_WIDTH | R | 0h | On Port A.. Width of Source ID 6 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC6_HEIGHT | R | 0h | On Port A.. Height of Source ID 6 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.20 VIP_PARSER_output_port_a_src7_size Register (offset = 4Ch) [reset = 0h]

VIP_PARSER_output_port_a_src7_size is shown in Figure 12-577 and described in Table 12-496.

Width and Height for Source 7

**Figure 12-577. VIP_PARSER_output_port_a_src7_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC7_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC7_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC7_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC7_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-496. VIP_PARSER_output_port_a_src7_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC7_WIDTH | R | 0h | On Port A.. Width of Source ID 7 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC7_HEIGHT | R | 0h | On Port A.. Height of Source ID 7 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.21 VIP_PARSER_output_port_a_src8_size Register (offset = 50h) [reset = 0h]

VIP_PARSER_output_port_a_src8_size is shown in Figure 12-578 and described in Table 12-497.

Width and Height for Source 8

**Figure 12-578. VIP_PARSER_output_port_a_src8_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC8_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC8_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC8_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC8_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-497. VIP_PARSER_output_port_a_src8_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC8_WIDTH | R | 0h | On Port A.. Width of Source ID 8 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC8_HEIGHT | R | 0h | On Port A.. Height of Source ID 8 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.22 VIP_PARSER_output_port_a_src9_size Register (offset = 54h) [reset = 0h]

VIP_PARSER_output_port_a_src9_size is shown in Figure 12-579 and described in Table 12-498.

Width and Height for Source 9

**Figure 12-579. VIP_PARSER_output_port_a_src9_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC9_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC9_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC9_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC9_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-498. VIP_PARSER_output_port_a_src9_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC9_WIDTH | R | 0h | On Port A.. Width of Source ID 9 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC9_HEIGHT | R | 0h | On Port A.. Height of Source ID 9 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.23 VIP_PARSER_output_port_a_src10_size Register (offset = 58h) [reset = 0h]

VIP_PARSER_output_port_a_src10_size is shown in Figure 12-580 and described in Table 12-499.

Width and Height for Source 10

**Figure 12-580. VIP_PARSER_output_port_a_src10_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC10_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC10_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC10_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC10_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-499. VIP_PARSER_output_port_a_src10_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC10_WIDTH | R | 0h | On Port A.. Width of Source ID 10 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC10_HEIGHT | R | 0h | On Port A.. Height of Source ID 10 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.24 VIP_PARSER_output_port_a_src11_size Register (offset = 5Ch) [reset = 0h]

VIP_PARSER_output_port_a_src11_size is shown in Figure 12-581 and described in Table 12-500.

Width and Height for Source 11

**Figure 12-581. VIP_PARSER_output_port_a_src11_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC11_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC11_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC11_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC11_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-500. VIP_PARSER_output_port_a_src11_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC11_WIDTH | R | 0h | On Port A.. Width of Source ID 11 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC11_HEIGHT | R | 0h | On Port A.. Height of Source ID 11 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.25 VIP_PARSER_output_port_a_src12_size Register (offset = 60h) [reset = 0h]

VIP_PARSER_output_port_a_src12_size is shown in Figure 12-582 and described in Table 12-501.

Width and Height for Source 12

**Figure 12-582. VIP_PARSER_output_port_a_src12_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC12_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC12_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC12_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC12_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-501. VIP_PARSER_output_port_a_src12_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC12_WIDTH | R | 0h | On Port A.. Width of Source ID 12 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC12_HEIGHT | R | 0h | On Port A.. Height of Source ID 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.26 VIP_PARSER_output_port_a_src13_size Register (offset = 64h) [reset = 0h]

VIP_PARSER_output_port_a_src13_size is shown in Figure 12-583 and described in Table 12-502.

Width and Height for Source 13

**Figure 12-583. VIP_PARSER_output_port_a_src13_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC13_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC13_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC13_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC13_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-502. VIP_PARSER_output_port_a_src13_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC13_WIDTH | R | 0h | On Port A.. Width of Source ID 13 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC13_HEIGHT | R | 0h | On Port A.. Height of Source ID 13 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.27 VIP_PARSER_output_port_a_src14_size Register (offset = 68h) [reset = 0h]

VIP_PARSER_output_port_a_src14_size is shown in Figure 12-584 and described in Table 12-503.

Width and Height for Source 14

**Figure 12-584. VIP_PARSER_output_port_a_src14_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC14_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC14_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC14_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC14_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-503. VIP_PARSER_output_port_a_src14_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC14_WIDTH | R | 0h | On Port A.. Width of Source ID 14 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC14_HEIGHT | R | 0h | On Port A.. Height of Source ID 14 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.28 VIP_PARSER_output_port_a_src15_size Register (offset = 6Ch) [reset = 0h]

VIP_PARSER_output_port_a_src15_size is shown in Figure 12-585 and described in Table 12-504.

Width and Height for Source 15

**Figure 12-585. VIP_PARSER_output_port_a_src15_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC15_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC15_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTA_SRC15_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTA_SRC15_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-504. VIP_PARSER_output_port_a_src15_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTA_SRC15_WIDTH | R | 0h | On Port A.. Width of Source ID 15 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTA_SRC15_HEIGHT | R | 0h | On Port A.. Height of Source ID 15 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.29 VIP_PARSER_output_port_b_src0_size Register (offset = 70h) [reset = 0h]

VIP_PARSER_output_port_b_src0_size is shown in Figure 12-586 and described in Table 12-505.

Width and Height for Source 0

**Figure 12-586. VIP_PARSER_output_port_b_src0_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC0_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC0_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC0_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC0_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-505. VIP_PARSER_output_port_b_src0_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC0_WIDTH | R | 0h | On Port B.. Width of Source ID 0 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC0_HEIGHT | R | 0h | On Port B.. Height of Source ID 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.30 VIP_PARSER_output_port_b_src1_size Register (offset = 74h) [reset = 0h]

VIP_PARSER_output_port_b_src1_size is shown in Figure 12-587 and described in Table 12-506.

Width and Height for Source 1

**Figure 12-587. VIP_PARSER_output_port_b_src1_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC1_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC1_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC1_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC1_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-506. VIP_PARSER_output_port_b_src1_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC1_WIDTH | R | 0h | On Port B.. Width of Source ID 1 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC1_HEIGHT | R | 0h | On Port B.. Height of Source ID 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.31 VIP_PARSER_output_port_b_src2_size Register (offset = 78h) [reset = 0h]

VIP_PARSER_output_port_b_src2_size is shown in Figure 12-588 and described in Table 12-507.

Width and Height for Source 2

**Figure 12-588. VIP_PARSER_output_port_b_src2_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC2_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC2_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC2_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC2_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-507. VIP_PARSER_output_port_b_src2_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC2_WIDTH | R | 0h | On Port B.. Width of Source ID 2 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC2_HEIGHT | R | 0h | On Port B.. Height of Source ID 2 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.32 VIP_PARSER_output_port_b_src3_size Register (offset = 7Ch) [reset = 0h]

VIP_PARSER_output_port_b_src3_size is shown in Figure 12-589 and described in Table 12-508.

Width and Height for Source 3

**Figure 12-589. VIP_PARSER_output_port_b_src3_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC3_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC3_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC3_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC3_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-508. VIP_PARSER_output_port_b_src3_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC3_WIDTH | R | 0h | On Port B.. Width of Source ID 3 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC3_HEIGHT | R | 0h | On Port B.. Height of Source ID 3 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.33 VIP_PARSER_output_port_b_src4_size Register (offset = 80h) [reset = 0h]

VIP_PARSER_output_port_b_src4_size is shown in Figure 12-590 and described in Table 12-509.

Width and Height for Source 4

**Figure 12-590. VIP_PARSER_output_port_b_src4_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC4_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC4_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC4_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC4_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-509. VIP_PARSER_output_port_b_src4_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC4_WIDTH | R | 0h | On Port B.. Width of Source ID 4 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC4_HEIGHT | R | 0h | On Port B.. Height of Source ID 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.34 VIP_PARSER_output_port_b_src5_size Register (offset = 84h) [reset = 0h]

VIP_PARSER_output_port_b_src5_size is shown in Figure 12-591 and described in Table 12-510.

Width and Height for Source 5

**Figure 12-591. VIP_PARSER_output_port_b_src5_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC5_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC5_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC5_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC5_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-510. VIP_PARSER_output_port_b_src5_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC5_WIDTH | R | 0h | On Port B.. Width of Source ID 5 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC5_HEIGHT | R | 0h | On Port B.. Height of Source ID 5 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.35 VIP_PARSER_output_port_b_src6_size Register (offset = 88h) [reset = 0h]

VIP_PARSER_output_port_b_src6_size is shown in Figure 12-592 and described in Table 12-511.

Width and Height for Source 6

**Figure 12-592. VIP_PARSER_output_port_b_src6_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC6_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC6_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC6_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC6_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-511. VIP_PARSER_output_port_b_src6_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC6_WIDTH | R | 0h | On Port B.. Width of Source ID 6 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC6_HEIGHT | R | 0h | On Port B.. Height of Source ID 6 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.36 VIP_PARSER_output_port_b_src7_size Register (offset = 8Ch) [reset = 0h]

VIP_PARSER_output_port_b_src7_size is shown in Figure 12-593 and described in Table 12-512.

Width and Height for Source 7

**Figure 12-593. VIP_PARSER_output_port_b_src7_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC7_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC7_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC7_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC7_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-512. VIP_PARSER_output_port_b_src7_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC7_WIDTH | R | 0h | On Port B.. Width of Source ID 7 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC7_HEIGHT | R | 0h | On Port B.. Height of Source ID 7 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.37 VIP_PARSER_output_port_b_src8_size Register (offset = 90h) [reset = 0h]

VIP_PARSER_output_port_b_src8_size is shown in Figure 12-594 and described in Table 12-513.

Width and Height for Source 8

**Figure 12-594. VIP_PARSER_output_port_b_src8_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC8_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC8_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC8_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC8_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-513. VIP_PARSER_output_port_b_src8_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC8_WIDTH | R | 0h | On Port B.. Width of Source ID 8 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC8_HEIGHT | R | 0h | On Port B.. Height of Source ID 8 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.38 VIP_PARSER_output_port_b_src9_size Register (offset = 94h) [reset = 0h]

VIP_PARSER_output_port_b_src9_size is shown in Figure 12-595 and described in Table 12-514.

Width and Height for Source 9

**Figure 12-595. VIP_PARSER_output_port_b_src9_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC9_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC9_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC9_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC9_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-514. VIP_PARSER_output_port_b_src9_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC9_WIDTH | R | 0h | On Port B.. Width of Source ID 9 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC9_HEIGHT | R | 0h | On Port B.. Height of Source ID 9 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.39 VIP_PARSER_output_port_b_src10_size Register (offset = 98h) [reset = 0h]

VIP_PARSER_output_port_b_src10_size is shown in Figure 12-596 and described in Table 12-515.

Width and Height for Source 10

**Figure 12-596. VIP_PARSER_output_port_b_src10_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC10_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC10_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC10_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC10_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-515. VIP_PARSER_output_port_b_src10_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC10_WIDTH | R | 0h | On Port B.. Width of Source ID 10 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC10_HEIGHT | R | 0h | On Port B.. Height of Source ID 10 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.40 VIP_PARSER_output_port_b_src11_size Register (offset = 9Ch) [reset = 0h]

VIP_PARSER_output_port_b_src11_size is shown in Figure 12-597 and described in Table 12-516.

Width and Height for Source 11

**Figure 12-597. VIP_PARSER_output_port_b_src11_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC11_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC11_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC11_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC11_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-516. VIP_PARSER_output_port_b_src11_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC11_WIDTH | R | 0h | On Port B.. Width of Source ID 11 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC11_HEIGHT | R | 0h | On Port B.. Height of Source ID 11 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.41 VIP_PARSER_output_port_b_src12_size Register (offset = A0h) [reset = 0h]

VIP_PARSER_output_port_b_src12_size is shown in Figure 12-598 and described in Table 12-517.

Width and Height for Source 12

**Figure 12-598. VIP_PARSER_output_port_b_src12_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC12_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC12_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC12_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC12_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-517. VIP_PARSER_output_port_b_src12_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC12_WIDTH | R | 0h | On Port B.. Width of Source ID 12 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC12_HEIGHT | R | 0h | On Port B.. Height of Source ID 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.42 VIP_PARSER_output_port_b_src13_size Register (offset = A4h) [reset = 0h]

VIP_PARSER_output_port_b_src13_size is shown in Figure 12-599 and described in Table 12-518.

Width and Height for Source 13

**Figure 12-599. VIP_PARSER_output_port_b_src13_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC13_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC13_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC13_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC13_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-518. VIP_PARSER_output_port_b_src13_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC13_WIDTH | R | 0h | On Port B.. Width of Source ID 13 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC13_HEIGHT | R | 0h | On Port B.. Height of Source ID 13 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.43 VIP_PARSER_output_port_b_src14_size Register (offset = A8h) [reset = 0h]

VIP_PARSER_output_port_b_src14_size is shown in Figure 12-600 and described in Table 12-519.

Width and Height for Source 14

**Figure 12-600. VIP_PARSER_output_port_b_src14_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC14_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC14_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC14_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC14_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-519. VIP_PARSER_output_port_b_src14_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC14_WIDTH | R | 0h | On Port B.. Width of Source ID 14 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC14_HEIGHT | R | 0h | On Port B.. Height of Source ID 14 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.44 VIP_PARSER_output_port_b_src15_size Register (offset = ACh) [reset = 0h]

VIP_PARSER_output_port_b_src15_size is shown in Figure 12-601 and described in Table 12-520.

Width and Height for Source 15

**Figure 12-601. VIP_PARSER_output_port_b_src15_size Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC15_WIDTH | | |
| R-0h | | | | | R-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC15_WIDTH | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PRTB_SRC15_HEIGHT | | |
| R-0h | | | | | R-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PRTB_SRC15_HEIGHT | | | | | | | |
| R-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-520. VIP_PARSER_output_port_b_src15_size Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | PRTB_SRC15_WIDTH | R | 0h | On Port B.. Width of Source ID 15 |
| 15-11 | Reserved | R | 0h | |
| 10-0 | PRTB_SRC15_HEIGHT | R | 0h | On Port B.. Height of Source ID 15 |



Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.45 VIP_PARSER_port_a_vdet_vec Register (offset = B0h) [reset = 0h]

VIP_PARSER_port_a_vdet_vec is shown in Figure 12-602 and described in Table 12-521.

Each bit represents the VDET bit setting for Line Mux Mode

**Figure 12-602. VIP_PARSER_port_a_vdet_vec Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PRTA_VDET_VEC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-521. VIP_PARSER_port_a_vdet_vec Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | PRTA_VDET_VEC | R | 0h | For Embedded Sync Only In Line Mux Mode.. each bit represents the vdet value on Port A for the corresponding source id. This vector is meaningless for 1x/2x/4x mux modes. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.46 VIP_PARSER_port_b_vdet_vec Register (offset = B4h) [reset = 0h]

VIP_PARSER_port_b_vdet_vec is shown in Figure 12-603 and described in Table 12-522.

Each bit represents the VDET bit setting for Line Mux Mode

**Figure 12-603. VIP_PARSER_port_b_vdet_vec Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PRTB_VDET_VEC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-522. VIP_PARSER_port_b_vdet_vec Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | PRTB_VDET_VEC | R | 0h | For Embedded Sync Only In Line Mux Mode.. each bit represents the vdet value on Port B for the corresponding source id. This vector is meaningless for 1x/2x/4x mux modes. |

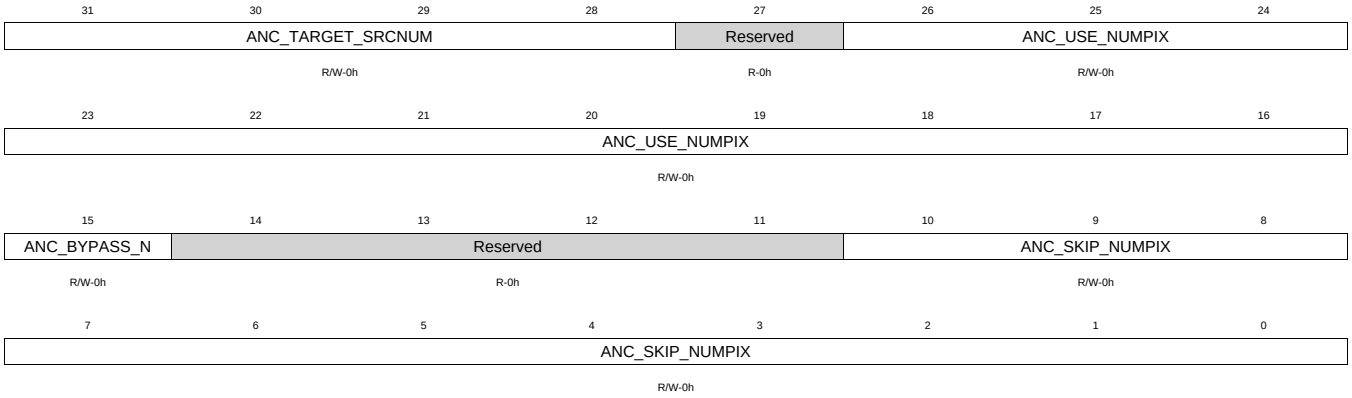

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.47 VIP_PARSER_xtra2_port_a Register (offset = B8h) [reset = 0h]

VIP_PARSER_xtra2_port_a is shown in Figure 12-604 and described in Table 12-523.

Ancillary Cropping Configuration for Input Port A

**Figure 12-604. VIP_PARSER_xtra2_port_a Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-523. VIP_PARSER_xtra2_port_a Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | ANC_TARGET_SRCNUM | R/W | 0h | The cropping module can work on only one srcnum, specified in this field, for each dss_vip_parser output port (Ancillary data). |
| 27 | Reserved | R | 0h | |
| 26-16 | ANC_USE_NUMPIX | R/W | 0h | When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width. |
| 15 | ANC_BYPASS_N | R/W | 0h | 0 : Bypass cropping module<br>1 : Cropping module enabled |
| 14-11 | Reserved | R | 0h | |
| 10-0 | ANC_SKIP_NUMPIX | R/W | 0h | The number of pixels to crop from the beginning of each line. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.48 VIP_PARSER_xtra3_port_a Register (offset = BCh) [reset = 0h]

VIP_PARSER_xtra3_port_a is shown in Figure 12-605 and described in Table 12-524.

Ancillary Cropping Configuration for Input Port A

**Figure 12-605. VIP_PARSER_xtra3_port_a Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | ANC_USE_NUMLINES | | |
| R-0h | | | | | R/W-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ANC_USE_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | ANC_SKIP_NUMLINES | | |
| R-0h | | | | | R/W-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ANC_SKIP_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-524. VIP_PARSER_xtra3_port_a Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | ANC_USE_NUMLINES | R/W | 0h | When cropping, the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnum's ancillary data region. |
| 15-11 | Reserved | R | 0h | |
| 10-0 | ANC_SKIP_NUMLINES | R/W | 0h | The number of lines to crop from the top of the vertical ancillary data region. |



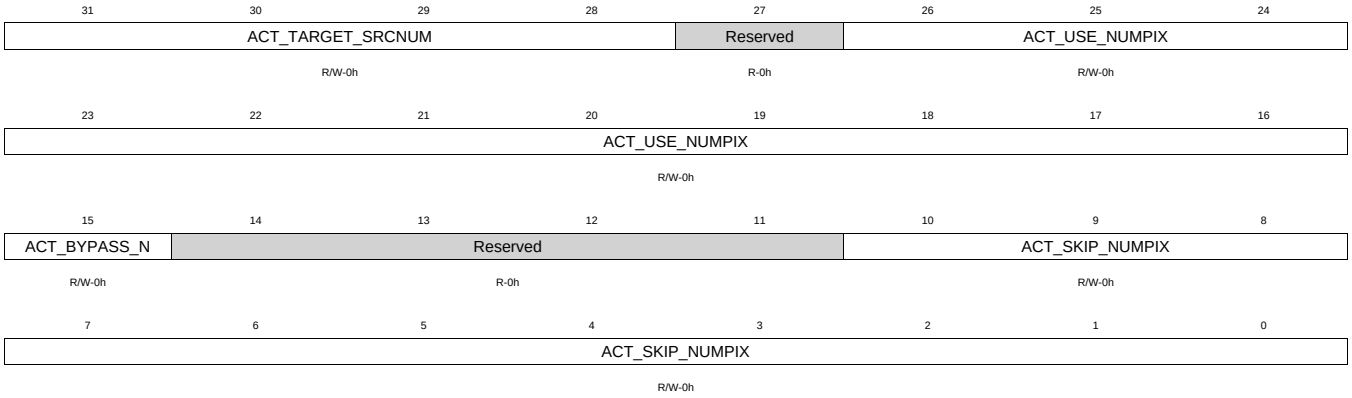
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.49 VIP_PARSER_xtra4_port_a Register (offset = C0h) [reset = 0h]

VIP_PARSER_xtra4_port_a is shown in Figure 12-606 and described in Table 12-525.

Active Video Cropping Configuration for Input Port A

**Figure 12-606. VIP_PARSER_xtra4_port_a Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-525. VIP_PARSER_xtra4_port_a Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | ACT_TARGET_SRCNUM | R/W | 0h | The cropping module can work on only one srcnum, specified in this field, for each dss_vip_parser output port (Active video). |
| 27 | Reserved | R | 0h | |
| 26-16 | ACT_USE_NUMPIX | R/W | 0h | When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width. |
| 15 | ACT_BYPASS_N | R/W | 0h | 0 : Bypass cropping module<br>1 : Cropping module enabled |
| 14-11 | Reserved | R | 0h | |
| 10-0 | ACT_SKIP_NUMPIX | R/W | 0h | The number of pixels to crop from the beginning of each line. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.50 VIP_PARSER_xtra5_port_a Register (offset = C4h) [reset = 0h]

VIP_PARSER_xtra5_port_a is shown in Figure 12-607 and described in Table 12-526.

Active Video Cropping Configuration for Input Port A

**Figure 12-607. VIP_PARSER_xtra5_port_a Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | ACT_USE_NUMLINES | | |
| R-0h | | | | | R/W-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ACT_USE_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | ACT_SKIP_NUMLINES | | |
| R-0h | | | | | R/W-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ACT_SKIP_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-526. VIP_PARSER_xtra5_port_a Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | ACT_USE_NUMLINES | R/W | 0h | When cropping, the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnum's active video region. |
| 15-11 | Reserved | R | 0h | |
| 10-0 | ACT_SKIP_NUMLINES | R/W | 0h | The number of lines to crop from the top of the vertical active video region. |



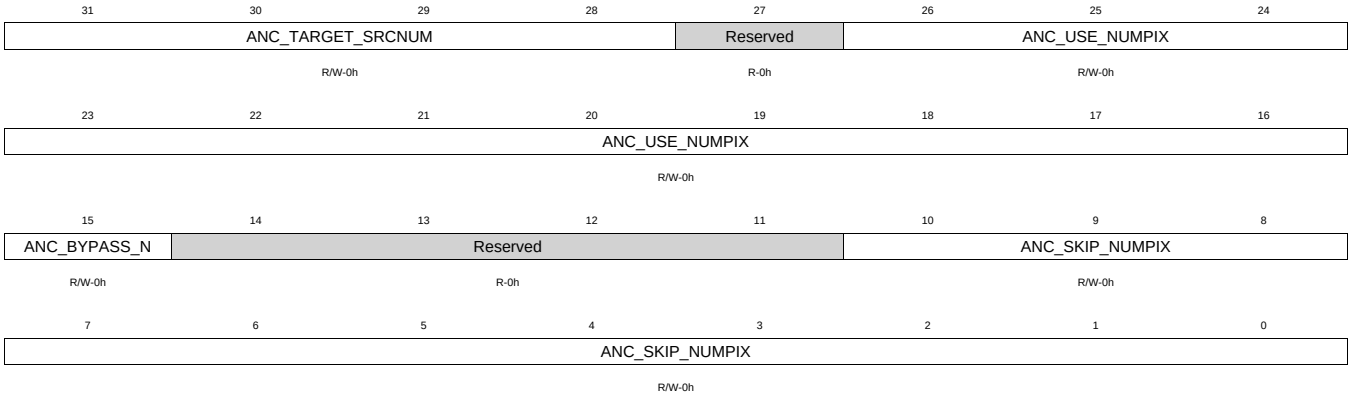
Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.51 VIP_PARSER_xtra2_port_b Register (offset = C8h) [reset = 0h]

VIP_PARSER_xtra2_port_b is shown in Figure 12-608 and described in Table 12-527.

Ancillary Cropping Configuration for Input Port B

**Figure 12-608. VIP_PARSER_xtra2_port_b Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-527. VIP_PARSER_xtra2_port_b Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | ANC_TARGET_SRCNUM | R/W | 0h | The cropping module can work on only one srcnum, specified in this field, for each dss_vip_parser output port (Ancillary data). |
| 27 | Reserved | R | 0h | |
| 26-16 | ANC_USE_NUMPIX | R/W | 0h | When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width. |
| 15 | ANC_BYPASS_N | R/W | 0h | 0 : Bypass cropping module<br>1 : Cropping module enabled |
| 14-11 | Reserved | R | 0h | |
| 10-0 | ANC_SKIP_NUMPIX | R/W | 0h | The number of pixels to crop from the beginning of each line. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.52 VIP_PARSER_xtra3_port_b Register (offset = CCh) [reset = 0h]

VIP_PARSER_xtra3_port_b is shown in Figure 12-609 and described in Table 12-528.

Ancillary Cropping Configuration for Input Port B

**Figure 12-609. VIP_PARSER_xtra3_port_b Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | ANC_USE_NUMLINES | | |
| R-0h | | | | | R/W-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ANC_USE_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | ANC_SKIP_NUMLINES | | |
| R-0h | | | | | R/W-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ANC_SKIP_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-528. VIP_PARSER_xtra3_port_b Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | ANC_USE_NUMLINES | R/W | 0h | When cropping, the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnum's ancillary data region. |
| 15-11 | Reserved | R | 0h | |
| 10-0 | ANC_SKIP_NUMLINES | R/W | 0h | The number of lines to crop from the top of the vertical ancillary data region. |

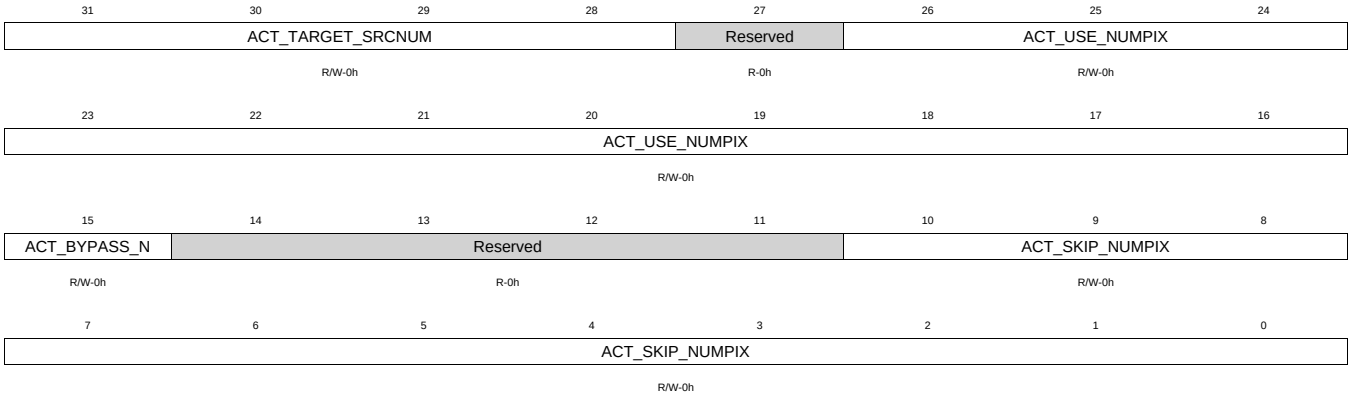

Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.53 VIP_PARSER_xtra4_port_b Register (offset = D0h) [reset = 0h]

VIP_PARSER_xtra4_port_b is shown in Figure 12-610 and described in Table 12-529.

Active Video Cropping Configuration for Input Port B

**Figure 12-610. VIP_PARSER_xtra4_port_b Register**




LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-529. VIP_PARSER_xtra4_port_b Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | ACT_TARGET_SRCNUM | R/W | 0h | The cropping module can work on only one srcnum, specified in this field, for each dss_vip_parser output port (Active video). |
| 27 | Reserved | R | 0h | |
| 26-16 | ACT_USE_NUMPIX | R/W | 0h | When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width. |
| 15 | ACT_BYPASS_N | R/W | 0h | 0 : Bypass cropping module<br>1 : Cropping module enabled |
| 14-11 | Reserved | R | 0h | |
| 10-0 | ACT_SKIP_NUMPIX | R/W | 0h | The number of pixels to crop from the beginning of each line. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.54 VIP_PARSER_xtra5_port_b Register (offset = D4h) [reset = 0h]

VIP_PARSER_xtra5_port_b is shown in Figure 12-611 and described in Table 12-530.

Active Video Cropping Configuration for Input Port B

**Figure 12-611. VIP_PARSER_xtra5_port_b Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | ACT_USE_NUMLINES | | |
| R-0h | | | | | R/W-0h | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| ACT_USE_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | ACT_SKIP_NUMLINES | | |
| R-0h | | | | | R/W-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ACT_SKIP_NUMLINES | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-530. VIP_PARSER_xtra5_port_b Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | Reserved | R | 0h | |
| 26-16 | ACT_USE_NUMLINES | R/W | 0h | When cropping, the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnum's active video region. |
| 15-11 | Reserved | R | 0h | |
| 10-0 | ACT_SKIP_NUMLINES | R/W | 0h | The number of lines to crop from the top of the vertical active video region. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.55 VIP_PARSER_xtra6_port_a Register (offset = D8h) [reset = 0h]

VIP_PARSER_xtra6_port_a is shown in Figure 12-612 and described in Table 12-531.

Cfg Disable Active Srcnum Vector Input for Port A

**Figure 12-612. VIP_PARSER_xtra6_port_a Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| yuv_srcnum_stop_immediately | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| yuv_srcnum_stop_immediately | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| anc_srcnum_stop_immediately | | | | | | | |
| R/W- | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| anc_srcnum_stop_immediately | | | | | | | |
| R/W- | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

**Table 12-531. VIP_PARSER_xtra6_port_a Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | yuv_srcnum_stop_immediately | R/W | 0h | For the Active Video Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A 0 in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A 1 in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to 0 , the port will never disable. |
| 15-0 | anc_srcnum_stop_immediately | R/W | 0h | For the Ancillary Data Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A 0 in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A 1 in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to 0 , the port will never disable. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 12.3.14.56 VIP_PARSER_xtra7_port_b Register (offset = DCh) [reset = 0h]

VIP_PARSER_xtra7_port_b is shown in Figure 12-613 and described in Table 12-532.

Cfg Disable Active Srcnum Vector Input for Port B

**Figure 12-613. VIP_PARSER_xtra7_port_b Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| yuv_srcnum_stop_immediately | | | | | | | |
| R/W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| yuv_srcnum_stop_immediately | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| anc_srcnum_stop_immediately | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| anc_srcnum_stop_immediately | | | | | | | |
| R/W- | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

**Table 12-532. VIP_PARSER_xtra7_port_b Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | yuv_srcnum_stop_immediately | R/W | 0h | For the Active Video Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A 0 in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A 1 in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to 0 , the port will never disable. |
| 15-0 | anc_srcnum_stop_immediately | R/W | 0h | For the Ancillary Data Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A 0 in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A 1 in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to 0 , the port will never disable. |



Copyright © 2011–2016, Texas Instruments Incorporated

# High-Definition Multimedia Interface (HDMI)

This chapter describes the high-definition multimedia interface (HDMI) module.

**Topic** **Page**

**13.1 Introduction** ..... **2700**
**13.2 Architecture** ..... **2704**
**13.3 HDMI Registers** ..... **2720**


Copyright © 2011–2016, Texas Instruments Incorporated

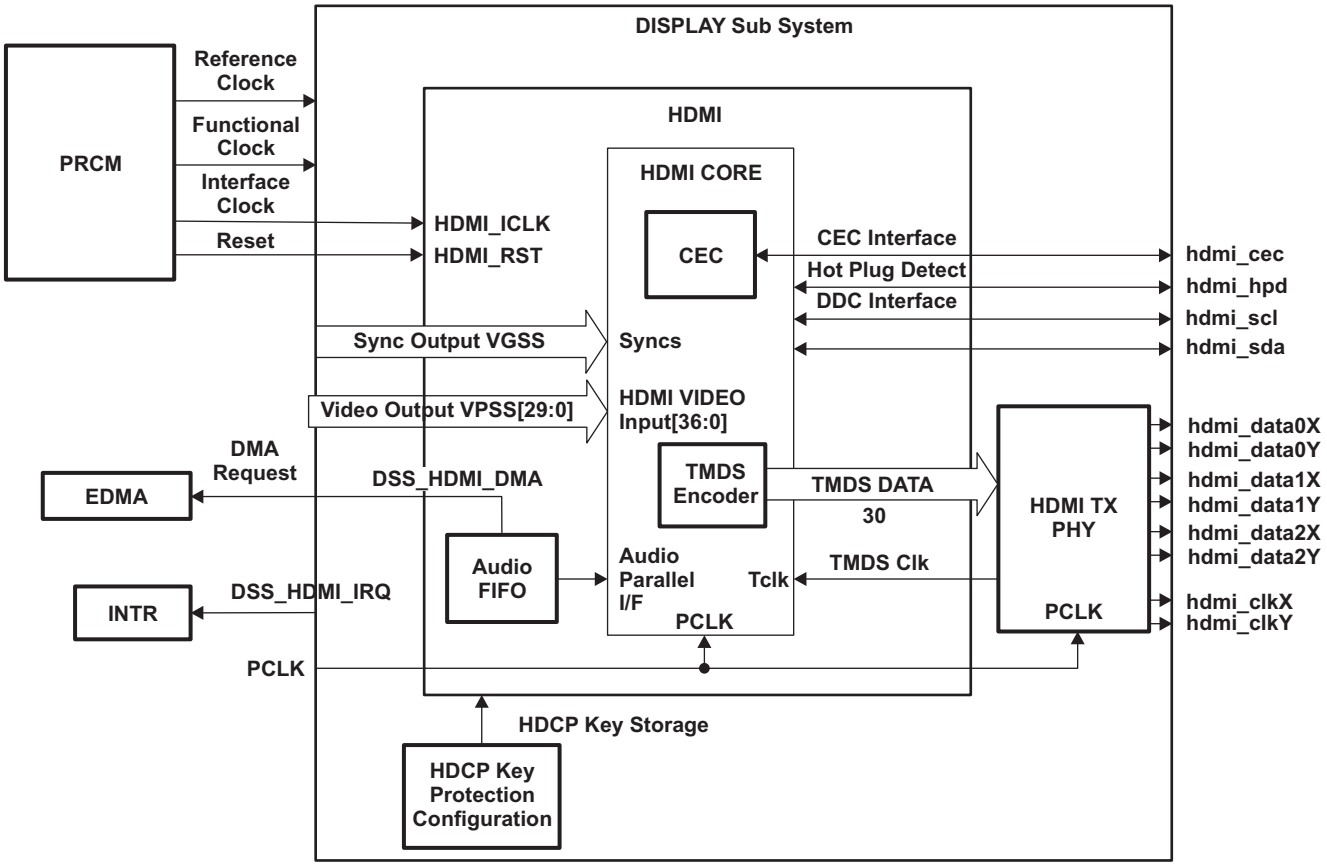
## 13.1 Introduction

### 13.1.1 Overview

The high-definition multimedia interface wrapper (HDMI_WP) module is used to encapsulate the HDMI, providing compatibility and interface to the HDMI TXPHY module. The HDMI_WP module contains the support logic for the HDMI IP, including the HDMI core (HDMI_CORE) and CEC core. The major additional parts comprise a slave interface port for configuration and buffering for the audio data streams.

Figure 13-1 is an overview of the HDMI module.

**Figure 13-1. HDMI Overview**




### 13.1.2 Main Features

The HDMI module provides the following main features:

- HDMI 1.3, HDCP 1.2, and DVI 1.0 compliant
- EIA/CEA-861-D video format support
- VESA Display Monitor Timing (DMT) video format support
- Support for deep-color mode:
  - 10-bit/component color depth up to 1080p at 60 Hz
  - 12-bit/component color depth up to 720p/1080i at 60 Hz
- Supports up to 165-MHz pixel clock (1920 × 1080p at 60 Hz)
- Video formats: 24-bit RGB
- Uncompressed multichannel (up to eight channels) audio (L-PCM) support
- Master inter-intergrated circuit (I2C) interface for display data channel (DDC) connection

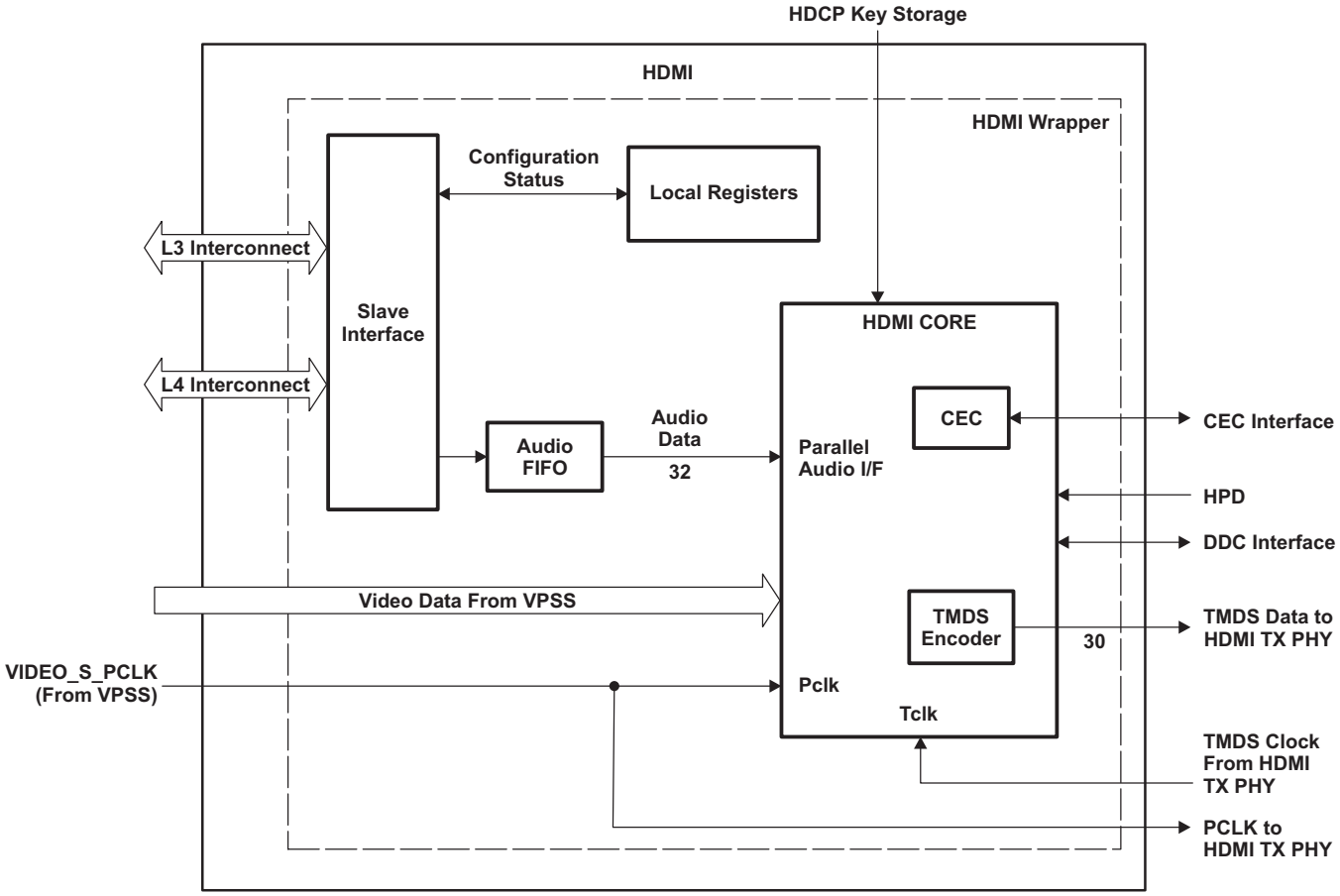

Copyright © 2011–2016, Texas Instruments Incorporated

- Consumer Electronic Control (CEC) interface
- Integrated High-bandwidth Digital Content Protection (HDCP) encryption engine for transmitting protected audio and video content (authentication performed by software)
- Integrated Transition Minimized Differential Signaling (TMDS) and TER4 encoders for data island support
- Integrated TMDS physical layer (PHY) (three TMDS differential data lanes + TMDS differential clock lane)
  - Up to 1.85625 Gbps per lane at (1080p at 60 Hz at 10 bits/component, lower resolutions at 12 bits/component)
  - 928.125 Mbps per lane at (720p/1080i at 60 Hz 10 bits/component, lower resolutions at 12 bits/component)
  - 337.5 Mbps at (576p at 50 Hz/480p at 60 Hz 10 bits/component)

### 13.1.3 Functional Block Diagram

Figure 13-2 is a functional block diagram of the HDMI module.

**Figure 13-2. HDMI Block Diagram**




### 13.1.4 Video Formats and Timings

Table 13-1 lists the video timings supported by the HDMI module, according to the CEA-861-D standard.

Table 13-2 lists the video timings supported by the HDMI module, according to the VESA DMT standard.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 13-1. HDMI Video Timings (CEA-861-D)**

| Refresh Rate | Resolution |
|---|---|
| 24 Hz (low-field rate) | 1920 x 1080p |
| 50 Hz | 1920 x 1080p |
| | 1920 x 1080i |
| | 1280 x 720p |
| | 2880 x 576p |
| | 1440 x 576p |
| | 1440 x 576i |
| | 720 x 576p |
| 59.94 Hz | 1920 x 1080p |
| | 2880 x 480p |
| | 1440 x 480p |
| | 1440 x 480i |
| | 720 x 480p |
| | 640 x 480p |
| 60 Hz | 1920 x 1080p |
| | 1920 x 1080i |
| | 2880 x 480p |
| | 1280 x 720p |
| | 1440 x 480p |
| | 1440 x 480i |
| | 720 x 480p |
| | 640 x 480p |

**Table 13-2. HDMI Video Timings (VESA DMT)**

| Refresh Rate | Resolution |
|---|---|
| 60 Hz | 640 x 480p |
| | 800 x 600p |
| | 848 x 480p |
| | 1024 x 768p |
| | 1280 x 768p |
| | 1280 x 800p |
| | 1280 x 960p |
| | 1280 x 1024p |
| | 1360 x 768p |
| | 1366 x 768p |
| | 1400 x 1050p |
| | 1440 x 900p |
| | 1680 x 1050p |
| | 1920 x 1080p |
| 60 Hz (reduced blanking) | 1280 x 768p |
| | 1280 x 800p |
| | 1400 x 1050p |
| | 1440 x 900p |
| | 1680 x 1050p |

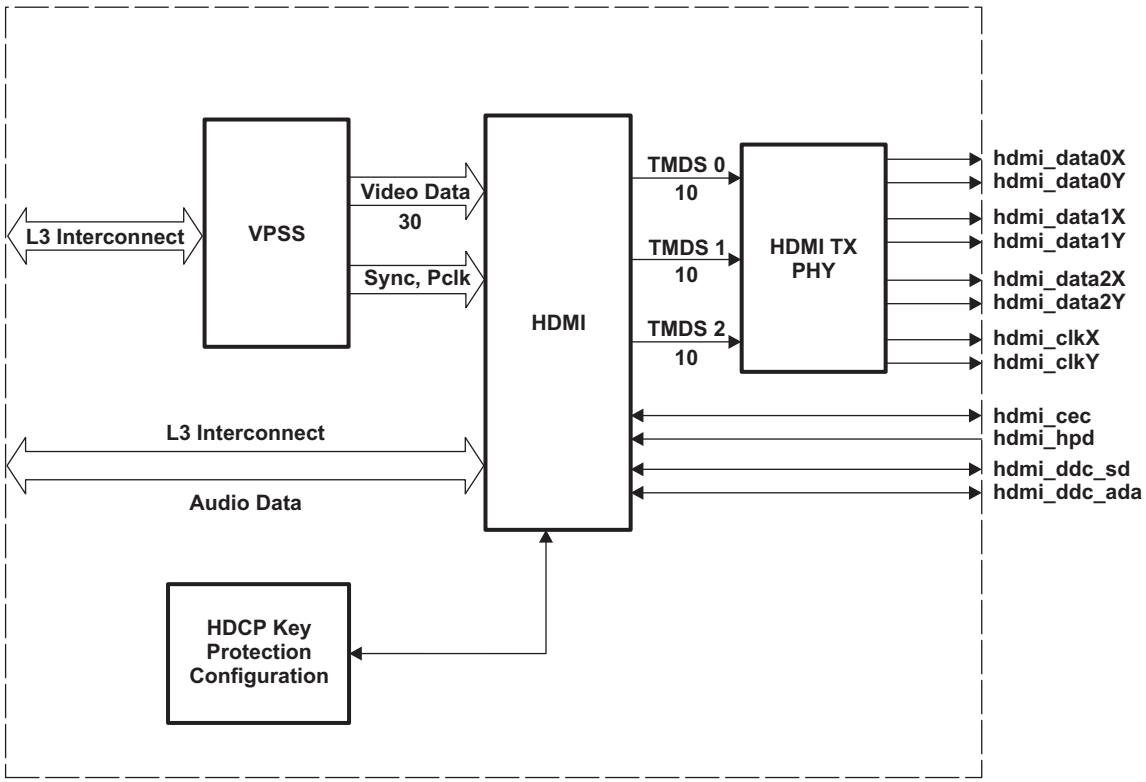

Copyright © 2011–2016, Texas Instruments Incorporated

### 13.1.5 Environment

This section describes the HDMI application fields from an environment point of view (external connections). It describes HDMI connectivity options, lists all possible interfaces, and describes the protocol and data format used in each case.

Figure 13-3 shows the external connections and simplified data path for the HDMI module.

**Figure 13-3. HDMI Environment**




The following paths are used to transmit video and audio data through the HDMI link:

- Video data path:
  - Video Processing Sub System (VPSS)
  - HDMI module
  - HDMI complex input/output (I/O) (HDMI_TXPHY)

The VPSS module provides synchronization signals to the HDMI and synchronously sends 30 bits of pixel data to the HDMI with these signals. The digital output of the VPSS is always a 30-bit RGB value .For more information about the connection between the VPSS and HDMI, see Section 1.4.7.1, HDMI Video Interface. For more information about the VPSS settings and features, see the Video Processing Sub System chapter in the device TRM. The video data received on the HDMI video slave port is always a 30-bit RGB value. The HDMI module performs pixel processing that allows manipulation of the pixels received on the video interface. The processed data is sent to the HDCP for encryption, or directly to the HDMI_TXPHY module after TDMS encoding, if there is no HDCP encryption enabled.

- Audio data path:
  - Level 3 (L3) interconnect
  - HDMI module
  - HDMI complex I/O (HDMI_TXPHY)


Copyright © 2011–2016, Texas Instruments Incorporated

The audio data is received on the buffered HDMI audio port through the L3 Interconnect using a direct memory access (DMA) request to the device system DMA (EDMA) module, or interrupt request (IRQ) to the device microprocessor unit (MPU) interrupt controllers (INTCs). In the HDMI, the data is packed, formatted, and sent to the HDMI_TXPHY module.

Additionally, the HDMI module provides universal remote control capability through the CEC protocol on the hdmi_cec pin, which allows common communication between different pieces of consumer electronics when the device is connected to them by HDMI cable.

The HDMI module can drive the display data channel (DDC) bus, which supports a variety of I2C commands, on the hdmi_ddc_scl and hdmi_ddc_sda pins. This allows the TV-set/monitor to provide information to the host device, such as supported resolutions.

Hot plug detection (HPD) capability is supported by the HDMI module on the hdmi_hpd pin. It allows detection of the presence of an attached TV display.

## 13.2 Architecture

This section describes the HDMI configuration.

### *13.2.1 Clock Configuration*

#### 13.2.1.1 Clock Domains

There are three main clock domains within the HDMI module:

- L3 clock domain: Runs at the frequency of the DSS_L3_ICLK clock, generated by the device PRCM module, and is asynchronous with the other clocks
- TCLK clock domain: Runs at the frequency of the TMDS clock (TMDS_CLK) generated by the HDMI_TXPHY module
- PCLK clock domain: VPSS to HDMI_WP input runs at PCLK generated by VPSS Venc.

#### 13.2.1.2 CEC Interface Clock Configuration

The clock for the CEC interface module (CEC_DDC_CLK) is generated inside the HDMI module from the DSS_HDMI clock (48 MHz) by division. The divider value is set through the HDMI_WP_CLK[5:0] CEC_DIV bit field, as defined in Table 13-3. The clock frequency required by the CEC protocol is 2 MHz, and is achieved by setting the CEC_DIV bit field to 18h.

**Table 13-3. CEC Clock Generation**

| Reference Clock | HDMI_WP_CLK[5:0] CEC_DIV Value | CEC Clock |
|---|---|---|
| DSS_HDMI | 0 | Gated |
| - | 1 | Free running |
| - | ... | ... |
| - | 18h | 2-MHz clock |
| - | ... | ... |

#### 13.2.1.3 Time-Out Capability

The HDMI module provides time-out capability to the DSS_HDMI clock. The OCP_TIME_OUT_INTR interrupt is generated, if the HDMI_WP_CLK[16] OCP_TIME_OUT_DIS bit is cleared to 0 and the DSS_HDMI clock is not provided to the HDMI module. For more information about the OCP_TIME_OUT_INTR interrupt, see Section 13.2.6. The time-out capability can be disabled by writing 1 to the HDMI_WP_CLK[16] OCP_TIME_OUT_DIS bit. By default, this option is enabled.


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.2.2 Signals

Table 13-4 describes the HDMI module signals.

**Table 13-4. HDMI I/O Signal Description**

| Signal | I/O[1] | Description | Function |
|---|---|---|---|
| hdmi_data0x<br>hdmi_data0y | O | Transmit data lane 0 | TMDS serial data output |
| hdmi_data1x<br>hdmi_data1y | O | Transmit data lane 1 | TMDS serial data output |
| hdmi_data2x<br>hdmi_data2y | O | Transmit data lane 2 | TMDS serial data output |
| hdmi_clockx<br>hdmi_clocky | O | Transmit clock lane | TMDS clock output |
| hdmi_cec | I/O | CEC interface line | CEC interface input/output |
| hdmi_hpd | I | Monitor charge input | Used to connect to the HDMI hot plug pin to detect the presence of an attached monitor |
| hdmi_ddc_scl | I/O | DDC I2C clock line | DDC clock input/output |
| hdmi_ddc_sda | I/O | DDC I2C clock line | DDC clock input/output |

(1) I = Input; O = Output

### 13.2.3 Integration

This section describes the module integration in the device (see Figure 13-4), including information about clocks, resets, and hardware requests.

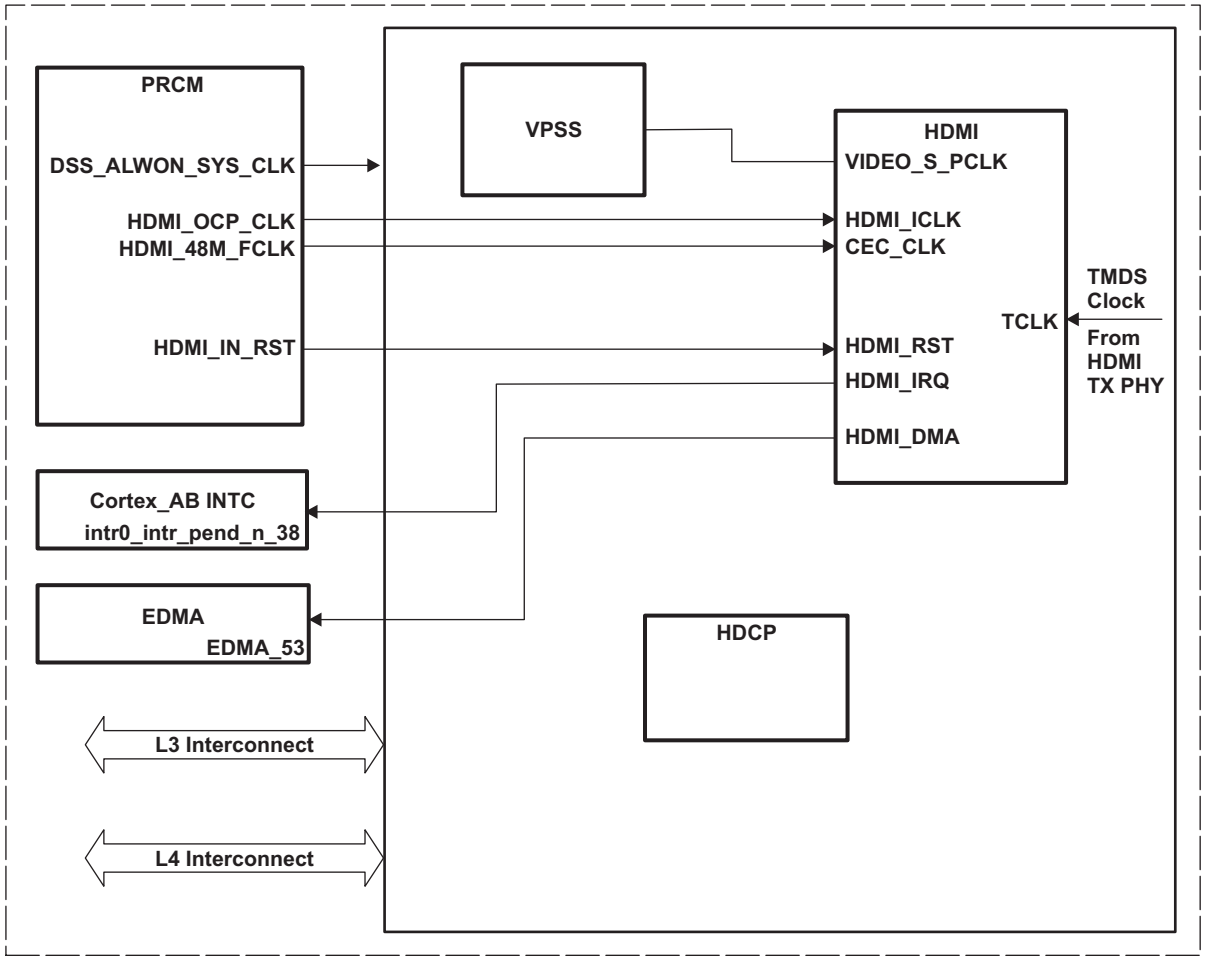
Table 13-5, Table 13-6, and Table 13-7 summarize the integration of the module in the device.

### 13.2.4 Software Reset

Global reset of the module is done at the PRCM module level. The HDMI module can be reset by software. This has the same effect as a hardware reset. The HDMI module can be reset by setting the HDMI_WP_SYSCONFIG[0] SOFTRESET bit to 1. Software can monitor this bit to wait for the reset to complete.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 13-4. HDMI Integration**




**NOTE:** For more information about the IDLE hardware handshake and the wake-up request, see Sleep (Idle) and Wake-Up Management, in Power, Reset, and Clock Management.

**Table 13-5. Integration Attributes**

| Module Instance | Attributes |
|---|---|
| HDMI | Power Domain — Interconnect<br>Active Domain —L3 for Data; L4 for Configuration |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 13-6. Clocks and Resets**

| Module Instance | Destination Signal Name | Source Signal Name | Source | Destination |
|---|---|---|---|---|
| | | **Clocks** | | |
| HDMI | DSS_HDMI | HDMI_PHY_48_S_FCLK | PRCM | The CEC/DDC clock is derived from this clock. Also reference clock for HDMI_TXPHY module. |
| | HDMI_ICLK | DSS_L3_ICLK | PRCM | Interface clock |
| | TCLK | TMDS_CLK | HDMI_TXPHY | The TMDS encoded data is sent to the HDMI_TXPHY module on this clock. |
| VPSS | VIDEO_S_PCLK | HDMI_CLK | VPSS | HDMI |
| | | **Resets** | | |
| HDMI | HDMI_RST | HDMI_IN_RST | PRCM | Nonretention reset |

**Table 13-7. Hardware Requests**

| Module Instance | Desination Signal Name | Source Signal Name | Destination | Description |
|---|---|---|---|---|
| HDMI | DSS_HDMI_IRQ | intr0_intr_pend_n_38 | CortexA8 | HDMI Interrupt Request |
| HDMI | DSS_HDMI_DMA | EDMA_53 | EDMA | HDMI audio DMA request |

### *13.2.5 Power Management*

Table 13-8 describes the power-management features available to the HDMI module.

**NOTE:** For information about source clock gating and a description of the sleep/wake-up transitions, see the Sleep (Idle) and Wake-Up Management section in the Power, Reset, and Clock Management..

For descriptions of the EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see the AutoIDLE Clock Control section in the Power, Reset, and Clock Management.

**Table 13-8. Local Power-Management Features**

| Feature | Registers | Description |
|---|---|---|
| Slave idle modes | HDMI_WP_SYSCONFIG[3:2] IDLEMODE bit field | Force-idle, no-idle, smart-idle, and smart-idle wakeup-capable modes are available. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.2.6 Interrupt Requests

One external interrupt line is generated by the HDMI module: DSS_HDMI_IRQ (for more information, see Table 13-7). Table 13-9 lists the event flags, and their masks, that can cause internal module interrupts that are mapped to DSS_HDMI_IRQ.

**Table 13-9. HDMI Interrupt Events**

| Event Flag | Event Mask | Map To | Description |
|---|---|---|---|
| HDMI_WP_IRQSTATUS[10] AUDIO_FIFO_SAMPLE_REQ_INTR | HDMI_WP_IRQENABLE_SET[10] ENABLE_SET_AUDIO_FIFO_ SAMPLE_REQ_INTR | AUDIO_FIFO_SAMPLE_ REQ_INTR | Audio FIFO requests some data in IRQ mode |
| HDMI_WP_IRQSTATUS[9] AUDIO_FIFO_OVERFLOW_INTR | HDMI_WP_IRQENABLE_SET[9] ENABLE_SET_AUDIO_FIFO_ OVERFLOW_INTR | AUDIO_FIFO_ OVERFLOW_INTR | Audio FIFO overflow |
| HDMI_WP_IRQSTATUS[8] AUDIO_FIFO_UNDERFLOW_INTR | HDMI_WP_IRQENABLE_SET[8] ENABLE_SET_AUDIO_FIFO_ UNDERFLOW_INTR | AUDIO_FIFO_ UNDERFLOW_INTR | Audio FIFO underflow |
| HDMI_WP_IRQSTATUS[4] OCP_TIME_OUT_INTR | HDMI_WP_IRQENABLE_SET[4] OCP_TIME_OUT_INTR | OCP_TIME_OUT_INTR | Time-out interrupt in case the DSS_HDMI clock is not provided |
| HDMI_WP_IRQSTATUS[0] CORE_INTR | HDMI_WP_IRQENABLE_ SET[0] ENABLE_SET_CORE_INTR | CORE_INTR | HDMI core has generated internally an interrupt. Software must read the interrupt status register in the HDMI core to identify the interrupt event(s). |

**NOTE:** The HDMI_WP_IRQENABLE_SET register is used to enable the interrupt event(s), by writing 1 in the desired bit field(s). To disable the interrupt event(s), the HDMI_WP_IRQENABLE_CLR register must used, by writing 1 in the respective bit field(s).

### 13.2.7 DMA Requests

The HDMI module generates one DMA request for audio data: DSS_HDMI_DMA (for more information, see Section 13.2.8.2.2.1).

### 13.2.8 Wrapper Functions

#### 13.2.8.1 Wrapper Video Interface

##### 13.2.8.1.1 Video Interface Features

- Video slave port with 30 bit RGB data format (deep color mode)
- Slave operational mode:
  - Synchronization signals, pixel clock, data-enable and data pixels are provided to the HDMI module
- CEA 861-D video timings support at 50 Hz, 59.94 Hz, and 60 Hz (low field rate 24 Hz included)
- VESA DMT video timings support at 60 Hz (reduced blanking included)
  - 10 bits per color component for 1080p/60fps and 1080p/50fps – 12 bits per color component otherwise
- Support for TCLK up to 185.625 MHz (TCLK derived from PCLK)


Copyright © 2011–2016, Texas Instruments Incorporated

#### 13.2.8.1.2 Video Interface Description

**NOTE:** The signals described in this section are internal and not bounded outside the device.

The HDMI module receives synchronization signals, pixel clock, and 30 bits of pixel data on its video slave port from VPSS module (Table 13-10). The digital output of the VPSS is always a 30-bit RGB value. This is extended to 36 bits by adding LSB zeroes, as listed in Table 13-11.

**Table 13-10. Video Port Signals**

| Signal Name | Width | Direction | Description |
|---|---|---|---|
| VIDEO_S_PCLK | --- | IN | Input pixel clock (PCLK) |
| VIDEO_S_DATA | 35:0 | IN | Video data input bus |
| VIDEO_S_DE | --- | IN | Data Enable |
| VIDEO_S_HS | --- | IN | Horizontal sync |
| VIDEO_S_VS | --- | IN | Vertical sync |
| VIDEO_S_FID | --- | IN | Field identifier |

**Table 13-11. HDMI Video Port Mapping**

| Color Component | VPSS HD_VENC Output Bus | HDMI Video Data Input Bus | Color Component |
|---|---|---|---|
| R9 | VPSS_DATA[29] | HDMI_VIDEO_DATA[35] | R11 |
| R8 | VPSS_DATA[28] | HDMI_VIDEO_DATA[34] | R10 |
| R7 | VPSS_DATA[27] | HDMI_VIDEO_DATA[33] | R9 |
| R6 | VPSS_DATA[26] | HDMI_VIDEO_DATA[32] | R8 |
| R5 | VPSS_DATA[25] | HDMI_VIDEO_DATA[31] | R7 |
| R4 | VPSS_DATA[24] | HDMI_VIDEO_DATA[30] | R6 |
| R3 | VPSS_DATA[23] | HDMI_VIDEO_DATA[29] | R5 |
| R2 | VPSS_DATA[22] | HDMI_VIDEO_DATA[28] | R4 |
| R1 | VPSS_DATA[21] | HDMI_VIDEO_DATA[27] | R3 |
| R0 | VPSS_DATA[20] | HDMI_VIDEO_DATA[26] | R2 |
| GND | GND | HDMI_VIDEO_DATA[25] | R1 |
| GND | GND | HDMI_VIDEO_DATA[24] | R0 |
| G9 | VPSS_DATA[19] | HDMI_VIDEO_DATA[23] | G11 |
| G8 | VPSS_DATA[18] | HDMI_VIDEO_DATA[22] | R10 |
| G7 | VPSS_DATA[17] | HDMI_VIDEO_DATA[21] | G9 |
| G6 | VPSS_DATA[16] | HDMI_VIDEO_DATA[20] | G8 |
| G5 | VPSS_DATA[15] | HDMI_VIDEO_DATA[19] | G7 |
| G4 | VPSS_DATA[14] | HDMI_VIDEO_DATA[18] | G6 |
| G3 | VPSS_DATA[13] | HDMI_VIDEO_DATA[17] | G5 |
| G2 | VPSS_DATA[12] | HDMI_VIDEO_DATA[16] | G4 |
| G1 | VPSS_DATA[11] | HDMI_VIDEO_DATA[15] | G3 |
| G0 | VPSS_DATA[10] | HDMI_VIDEO_DATA[14] | G2 |
| GND | GND | HDMI_VIDEO_DATA[13] | G1 |
| GND | GND | HDMI_VIDEO_DATA[12] | G0 |
| B9 | VPSS_DATA[9] | HDMI_VIDEO_DATA[11] | B11 |
| B8 | VPSS_DATA[8] | HDMI_VIDEO_DATA[10] | B10 |
| B7 | VPSS_DATA[7] | HDMI_VIDEO_DATA[9] | B9 |
| B6 | VPSS_DATA[6] | HDMI_VIDEO_DATA[8] | B8 |
| B5 | VPSS_DATA[5] | HDMI_VIDEO_DATA[7] | B7 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 13-11. HDMI Video Port Mapping (continued)**

| Color Component | VPSS HD_VENC Output Bus | HDMI Video Data Input Bus | Color Component |
|---|---|---|---|
| B4 | VPSS_DATA[4] | HDMI_VIDEO_DATA[6] | B6 |
| B3 | VPSS_DATA[3] | HDMI_VIDEO_DATA[5] | B5 |
| B2 | VPSS_DATA[2] | HDMI_VIDEO_DATA[4] | B4 |
| B1 | VPSS_DATA[1] | HDMI_VIDEO_DATA[3] | B3 |
| B0 | VPSS_DATA[0] | HDMI_VIDEO_DATA[2] | B2 |
| GND | GND | HDMI_VIDEO_DATA[1] | B1 |
| GND | GND | HDMI_VIDEO_DATA[0] | B0 |

#### *13.2.8.1.3 Video Interface Configuration*

The slave port is connected to the HDMI module. Always clear the HDMI_WP_VIDEO_CFG[1:0] Mode bit field to 0.

According to the video data bus connection and pixel format, packing mode must be selected. The format supported by the HDMI module is 36-bit RGB. The packing can be selected through the HDMI_WP_VIDEO_CFG[10:8] PACKING_MODE bit field:

- HDMI_WP_VIDEO_CFG[10:8] PACKING_MODE = 0: If 10-bit deep-color mode is selected and

RGB format is used on the video port data bus:

- HDMI_WP_VIDEO_CFG[10:8] PACKING_MODE = 1: If 8-bit deep-color mode is not selected and

RGB format is used on the video port data bus:

- HDMI_WP_VIDEO_CFG[10:8] PACKING_MODE = 7h: If 12-bit deep-color mode is selected and

RGB format is used on the video port data bus. No change on input video data lines. The VIDEO_S_DATA[35:0] data is provided without any change in it.

Video Timing is controlled by VENC module inside VPSS which drives HDMI module. Please refer VPSS PRG for information about timing for different resolutions.

### 13.2.8.2 Wrapper Audio Interface

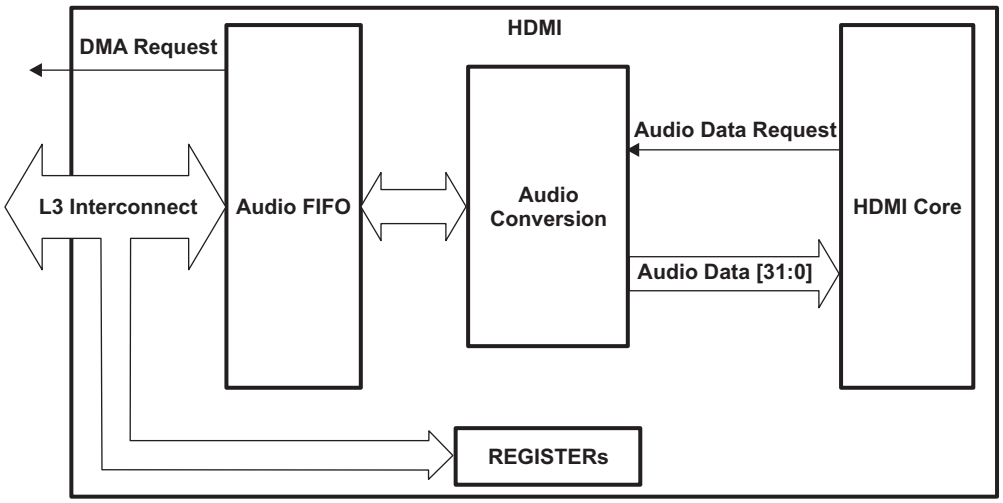
#### *13.2.8.2.1 Audio Interface Description*

The HDMI module supports parallel interface that is based on a filling of the audio FIFO through the L3 interconnect. The audio data and register configuration is received through the L3 interconnect on a request to the device EDMA module. Each 32-bit access is split into four consecutive accesses to the HDMI Core 8-bit registers of the 32-bit access, starting with the least-significant bit (LSB) to the MSB.

Figure 13-5 is an overview of the audio interface to the HDMI core.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 13-5. HDMI Audio Interface Overview**




#### 13.2.8.2.2 Audio FIFO Overview

The audio FIFO provides buffering to ensure that audio samples can always be delivered to the HDMI core exactly when required, despite the L3 interconnect latency.

##### 13.2.8.2.2.1 Audio Data Request

By default, the audio FIFO is filled by the device EDMA using a standard DMA request signal mapped to DSS_HDMI_DMA on the display subsystem level. For additional mapping information, see Table 13-7.

The FIFO can also use an IRQ signal (AUDIO_FIFO_SAMPLE_REQ_INTR) to request audio data, instead of a DMA request, using the same threshold. The IRQ status can be read in the HDMI_WP_IRQSTATUS[10] AUDIO_FIFO_SAMPLE_REQ_INTR bit. For more information about the audio IRQ, see Table 13-9.

The selection between DMA or IRQ request can be done through the HDMI_WP_AUDIO_CTRL[9] DMA_OR_IRQ bit.

The audio data is written in the HDMI_WP_AUDIO_DATA[31:0] FIFO_DAT bit field.

The request is made based on comparison of fullness with register programmable thresholds. The DMA/IRQ is generated when the number of samples (24- or 16-bit) in the FIFO is less than or equal to the threshold value.

- The size of samples is selected through the HDMI_WP_AUDIO_CFG[0] SAMPLE_SIZE bit.
- The valid number of samples can be read from the HDMI_WP_AUDIO_CTRL[25:16].

NUMBER_OF_SAMPLE bit field:

- The threshold value can be set in the HDMI_WP_AUDIO_CTRL[8:0] TRESHOLD_VALUE bit field.

The DSS_HDMI_DMA signal is released (deactivated) when the last transfer of the DMA burst is performed in the FIFO. An internal counter is used to do this. The reset value of this counter is set to 16 and the maximum value is set to 64. In this case, the maximum length of the DMA transfer is 64 (32-bit interface access). The counter can be configured through the HDMI_WP_AUDIO_CFG2[15:8] DMA_TRANSFER bit field.

The threshold value and the DMA burst size must be set to ensure the filling of the FIFO to four samples. The minimum level (otherwise, underflow occurs) is four audio samples, which can be 2 × 32-bit data (in case of two samples per word) or 4 × 32-bit data (IEC or one sample per word). The interconnect latency must be considered. The number of samples is controlled through the HDMI_WP_AUDIO_CFG[1] SAMPLE_NBR bit.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 13.2.8.2.2.2 Audio FIFO Error (Underflow and Overflow)

The underflow and overflow status are required and highlighted by sending interrupt requests IRQs).

The FIFO underflow condition occurs when the HDMI core is requesting data from the FIFO when the FIFO is empty. The status can be read through the HDMI_WP_IRQSTATUS[8] AUDIO_FIFO_UNDERFLOW_INTR bit.

The underflow IRQ may happen during a sequence only if one of the following occurs:

- The audio data path is not set correctly. In this case, the device EDMA cannot fill the FIFO fast enough.
- It is the end of the sequence and the number of samples is not a multiple of two times the value in the HDMI_WP_AUDIO_CFG2[7:0] BLOCK_SIZE bit field (384 samples by default).

The FIFO filling must be ensured by the system before the HDMI core is enabled; this means that in steady state the filling of the FIFO is always four words of 32 bits (except at the end).

The FIFO overflow condition occurs when an audio data is written (by the device EDMA or MPU modules) while the FIFO is full. The status can be read through the HDMI_WP_IRQSTATUS[9] AUDIO_FIFO_OVERFLOW_INTR bit.

The overflow IRQ may happen only if:

- The device EDMA module does not respect the DMA transfer length.
- The threshold is not set to the correct value in the HDMI_WP_AUDIO_CTRL[8:0] TRESHOLD_VALUE bit field.
- The device MPU module does not check the filling of the FIFO when the audio data is requested through an IRQ.

#### 13.2.8.2.2.3 Audio FIFO Reset

The audio sample in the FIFO can be flushed by writing 0 in the HDMI_WP_AUDIO_CTRL[31] WRAPPER_ENABLE bit. As soon as the enable signal is deactivated (set to 0),the following occur: 1. The audio FIFO is reset. 2. The read and write pointers are reset. 3. The DSS_HDMI_DMA signal is deasserted. 4. Only the threshold value in the HDMI_WP_AUDIO_CTRL[8:0] TRESHOLD_VALUE bit field is kept.

#### 13.2.8.2.3 Audio FIFO Data Formats

The format of the audio data in the audio FIFO can be configured through the HDMI_WP_AUDIO_CFG[4] IEC bit field.

- IEC = 0: The audio format is L-PCM.
- IEC = 1: The audio format is IEC 60958/IEC 61937.

#### 13.2.8.2.3.1 L-PCM Format in Audio FIFO

Table 13-12 describes the 16-bit data format in the audio FIFO.

**Table 13-12. PCM, 16-Bit Format**

| Order | Bits [31:16] | Bits [15:0] |
|---|---|---|
| --- | 16 bits, 2 channels (stereo) | --- |
| 1 | Right - Channel 2 | Left - Channel 1 |
| --- | 16 bits, 4 channels | --- |
| 1 | Right - Channel 2 | Left - Channel 1 |
| 2 | Right - Channel 4 | Left - Channel 3 |
| 3 | Right - Channel 2 | Left - Channel 1 |
| 4 | Right - Channel 4 | Left - Channel 3 |
| --- | 16 bits, 8 channels | --- |
| 1 | Right - Channel 2 | Left - Channel 1 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 13-12. PCM, 16-Bit Format (continued)**

| Order | Bits [31:16] | Bits [15:0] |
|---|---|---|
| 2 | Right - Channel 4 | Left - Channel 3 |
| 3 | Right - Channel 6 | Left - Channel 5 |
| 4 | Right - Channel 8 | Left - Channel 7 |

Table 13-13 describes the 24-bit data format in the audio FIFO. For the following formats, if less than 24 bits are valid, the data can be MSB- or LSB-justified. This is controlled through the HDMI_WP_AUDIO_CFG[3] JUSTIFY bit field.

**Table 13-13. PCM, 24-Bit Format**

| Order | Bits [31:24] | Bits [23:0] |
|---|---|---|
| --- | 24 bits, 2 channels (stereo right - justified) | --- |
| 1 | Unused | Left - Channel 1 |
| 2 | Unused | Right - Channel 2 |
| --- | 24 bits, 2 channels stereo left - justified | --- |
| 1 | Left - Channel 1 | Unused |
| 2 | Right - Channel 2 | Unused |
| --- | 24 bits, 3 channels stereo right - justified | --- |
| 1 | Unused | Left - Channel 1 |
| 2 | Unused | Right - Channel 2 |
| 3 | Unused | Center - Channel 3 |
| 4 | Unused | Left - Channel 1 |
| 5 | Unused | Right - Channel 2 |
| 6 | Unused | Center - Channel 3 |
| --- | 24 bits, 8 channels (right-justified) | --- |
| 1 | Unused | Left - Channel 1 |
| 2 | Unused | Right - Channel 2 |
| 3 | Unused | Left - Channel 3 |
| 4 | Unused | Right - Channel 4 |
| 5 | Unused | Left - Channel 5 |
| 6 | Unused | Right - Channel 6 |
| 7 | Unused | Left - Channel 7 |
| 8 | Unused | Right - Channel 8 |

**NOTE:** Left justification for three- and eight-channel data formats is also possible.

#### *13.2.8.2.3.2 Speaker Mapping and Channel Ordering in the Audio FIFO*

The speaker mapping in the eight channels is defined by the CEA-861-D specification and is described in Table 13-14. The CEA code is used to program the Packet Byte 4 of the Audio Info FRAME register inside the HDMI core (for more information, see the HDMI standard v1.3).

**Table 13-14. Speaker Mapping Versus Channel**

| CEA Code | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|
| 0h | - | - | - | - | - | - | FR | FL |
| 1h | - | - | - | - | - | LFE | FR | FL |
| 2h | - | - | - | - | FC | - | FR | FL |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 13-14. Speaker Mapping Versus Channel (continued)**

| CEA Code | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|
| 3h | - | - | - | - | FC | LFE | FR | FL |
| 4h | - | - | - | RC | - | - | FR | FL |
| 5h | - | - | - | RC | - | LFE | FR | FL |
| 6h | - | - | - | RC | FC | - | FR | FL |
| 7h | - | - | - | RC | FC | LFE | FR | FL |
| 8h | - | - | RR | RL | - | - | FR | FL |
| 9h | - | - | RR | RL | - | LFE | FR | FL |
| Ah | - | - | RR | RL | FC | - | FR | FL |
| Bh | - | - | RR | RL | FC | LFE | FR | FL |
| Ch | - | RC | RR | RL | - | - | FR | FL |
| Dh | - | RC | RR | RL | - | LFE | FR | FL |
| Eh | - | RC | RR | RL | FC | - | FR | FL |
| Fh | - | RC | RR | RL | FC | LFE | FR | FL |
| 10h | RRC | RLC | RR | RL | - | - | FR | FL |
| 11h | RRC | RLC | RR | RL | - | LFE | FR | FL |
| 12h | RRC | RLC | RR | RL | FC | - | FR | FL |
| 13h | RRC | RLC | RR | RL | FC | LFE | FR | FL |
| 14h | FRC | FLC | - | - | - | - | FR | FL |
| 15h | FRC | FLC | - | - | - | LFE | FR | FL |
| 16h | FRC | FLC | - | - | FC | - | FR | FL |
| 17h | FRC | FLC | - | - | FC | LFE | FR | FL |
| 18h | FRC | FLC | - | RC | - | - | FR | FL |
| 19h | FRC | FLC | - | RC | - | LFE | FR | FL |
| 1Ah | FRC | FLC | - | RC | FC | - | FR | FL |
| 1Bh | FRC | FLC | - | RC | FC | LFE | FR | FL |
| 1Ch | FRC | FLC | RR | RL | - | - | FR | FL |
| 1Dh | FRC | FLC | RR | RL | - | LFE | FR | FL |
| 1Eh | FRC | FLC | RR | RL | FC | - | FR | FL |
| 1Fh | FRC | FLC | RR | RL | FC | LFE | FR | FL |
| 20h | - | FCH | RR | RL | FC | - | FR | FL |
| 21h | - | FCH | RR | RL | FC | LFE | FR | FL |
| 22h | TC | - | RR | RL | FC | - | FR | FL |
| 23h | TC | - | RR | RL | FC | LFE | FR | FL |
| 24h | FRH | FLH | RR | RL | - | - | FR | FL |
| 25h | FRH | FLH | RR | RL | - | LFE | FR | FL |
| 26h | FRW | FLW | RR | RL | - | - | FR | FL |
| 27h | FRW | FLW | RR | RL | - | LFE | FR | FL |
| 28h | TC | RC | RR | RL | FC | - | FR | FL |
| 29h | TC | RC | RR | RL | FC | LFE | FR | FL |
| 2Ah | FCH | RC | RR | RL | FC | - | FR | FL |
| 2Bh | FCH | RC | RR | RL | FC | LFE | FR | FL |
| 2Ch | TC | FCH | RR | RL | FC | - | FR | FL |
| 2Dh | TC | FCH | RR | RL | FC | LFE | FR | FL |
| 2Eh | FRH | FLH | RR | RL | FC | - | FR | FL |
| 2Fh | FRH | FLH | RR | RL | FC | LFE | FR | FL |
| 30h | FRW | FLW | RR | RL | FC | - | FR | FL |
| 31h | FRW | FLW | RR | RL | FC | LFE | FR | FL |


Copyright © 2011–2016, Texas Instruments Incorporated

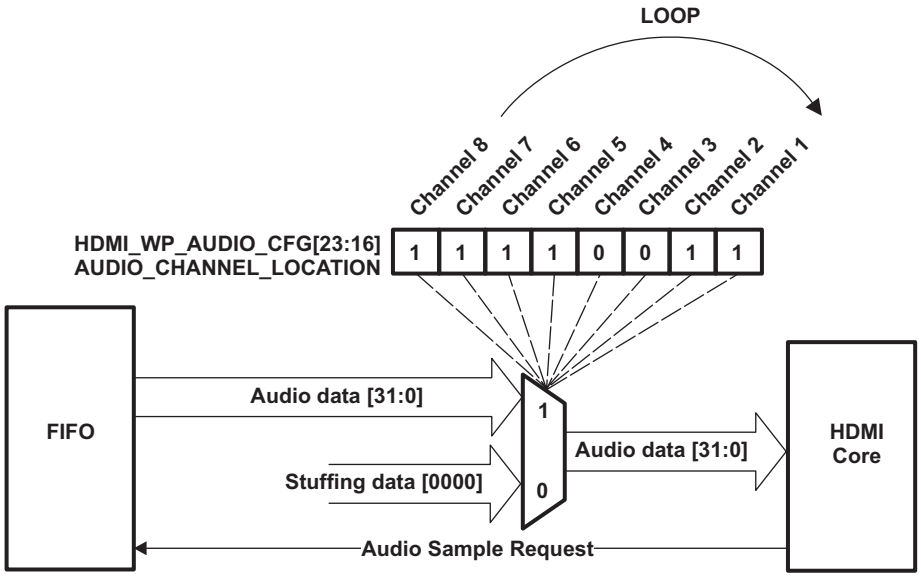
In Table 13-14, the following acronym definitions are used:

- FL: Front left
- FC: Front center
- FR: Front right
- FLC: Front left center
- FRC: Front right center
- RL: Rear left
- RC: Rear center
- RR: Rear right
- RLC: Rear left center
- RRC: Rear right center
- LFE: Low-frequency effect
- FLW: Front left wide
- FRW: Front right wide
- FLH: Front left high
- FCH: Front center high
- FRH: Front right high
- TC: Top center

The audio FIFO can be filled from two to eight channels, but the HDMI core supports only an even number of channels (the missing sample must be filled with 0, depending on the audio format).

The HDMI_WP_AUDIO_CFG[23:16] AUDIO_CHANNEL_LOCATION bit field indicates the location and the active channels. The AUDIO_CHANNEL_LOCATION field contains 8-bit fields, each corresponding to one channel. The HDMI_WP_AUDIO_CFG[26:24] STEREO_CHANNEL_ENABLE bit field indicates the number of stereo channels used. In Figure 13-6, the registers are filled accordingly with the CEA channel/speaker allocation code 10h received from the HDMI core. Upon a data request from the HDMI core, the data coming from the audio FIFO or the stuffing (filling zeros) data are sent.

**Figure 13-6. Audio Data Stuffing Behavior**

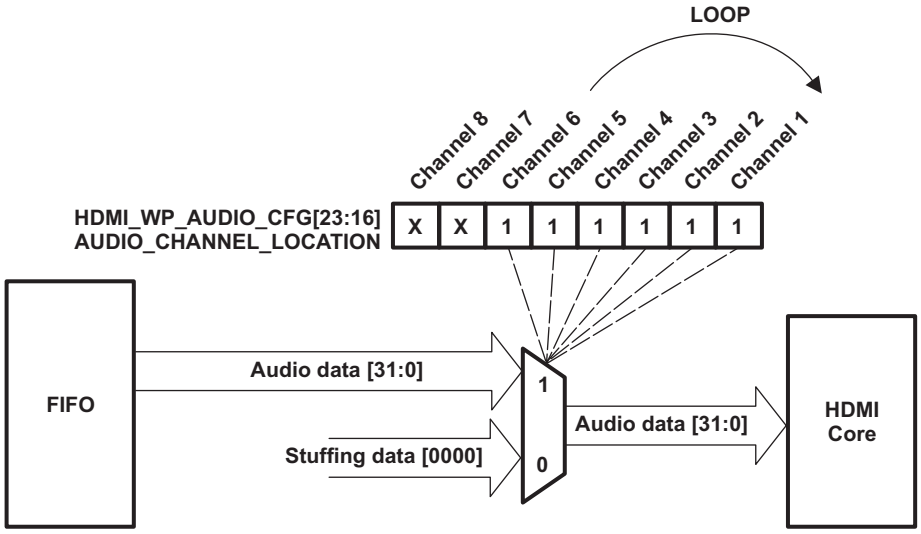



If only three stereo channels are set in the HDMI wrapper, the loop can be optimized and set to 6. If three stereo channels are enabled in the HDMI core module, the function must be set to 3h in the HDMI_WP_AUDIO_CFG[26:24] STEREO_CHANNEL_ENABLE bit field.


Copyright © 2011–2016, Texas Instruments Incorporated

Figure 13-7 shows an example with the CEA channel/speaker allocation code 7h and only three stereo channels enabled in the HDMI core. This format is also supported with the four channels enabled, but channel 7/8 must be cleared to 0.

**Figure 13-7. Audio Data Stuffing Behavior with Only Three Stereo Channels Active**




#### 13.2.8.2.3.3 IEC 61958 Format

In the IEC 61958 format (Table 13-15), the channel status and user data bits are supplied with the audio data.

The valid bits are supplied with the audio data but may be overridden based on the audio FIFO operation(forced to invalid, if FIFO data is invalid). They may also be forced to valid or invalid.

This format supports only two channels (stereo) or two monophonic channels.

The audio FIFO can be filled directly with data aligned with the 60958 format. In this case, the VUCP and the preamble data are available.

**Table 13-15. IEC 60958 Sample Format**

| Order | Bits [31:28] | Bits [27:4] | Bits [3:0] |
|---|---|---|---|
| 1 | VUCP | Left - Channel 1 | Preamble |
| 2 | VUCP | Right - Channel 2 | Preamble |

#### 13.2.8.2.3.4 IEC 61937 Format

The IEC 61937 format (Table 13-16) is similar to the IEC 61958 format, with multichannel supported.

**Table 13-16. IEC 60937 Format**

| Order | Bits [31:28] | Bits [27:4] | Bits [3:0] |
|---|---|---|---|
| 1 | VUCP | Left - Channel 1 | Preamble |
| 2 | VUCP | Right - Channel 2 | Preamble |
| 3 | VUCP | Left - Channel 3 | Preamble |
| 4 | VUCP | Right - Channel 4 | Preamble |
| 5 | VUCP | Left - Channel 5 | Preamble |
| 6 | VUCP | Right - Channel 6 | Preamble |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 13-16. IEC 60937 Format (continued)**

| Order | Bits [31:28] | Bits [27:4] | Bits [3:0] |
|---|---|---|---|
| 7 | VUCP | Left - Channel 7 | Preamble |
| 8 | VUCP | Right - Channel 8 | Preamble |

**NOTE:** The compressed formats are supported only if software can stuff the missing data with zeros to reach the bandwidth defined in the IEC 60958 format.

#### *13.2.8.2.4 AUDIO FIFO Data Adaptation*

The HDMI core supports only the IEC 60958 format, or similar formats. Between the audio FIFO and the HDMI core, a format adaptation is done to optimize the audio data transfer.

There are three different formats in the audio FIFO:

- Two L-PCM samples on 16 bits in a 32-bit container
- One L-PCM sample on 24 bits in a 32-bit container
- IEC 61937 or IEC 60958 sample format compliant

The audio formats supported include 16- and 24-bit modes for stereo (two channels) and 8-channel modes.

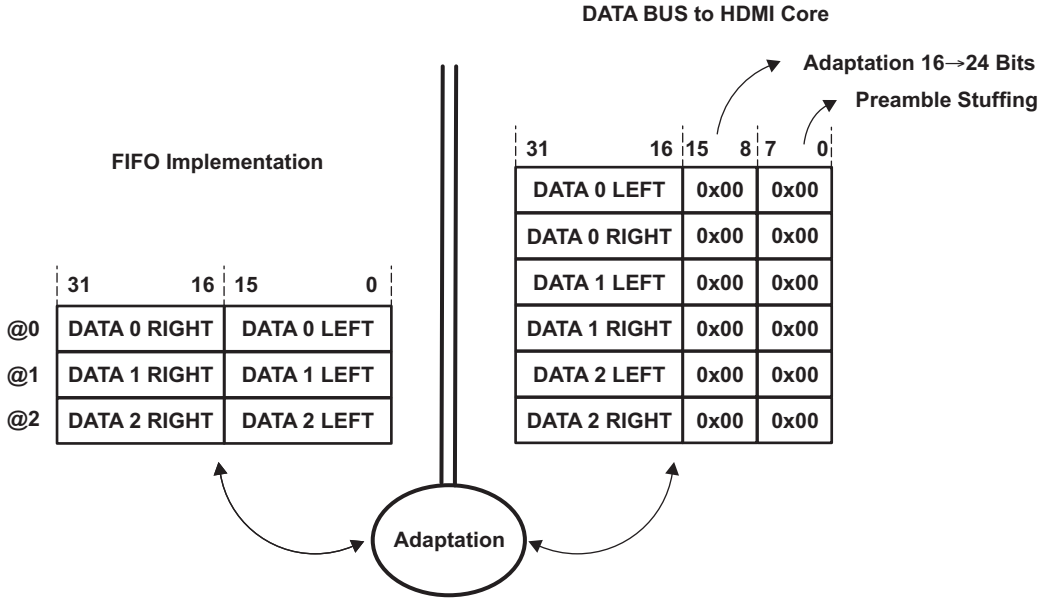
##### *13.2.8.2.4.1 IEC 61937 and IEC 60958 Format Adaptation*

The IEC 61937 and IEC 60958 formats are fully compatible with the HDMI core. In this case, data in the FIFO is sent to the HDMI core without adaptation. This is true for 16- and 24-bit modes and it is channel-number independent.

##### *13.2.8.2.4.2 L-PCM 16-Bit Format Adaptation*

When the audio FIFO is filled with 16-bit samples on the 32-bit container, the data is adapted to be similar to the IEC 61937/IEC 60958 format, as shown in Figure 13-8.

**Figure 13-8. L-PCM 16-Bit Format Adaptation**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 13.2.8.2.4.3 L-PCM 24-bit Format Adaptation

When the audio FIFO is filled with a 24-bit sample on the 32-bit container, it is already in a format similar to IEC 61937/IEC 60958, left- or right-justified, depending on the FIFO filling (the HDMI_WP_AUDIO_CFG[3] JUSTIFY bit). In this case, the data in the audio FIFO is sent directly to the HDMI core without adaptation. However, the 8 dummy bits (MSB or LSB, depending on the justification) are filled with 0 data.

### 13.2.9 TXPHY Functions

#### 13.2.9.1 TXPHY Overview

**NOTE:** The HDMI_TXPHY module must be configured before any transfer on the HDMI link. The HDMI module is fully functional only when the TMDS clock is provided. It has an internal PLL which will take in PCLK as an input and provide the required TCLK based on register configuration.

The HDMI_TXPHY receives 30 bits of TMDS parallel encoded data from the HDMI module and then prepares the data for transmission by serializing it. HDMI_TXPHY is a complex I/O with four unidirectional lane modules. This includes three data lane modules and one clock lane module. Each lane module has two data pads (DX and DY), which form a differential pair. These correspond to the HDMI serial output transmit signals, as described in Table 13-4. The data pads are connected with a complementary lane module on the external HDMI receiver device using a point-to-point interconnect.

The maximum data rate supported is 1.85625 Gbps per data lane. The lane module function and position are configurable. That is, any lane module can be chosen as a clock or a data lane module, and the DX/DY data pad for each lane module can be configured as a positive polarity (DP) or negative polarity (DN) pin.

#### 13.2.9.2 TXPHY Clock Domains

The HDMI_TXPHY module has the following clock domains:

- TMDS clock domain: This is the data interface clock to the HDMI module. The data received on the HDMI_TXPHY inputs are sampled on this clock.
- HFBITCLK clock domain: This is the high frequency clock on which the transmit of the serialized data occurs.
- REFCLK clock domain: This is the 48-MHz free-running clock (DSS_HDMI) that is used as a reference for the switched capacitor circuit.

#### 13.2.9.3 TXPHY RX Connection Detect

The HDMI_TXPHY can detect a pullup to 3.3 V on any clock and data lines. This situation occurs when an external HDMI receiver is connected to the device.

The information can be read by reading TMDS_CNTL2[0], RSEN.

#### 13.2.9.4 TXPHY Data Interface

##### 13.2.9.4.1 Input Interface

The HDMI_TXPHY module receives the 30-bit parallel data on its D0 to D2 inputs. The data is sampled on the rising edge of the TMDS clock.

##### 13.2.9.4.2 Output Interface

The serialized data on differential output lines DX/DY has the order LSB first and MSB last. The TMDS clock on the clock lane has a falling edge at the start of the 10-bit serial data word on the line.

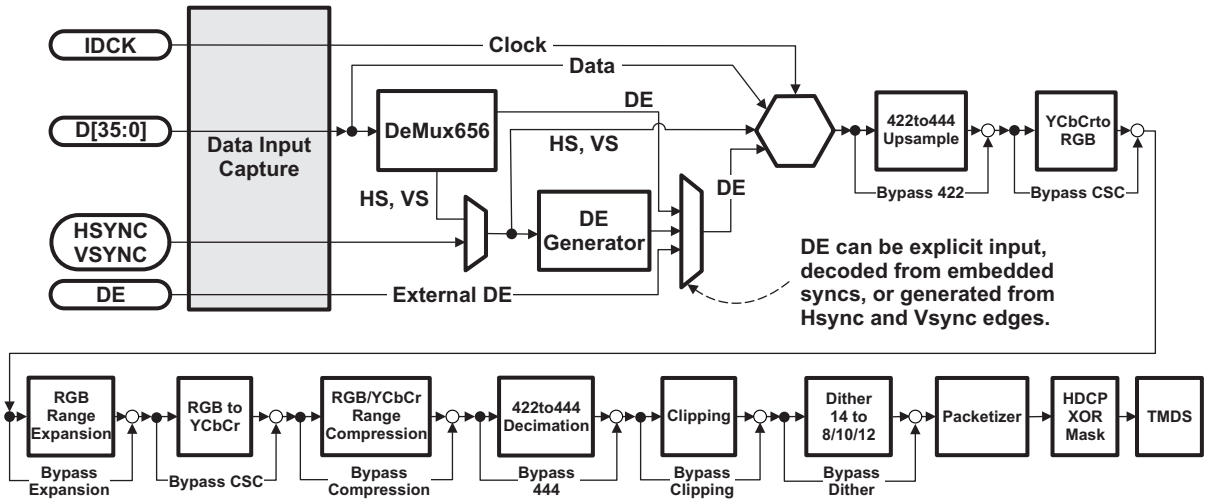

Copyright © 2011–2016, Texas Instruments Incorporated

### 13.2.10 Programming Video Input Mode and Video Output Mode

Specific registers in the HDMI Transmitter must be programmed according to the selection of input video bus mode and output video format. See Figure 13-9.

- The DE parameters need to be programmed only when the DE Generator is enabled. It is not recommended to enable the DE Generator.
- Set RANGE to 1 whenever converting YCbCr data and outputting full-range (0-255) RGB (PC mode) data across HDMI. When outputting limited-range (16-235) RGB (CE mode) data across the link, clear RANGE to 0.
- Set WIDE_BUS to the number of bits per input video channel.
- Set DITHER_MODE to the number of bits per output video channel supported by the Sink. By default, the HDMI Transmitter dithers the input (if DITHER is enabled) or truncates the input (if DITHER is disabled) to 8 bits.
- PB_CTRL1 and PB_CTRL2 should be programmed once the core is powered up only.
- Always clear (set to 0) the DE_ADJ field in IADJUST register when DE is not enabled ( Recommended usage).
- DIV_ENC_BYP field in TEST_TXCTRL register should always be programmed to zero.
- ten_bit_bypass in TMDS_CNTL9 register and enc_byp in BIST_CNTL register should always be set to 1.
- Program the HDMI registers and then enable the VENC on VPSS side. All timings are provided by the VENC module in VPSS.

**Figure 13-9. Transmitter Video Data Processing Path**





Copyright © 2011–2016, Texas Instruments Incorporated

## 13.3 HDMI Registers

### *13.3.1 HDMI Wrapper Registers*

Table 13-17 lists the HDMI wrapper registers.

**Table 13-17. HDMI Wrapper Registers**

| Address Offset | Acronym | Register Name |
|---|---|---|
| 00h | HDMI_WP_REVISION | IP Revision Identifier |
| 10h | HDMI_WP_SYSCONFIG | Clock management configuration |
| 24h | HDMI_WP_IRQSTATUS_RAW | Raw Interrupt Status |
| 28h | HDMI_WP_IRQSTATUS | Interrupt Status |
| 2Ch | HDMI_WP_IRQENABLE_SET | Interrupt Enable |
| 30h | HDMI_WP_IRQENABLE_CLR | Interrupt Disable |
| 44h | HDMI_WP_DEBOUNCE | Glitch filter |
| 50h | HDMI_WP_VIDEO_CFG | Configuration of HDMI Wrapper video |
| 70h | HDMI_WP_CLK | Configuration of clocks |
| 80h | HDMI_WP_AUDIO_CFG | Audio Configuration in FIFO |
| 84h | HDMI_WP_AUDIO_CFG2 | Audio configuration of DMA |
| 88h | HDMI_WP_AUDIO_CTRL | Audio FIFO control |
| 8Ch | HDMI_WP_AUDIO_DATA | TX Data of FIFO |

#### 13.3.1.1 IP Revision Identifier Register (HDMI_WP_REVISION)

The IP revision identifier register is shown in Figure 13-10 and described in Table 13-18.

**Figure 13-10. IP Revision Identifier Register (HDMI_WP_REVISION)**

| 31 | 0 |
|---|---|
| Revision | |
| R-5003 0200h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-18. IP Revision Identifier Register (HDMI_WP_REVISION) Field Descriptions**

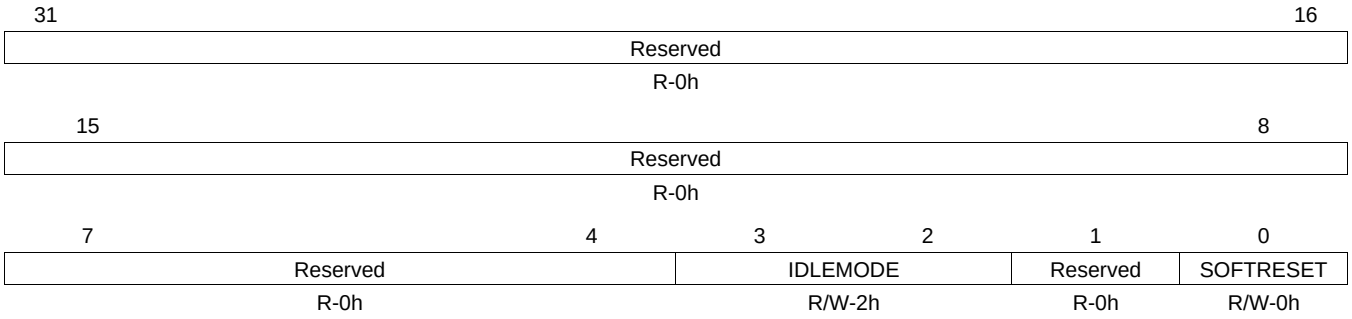
| Bit | Field | Description |
|---|---|---|
| 31-0 | Revision | IP Revision |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.2 Clock Management Configuration Register (HDMI_WP_SYSCONFIG)

The Clock management configuration register is shown in Figure 25-24 and described in Table 13-19.

**Figure 13-11. Clock Management Configuration Register (HDMI_WP_SYSCONFIG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-19. Clock Management Configuration Register (HDMI_WP_SYSCONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3-2 | IDLEMODE | | Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. Default = 2h. |
| | | 0 | Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is regardless of the IP module's internal requirements. Backup mode, for debug only. |
| | | 1h | No-idle mode: local target never enters idle state. Used in case Audio is transferred (to avoid DSS_L3_ICLK clock to be shut down) |
| | | 2h | Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module should not generate (IRQ- or DMA-request-related) wakeup events. To be used, if only Video is transmitted (since the DSS_L3_ICLK clock is not required) |
| | | 3h | Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. |
| 1 | Reserved | 0 | Reserved |
| 0 | SOFTRESET | | Software reset. (Optional) |
| | | R0 | Software reset done, no pending action |
| | | W0 | No action |
| | | R1 | Software reset ongoing |
| | | W1 | Initiate software reset |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.3 Raw Interrupt Status Register (HDMI_WP_IRQSTATUS_RAW)

The raw interrupt status register is shown in Figure 13-12 and described in Table 13-20.

**Figure 13-12. Raw Interrupt Status Register (HDMI_WP_IRQSTATUS_RAW)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | AUDIO_FIFO_SAMPLE_REQ_INTR | AUDIO_FIFO_OVERFLOW_INTR | AUDIO_FIFO_UNDERFLOW_INTR |
| R-0h | | | | | R/W1S-0h | R/W1S-0h | R/W1S-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | OCP_TIME_OUT_INTR | Reserved | | | CORE_INTR |
| R-0h | | | R/W1S-0h | R-0h | | | R/W1S-0h |

LEGEND: R/W = Read/Write; R = Read only; W1S = Write 1 to set, write 0 has no effect; -*n* = value after reset

**Table 13-20. Raw Interrupt Status Register (HDMI_WP_IRQSTATUS_RAW) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Reserved | 0 | Reserved |
| 10 | AUDIO_FIFO_SAMPLE_REQ_INTR | | Settable raw status for audio events |
| | | R0 | No event pending |
| | | W0 | No action |
| | | R1 | IRQ event pending |
| | | W1 | Set event |
| 9 | AUDIO_FIFO_OVERFLOW_INTR | | Settable raw status for audio events |
| | | R0 | No event pending |
| | | W0 | No action |
| | | R1 | IRQ event pending |
| | | W1 | Set event |
| 8 | AUDIO_FIFO_UNDERFLOW_INTR | | Settable raw status for audio events |
| | | R0 | No event pending |
| | | W0 | No action |
| | | R1 | IRQ event pending |
| | | W1 | Set event |
| 7-5 | Reserved | 0 | Reserved |
| 4 | OCP_TIME_OUT_INTR | | Settable raw status for OCP timeout interrupt |
| | | R0 | No event pending |
| | | W0 | No action |
| | | R1 | IRQ event pending |
| | | W1 | Set event |
| 3-1 | Reserved | 0 | Reserved |
| 0 | CORE_INTR | | Settable raw status for HDMI Core interrupt |
| | | R0 | Software reset done, no pending action |
| | | W0 | No action |
| | | R1 | Software reset ongoing |
| | | W1 | Set event |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.4 Interrupt Status Register (HDMI_WP_IRQSTATUS)

The interrupt status register is shown in Figure 13-13 and described in Table 13-21.

**Figure 13-13. Interrupt Status Register (HDMI_WP_IRQSTATUS)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | AUDIO_FIFO_SAMPLE_REQ_ INTR | AUDIO_FIFO_OVERFLOW_INTR | AUDIO_FIFO_UNDERFLOW_INTR |
| R-0h | | | | | R/W1C-0h | R/W1C-0h | R/W1C-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | OCP_TIME_OUT_INTR | Reserved | | | CORE_INTR |
| R-0h | | | R/W1C-0h | R-0h | | | R/W1C-0h |

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear, write 0 has no effect; *-n* = value after reset

**Table 13-21. Interrupt Status Register (HDMI_WP_IRQSTATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Reserved | 0 | Reserved |
| 10 | AUDIO_FIFO_SAMPLE_REQ_INTR | R0 | No event pending |
| | | W0 | No action |
| | | R1 | IRQ event pending |
| | | W1 | Set event |
| 9 | AUDIO_FIFO_OVERFLOW_INTR | R0 | No event pending |
| | | W0 | No action |
| | | R1 | Clear pending event, if any |
| | | W1 | Set event |
| 8 | AUDIO_FIFO_UNDERFLOW_INTR | R0 | No event pending |
| | | W0 | No action |
| | | R1 | Clear pending event, if any |
| | | W1 | Set event |
| 7-5 | Reserved | 0 | Reserved |
| 4 | OCP_TIME_OUT_INTR | R0 | No event pending |
| | | W0 | No action |
| | | R1 | Clear pending event, if any |
| | | W1 | Set event |
| 3-1 | Reserved | 0 | Reserved |
| 0 | CORE_INTR | R0 | Software reset done, no pending action |
| | | W0 | No action |
| | | R1 | Clear pending event, if any |
| | | W1 | Set event |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.5 Interrupt Enable Register (HDMI_WP_IRQENABLE_SET)

The interrupt enable register is shown in Figure 13-14 and described in Table 13-22.

**Figure 13-14. Interrupt Enable Register (HDMI_WP_IRQENABLE_SET)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|
| Reserved | | ENABLE_SET_AUDIO_FIFO_SAMPLE_REQ_INTR | ENABLE_SET_AUDIO_FIFO_OVERFLOW_INTR | ENABLE_SET_AUDIO_FIFO_UNDERFLOW_INTR |
| R-0h | | R/W1S-0h | R/W1S-0h | R/W1S-0h |

| 7 | 5 | 4 | 3 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | ENABLE_SET_OCP_TIME_OUT_INTR | Reserved | | ENABLE_SET_CORE_INTR |
| R-0h | | R/W1S-0h | R-0h | | R/W1S-0h |

LEGEND: R/W = Read/Write; R = Read only; W1S = Write 1 to set, write 0 has no effect; -*n* = value after reset

**Table 13-22. Interrupt Enable Register (HDMI_WP_IRQENABLE_SET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Reserved | 0 | Reserved |
| 10 | ENABLE_SET_AUDIO_FIFO_SAMPLE_REQ_INTR | | Enable for audio interrupt events for sample request. Generated only in IRQ mode (instead of DMA mode default value) |
| | | R0 | Interrupt disabled |
| | | W0 | No action |
| | | R1 | Interrupt enabled |
| | | W1 | Enable interrupt |
| 9 | ENABLE_SET_AUDIO_FIFO_OVERFLOW_INTR | | Enable for audio interrupt events for FIFO overflow |
| | | R0 | Interrupt disabled |
| | | W0 | No action |
| | | R1 | Interrupt enabled |
| | | W1 | Enable interrupt |
| 8 | ENABLE_SET_AUDIO_FIFO_UNDERFLOW_INTR | | Enable for audio interrupt events for FIFO underflow |
| | | R0 | Interrupt disabled |
| | | W0 | No action |
| | | R1 | Interrupt enabled |
| | | W1 | Enable interrupt |
| 7-5 | Reserved | 0 | Reserved |
| 4 | ENABLE_SET_OCP_TIME_OUT_INTR | | Enable for interrupt events for OCP timeout interrupt |
| | | R0 | Interrupt disabled |
| | | W0 | No action |
| | | R1 | Interrupt enabled |
| | | W1 | Enable interrupt |
| 3-1 | Reserved | 0 | Reserved |
| 0 | ENABLE_SET_CORE_INTR | | Enable for audio interrupt events for core interrupt |
| | | R0 | Interrupt disabled |
| | | W0 | No action |
| | | R1 | Interrupt enabled |
| | | W1 | Enable interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.6 Interrupt Disable Register (HDMI_WP_IRQENABLE_CLEAR)

The interrupt disable register is shown in Figure 13-15 and described in Table 13-23.

**Figure 13-15. Interrupt Disable Register (HDMI_WP_IRQENABLE_CLEAR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | ENABLE_CLEAR_AUDIO_FIFO_SAMPLE_REQ_INTR | ENABLE_CLEAR_AUDIO_FIFO_OVERFLOW_INTR | ENABLE_CLEAR_AUDIO_FIFO_UNDERFLOW_INTR |
| R-0h | | | | | R/W1C-0h | R/W1C-0h | R/W1C-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | ENABLE_CLEAR_OCP_TIME_OUT_INTR | Reserved | | | ENABLE_CLEAR_CORE_INTR |
| R-0h | | | R/W1C-0h | R-0h | | | R/W1C-0h |

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear, write 0 has no effect; *-n* = value after reset

**Table 13-23. Interrupt Disable Register (HDMI_WP_IRQENABLE_CLEAR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Reserved | 0 | Reserved |
| 10 | ENABLE_CLEAR_AUDIO_FIFO_SAMPLE_REQ_INTR | R0<br>W0<br>R1<br>W1 | Interrupt disabled<br>No action<br>Interrupt enabled<br>Enable interrupt |
| 9 | ENABLE_CLEAR_AUDIO_FIFO_OVERFLOW_INTR | R0<br>W0<br>R1<br>W1 | Interrupt disabled<br>No action<br>Interrupt enabled<br>Enable interrupt |
| 8 | ENABLE_CLEAR_AUDIO_FIFO_UNDERFLOW_INTR | R0<br>W0<br>R1<br>W1 | Interrupt disabled<br>No action<br>Interrupt enabled<br>Enable interrupt |
| 7-5 | Reserved | 0 | Reserved |
| 4 | ENABLE_CLEAR_OCP_TIME_OUT_INTR | R0<br>W0<br>R1<br>W1 | Interrupt disabled<br>No action<br>Interrupt enabled<br>Enable interrupt |
| 3-1 | Reserved | 0 | Reserved |
| 0 | ENABLE_CLEAR_CORE_INTR | R0<br>W0<br>R1<br>W1 | Interrupt disabled<br>No action<br>Interrupt enabled<br>Enable interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.7 Glitch Filter Register (HDMI_WP_DEBOUNCE)

The glitch filter register is shown in Figure 13-16 and described in Table 13-24.

**Figure 13-16. Glitch Filter Register (HDMI_WP_DEBOUNCE)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 14 | 13 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | RXDET | | Reserved | | LINE5VSHORT | |
| R-0h | | R/W-14h | | R-0h | | R/W-14h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-24. Glitch Filter Register (HDMI_WP_DEBOUNCE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-14 | Reserved | 0 | Reserved |
| 13-8 | RXDET | | Glitch Filter for RXDET Input. |
| | | 0 | Disabled |
| | | 1-3Fh | From 1–63, size of the glitch filtered based on OCP clock frequency |
| 7-6 | Reserved | 0 | Reserved |
| 5-0 | LINE5VSHORT | | Glitch filter for line 5v short input. |
| | | 0 | Disabled |
| | | 1-3Fh | From 1–63, size of the glitch filtered based on OCP clock frequency |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.8 Configuration of HDMI Wrapper Video Register (HDMI_WP_VIDEO_CFG)

The configuration of HDMI wrapper video register is shown in Figure 13-17 and described in Table 13-25.

**Figure 13-17. Configuration of HDMI Wrapper Video Register (HDMI_WP_VIDEO_CFG)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | PACKING_MODE | | |
| R-20h | | | | | R/W-0h | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | CORE_VSYNC_INV | CORE_HSYNC_INV | Reserved | | MODE | |
| R-0h | | R/W-0h | R/W-0h | R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-25. Configuration of HDMI Wrapper Video Register (HDMI_WP_VIDEO_CFG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Reserved | 20h | Reserved |
| 10-8 | PACKING_MODE | 0 | Used to pack 10-bits RGB or YUV444 input data in case the video input is not already packed. Will pack video_data[35:26] - video_data[23:14] - video_data[11:2] |
| | | 1h | Used to pack 24-bits RGB, 24-bit YUV444, 16-bit YUV422 or 24-bit YUV422 input data in case the video input is not already packed. Will pack video_data[35:28] - video_data[23:16] - video_data[11:4] |
| | | 2h | Used to pack 20-bits YUV422 input data in case the video input is not already packed. Will pack video_data[35:28] - video_data[23:16] - video_data[11:10] - video_data[7:6] |
| | | 3h-6h | Reserved |
| | | 7h | No change on input video_data lines. The video_data[35:0] is provided to the core without any change on it. |
| 7-6 | Reserved | 0 | Reserved |
| 5 | CORE_VSYNC_INV | | Enables to invert or not the VSYNC signal provided to the HDMI core |
| | | 0 | VSYNC is unchanged |
| | | 1 | VSYNC is inverted |
| 4 | CORE_HSYNC_INV | | Enables to invert or not the HSYNC signal provided to the HDMI core |
| | | 0 | HSYNC is unchanged |
| | | 1 | HSYNC is inverted |
| 3-2 | Reserved | 0 | Reserved |
| 1-0 | MODE | 0 | Always program to 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.9 Configuration of Clocks Register (HDMI_WP_CLK)

The configuration of clocks register is shown in Figure 13-18 and described in Table 13-26.

**Figure 13-18. Configuration of Clocks Register (HDMI_WP_CLK)**

| 31 | 17 | 16 |
|---|---|---|
| Reserved | | OCP_TIME_OUT_DIS |
| R-0h | | R/W-0h |

| 15 | 6 | 5 | 0 |
|---|---|---|---|
| Reserved | | CEC_DIV | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-26. Configuration of Clocks Register (HDMI_WP_CLK) Field Descriptions**

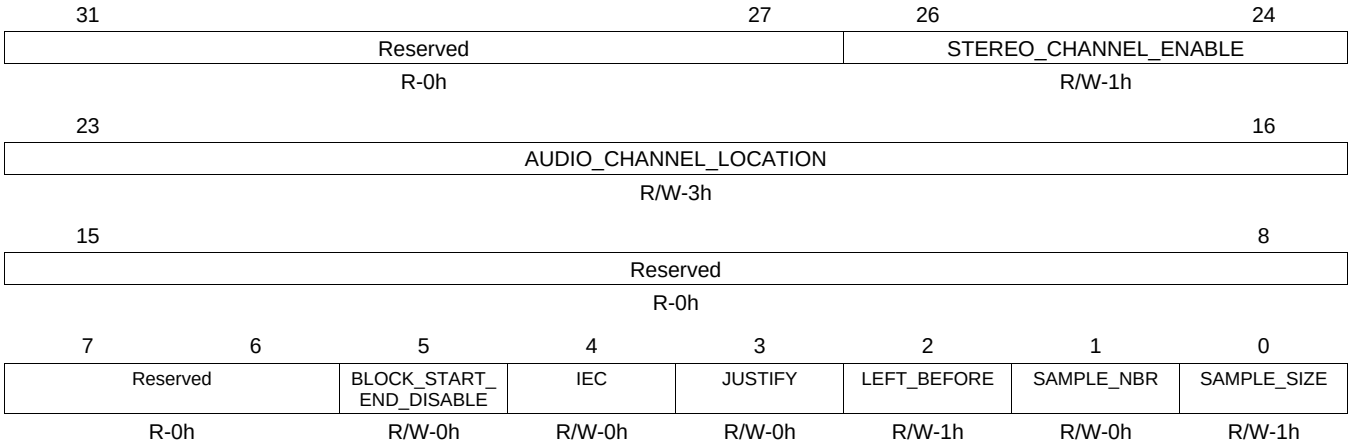
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved |
| 16 | OCP_TIME_OUT_DIS | | Timeout in case CEC_DDC_CLK not provided. |
| | | 0 | Timeout after 4095 OCP clock cycles after inactivity. HDMI Core register interface (due to CEC_DDC_CLK not provided to HDMI). An interrupt is generated. No error response is provided on the OCP interface. |
| | | 1 | No timeout capability |
| 15-6 | Reserved | 0 | Reserved |
| 5-0 | CEC_DIV | | Defines the divisor value to be used for the generation of the CEC clock (1 MHz) from the input CEC_DDC clock (48 MHz). If 48 MHz is provided, the division by 24 is required (18h) to get the expected CEC clock speed (2 MHz) The valid values are from 0 to 63. |
| | | 0 | Gated |
| | | 1 | Free-running |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.10 Audio Configuration in FIFO Register (HDMI_WP_AUDIO_CFG)

The audio configuration in FIFO register is shown in Figure 13-19 and described in Table 13-27.

**Figure 13-19. Audio Configuration in FIFO Register (HDMI_WP_AUDIO_CFG)**




LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-27. Audio Configuration in FIFO Register (HDMI_WP_AUDIO_CFG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-27 | Reserved | 0 | Reserved |
| 26-24 | STEREO_CHANNEL_ENABLE | | This field contains the number of stereo channels enabled in the HDMI_CORE module. Must be set by the software and aligned with the register set in the HDMI_core. |
| | | 0 | No stereo channel enabled in the HDMI_core module |
| | | 1h | 1 stereo channel enabled in the HDMI_core module |
| | | 2h | 2 stereo channels enabled in the HDMI_core module |
| | | 3h | 3 stereo channels enabled in the HDMI_core module |
| | | 4h | 4 stereo channels enabled in the HDMI_core module |
| | | 5h-7h | Reserved |
| 23-16 | AUDIO_CHANNEL_LOCATION | 3h | This field contains which channels are active. It is used also to define which registers are stuffed with zero. |
| 15-6 | Reserved | 0 | Reserved |
| 5 | BLOCK_START_END_DISABLE | | This field disables or not the block end start generation. This field does not affect the pcm format. |
| | | 0 | Block signals are generated. |
| | | 1 | Block signals are not generated; Block start is set to high and block end to low |
| 4 | IEC | | Indicate if the format of the FIFO is compliant with the IEC format. |
| | | 0 | Audio format is L-PCM; Format not aligned with IEC |
| | | 1 | Audio format is IEC 60 958/61937; Format is aligned with IEC format |
| 3 | JUSTIFY | | This field shows the justification (not applicable if IEC format) |
| | | 0 | Justify left |
| | | 1 | Justify right |
| 2 | LEFT_BEFORE | | DO NOT USE—Software always uses LEFT_BEFORE = 1 |
| | | 0 | The first sample is the right |
| | | 1 | The first sample is the left |
| 1 | SAMPLE_NBR | | This field indicates the number of sample per word (32 bits) (not applicable if IEC format) |
| | | 0 | 1 sample per word, 32 bit |
| | | 1 | 2 samples per word, 32 bits (only compatible with sample on 16 bits) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 13-27. Audio Configuration in FIFO Register (HDMI_WP_AUDIO_CFG) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | SAMPLE_SIZE | | Audio sample size, 16 bits or 24 bits (not applicable if IEC format) |
| | | 0 | Sample is on 16 bits |
| | | 1 | Sample is on 24 bits |

### 13.3.1.11 Audio Configuration of DMA Register (HDMI_WP_AUDIO_CFG2)

The audio configuration of DMA register is shown in Figure 13-20 and described in Table 13-28.

**Figure 13-20. Audio Configuration of DMA Register (HDMI_WP_AUDIO_CFG2)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| DMA_TRANSFER | | BLOCK_SIZE | |
| R/W-10h | | R/W-C0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-28. Audio Configuration of DMA Register (HDMI_WP_AUDIO_CFG2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-16 | Reserved | Reserved |
| 15-8 | DMA_TRANSFER | Use to control the dma request. When the number of OCP dma access is performed, the DMA request signal is de-asserted (set to low) All dma access are performed on 32 bit in HDMI_WP_AUDIO_DATA register Encoded value (from 0 to 255). The value 0 is invalid. |
| 7-0 | BLOCK_SIZE | Define the block size if audio sample are compressed default value is 192 to match the IEC 60958 standard. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.12 Audio FIFO Control Register (HDMI_WP_AUDIO_CTRL)

The audio FIFO control register is shown in Figure 13-21 and described in Table 13-29.

**Figure 13-21. Audio FIFO Control Register (HDMI_WP_AUDIO_CTRL)**

| 31 | 30 | 29 26 | 25 24 |
|---|---|---|---|
| WRAPPER_ENABLE | CORE_REQ_ENABLE | Reserved | NUMBER_OF_SAMPLE |
| R/W-0h | R/W-0h | R-0h | R-0h |

| 23 16 |
|---|
| NUMBER_OF_SAMPLE |
| R-0h |

| 15 10 | 9 | 8 |
|---|---|---|
| Reserved | DMA_OR_IRQ | THRESHOLD_VALUE |
| R-0h | R/W-0h | R/W-0h |

| 7 0 |
|---|
| THRESHOLD_VALUE |
| R/W-10h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-29. Audio FIFO Control Register (HDMI_WP_AUDIO_CTRL) Field Descriptions**

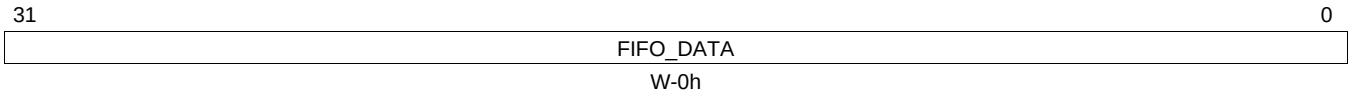
| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | WRAPPER_ENABLE | | This filed enable the audio wrapper. When disable the audio path is under reset Register setting are not affected by the enable. |
| | | 0 | Wrapper is disabled |
| | | 1 | Wrapper is enabled |
| 30 | CORE_REQ_ENABLE | 0 | Enables the Audio data request generated by the core. This is used to start the audio data transmission from the wrapper to the core. This must be enabled only after the 5th VSYNC is generated to ensure that the Audio and Video are in Sync. |
| 29-26 | Reserved | 0 | Reserved |
| 25-16 | NUMBER_OF_SAMPLE | 0 | Shows the number of valid sample (16 or 24 bits) in the FIFO (depends of the FIFO format setting) |
| 15-10 | Reserved | 0 | Reserved |
| 9 | DMA_OR_IRQ | | Indicated if the threshold generates a DMA or an IRQ |
| | | 0 | When the threshold is reached a DMA req will be generated |
| | | 1 | When the threshold is reached an IRQ will be generated |
| 8-0 | THRESHOLD_VALUE | 10h | The DMA/IRQ is generated if the number of sample (24 or 16 bit) in the FIFO is less or equal to the threshold value. The maximum threshold is 511 and the minimum is 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.1.13 TX Data of FIFO Register (HDMI_WP_AUDIO_DATA)

The TX data of FIFO register is shown in Figure 13-22 and described in Table 13-30.

**Figure 13-22. TX Data of FIFO Register (HDMI_WP_AUDIO_DATA)**




LEGEND: W = Write only; *-n* = value after reset

**Table 13-30. TX Data of FIFO Register (HDMI_WP_AUDIO_DATA) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | FIFO_DATA | Audio TX DATA. Can be accessed in 32-bit mode only. Read returns 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2 HDMI Core System Registers

Table 13-31 lists the HDMI core system registers.

**Table 13-31. HDMI Core System Registers**

| Address Offset | Acronym | Register Name |
|---|---|---|
| 00h | VND_IDL | Vendor ID Register |
| 04h | VND_IDH | Vendor ID Register |
| 08h | DEV_IDL | Device ID Register |
| 0Ch | DEV_IDH | Device ID Register |
| 10h | DEV_REV | Device Revision Register |
| 14h | SRST | Software Reset Register |
| 20h | SYS_CTRL1 | System Control Register 1 |
| 24h | SYS_STAT | System Status Register |
| 28h | SYS_CTRL3 | Legacy Registers |
| 34h | DCTL | Data Control Register |
| 3Ch | HDCP_CTRL | HDCP Control Register |
| 40h-50h | BKSV__0 - BKSV__4 | HDCP BKSV Register |
| 54h-70h | AN__0 - AN__7 | HDCP AN Register |
| 74h-84h | AKSV__0 - AKSV__4 | HDCP AKSV Register |
| 88h | RI1 | HDCP Ri Register |
| 8Ch | RI2 | HDCP Ri Register |
| 90h | RI_128_COMP | HDCP Ri 128 Compare Register |
| 94h | I_CNT | HDCP I Counter Register |
| 98h | RI_STAT | Ri Status Register |
| 9Ch | RI_CMD | Ri Command Register |
| A0h | RI_START | Ri Line Start Register |
| A4h | RI_RX_L | Ri From RX Registers |
| A8h | RI_RX_H | Ri From RX Registers |
| ACh | RI_DEBUG | Ri Debug Registers |
| C8h | DE_DLY | VIDEO DE Delay Register |
| C8h | DE_DLY | VIDEO DE Delay Register |
| CCh | DE_CTRL | VIDEO DE Control Register |
| D0h | DE_TOP | VIDEO DE Top Register |
| D8h | DE_CNTL | VIDEO DE Count Register |
| DCh | DE_CNTH | VIDEO DE Count Register |
| E0h | DE_LINL | VIDEO DE Line Register |
| E4h | DE_LINH_1 | VIDEO DE Line Register |
| E8h | HRES_L | Video H Resolution Register |
| ECh | HRES_H | Video H Resolution Register |
| F0h | VRES_L | Video V Resolution Register |
| F4h | VRES_H | Video V Resolution Register |
| F8h | IADJUST | Video Interlace Adjustment Register |
| FCh | POL_DETECT | Video SYNC Polarity Detection Register |
| 100h | HBIT_2HSYNC1 | Video Hbit to HSYNC Register |
| 104h | HBIT_2HSYNC2 | Video Hbit to HSYNC Register |
| 108h | FLD2_HS_OFSTL | Video Field2 HSYNC Offset Register |
| 10Ch | FLD2_HS_OFSTH | Video Field2 HSYNC Offset Register |
| 110h | HWIDTH1 | Video HSYNC Length Register |
| 114h | HWIDTH2 | Video HSYNC Length Register |
| 118h | VBIT_TO_VSYNC | Video Vbit to VSYNC Register |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 13-31. HDMI Core System Registers (continued)**

| Address Offset | Acronym | Register Name |
|---|---|---|
| 11Ch | VWIDTH | Video VSYNC Length Register |
| 120h | VID_CTRL | Video Control Register |
| 124h | VID_ACEN | Video Action Enable Register |
| 128h | VID_MODE | Video Mode1 Register |
| 12Ch | VID_BLANK1 | Video Blanking Registers |
| 130h | VID_BLANK2 | Video Blanking Registers |
| 134h | VID_BLANK3 | Video Blanking Registers |
| 138h | DC_HEADER | Deep Color Header Register |
| 13Ch | VID_DITHER | Video Mode2 Register |
| 140h | RGB2XVYCC_CT | RGB_2_xvYCC control Register |
| 144h | R2Y_COEFF_LOW | RGB_2_xvYCC Conversion R_2_Y Register |
| 148h | R2Y_COEFF_UP | RGB_2_xvYCC Conversion R_2_Y Register |
| 14Ch | G2Y_COEFF_LOW | RGB_2_xvYCC Conversion G_2_Y Register |
| 150h | G2Y_COEFF_UP | RGB_2_xvYCC Conversion G_2_Y Register |
| 154h | B2Y_COEFF_LOW | RGB_2_xvYCC Conversion B_2_Y Register |
| 158h | B2Y_COEFF_UP | RGB_2_xvYCC Conversion B_2_Y Register |
| 15Ch | R2CB_COEFF_LOW | RGB_2_xvYCC Conversion R_2_Cb Register |
| 160h | R2CB_COEFF_UP | RGB_2_xvYCC Conversion R_2_Cb Register |
| 164h | G2CB_COEFF_LOW | RGB_2_xvYCC Conversion G_2_Cb Register |
| 168h | G2CB_COEFF_UP | RGB_2_xvYCC Conversion G_2_Cb Register |
| 16Ch | B2CB_COEFF_LOW | RGB_2_xvYCC Conversion B_2_Cb Register |
| 170h | B2CB_COEFF_UP | RGB_2_xvYCC Conversion B_2_Cb Register |
| 174h | R2CR_COEFF_LOW | RGB_2_xvYCC Conversion R_2_Cr Register |
| 178h | R2CR_COEFF_UP | RGB_2_xvYCC Conversion R_2_Cr Register |
| 17Ch | G2CR_COEFF_LOW | RGB_2_xvYCC Conversion G_2_Cr Register |
| 180h | G2CR_COEFF_UP | RGB_2_xvYCC Conversion G_2_Cr Register |
| 184h | B2CR_COEFF_LOW | RGB_2_xvYCC Conversion B_2_Cr Register |
| 188h | B2CR_COEFF_UP | RGB_2_xvYCC Conversion B_2_Cr Register |
| 18Ch | RGB_OFFSET_LOW | RGB_2_xvYCC RGB Input Offset Register |
| 190h | RGB_OFFSET_UP | RGB_2_xvYCC RGB Input Offset Register |
| 194h | Y_OFFSET_LOW | RGB_2_xvYCC Conversion Y Output Offset Register |
| 198h | Y_OFFSET_UP | RGB_2_xvYCC Conversion Y Output Offset Register |
| 19Ch | CBCR_OFFSET_LOW | RGB_2_xvYCC Conversion CbCr Output Offset Register |
| 1A0h | CBCR_OFFSET_UP | RGB_2_xvYCC Conversion CbCr Output Offset Register |
| 1C0h | INTR_STATE | Interrupt State Register |
| 1C4h | INTR1 | Interrupt Source Register |
| 1C8h | INTR2 | Interrupt Source Register |
| 1CCh | INTR3 | Interrupt Source Register |
| 1D0h | INTR4 | Interrupt Source Register |
| 1D4h | INT_UNMASK1 | Interrupt Unmask Register |
| 1D8h | INT_UNMASK2 | Interrupt Unmask Register |
| 1DCh | INT_UNMASK3 | Interrupt Unmask Register |
| 1E0h | INT_UNMASK4 | Interrupt Unmask Register |
| 1E4h | INT_CTRL | Interrupt Control Register |
| 240h | XVYCC2RGB_CTL | xvYCC_2_RGB Control Register |
| 244h | Y2R_COEFF_LOW | xvYCC_2_RGB Conversion Y_2_R Register |
| 248h | Y2R_COEFF_UP | xvYCC_2_RGB Conversion Y_2_R Register |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 13-31. HDMI Core System Registers (continued)**

| Address Offset | Acronym | Register Name |
|---|---|---|
| 24Ch | CR2R_COEFF_LOW | xvYCC_2_RGB Conversion Cr_2_R Register |
| 250h | CR2R_COEFF_UP | xvYCC_2_RGB Conversion Cr_2_R Register |
| 254h | CB2B_COEFF_LOW | xvYCC_2_RGB Conversion Cb_2_B Register |
| 258h | CB2B_COEFF_UP | xvYCC_2_RGB Conversion Cb_2_B Register |
| 25Ch | CR2G_COEFF_LOW | xvYCC_2_RGB Conversion Cr_2_G Register |
| 260h | CR2G_COEFF_UP | xvYCC_2_RGB Conversion Cr_2_G Register |
| 264h | CB2G_COEFF_LOW | xvYCC_2_RGB Conversion Cb_2_G Register |
| 268h | CB2G_COEFF_UP | xvYCC_2_RGB Conversion Cb_2_G Register |
| 26Ch | YOFFSET1_LOW | xvYCC_2_RGB Conversion Y Offset Register |
| 270h | YOFFSET1_UP | xvYCC_2_RGB Conversion Y Offset Register |
| 274h | OFFSET1_LOW | xvYCC_2_RGB Conversion Offset1 Register |
| 278h | OFFSET1_MID | xvYCC_2_RGB Conversion Offset1 Register |
| 27Ch | OFFSET1_UP | xvYCC_2_RGB Conversion Offset1 Register |
| 280h | OFFSET2_LOW | xvYCC_2_RGB Conversion Offset2 Register |
| 284h | OFFSET2_UP | xvYCC_2_RGB Conversion Offset2 Register |
| 288h | DCLEVEL_LOW | xvYCC_2_RGB Conversion DC Level Register |
| 28Ch | DCLEVEL_UP | xvYCC_2_RGB Conversion DC Level Register |
| 3B0h | DDC_MAN | DDC I2C Manual Register |
| 3B4h | DDC_ADDR | DDC I2C Target Slave Address Register |
| 3B8h | DDC_SEGM | DDC I2C Target Segment Address Register |
| 3BCh | DDC_OFFSET | DDC I2C Target Offset Address Register |
| 3C0h | DDC_COUNT1 | DDC I2C Data Count Register |
| 3C4h | DDC_COUNT2 | DDC I2C Data Count Register |
| 3C8h | DDC_STATUS | DDC I2C Status Register |
| 3CCh | DDC_CMD | DDC I2C Command Register |
| 3D0h | DDC_DATA | DDC I2C Data Register |
| 3D4h | DDC_FIFOCNT | DDC I2C FIFO Count Register |
| 3E4h | EPST | ROM Status Register |
| 3E8h | EPCM | ROM Command Register |

### 13.3.2.1 Vendor IDL Register (VND_IDL)

The vendor IDL register is shown in Figure 13-23 and described in Table 13-32.

**Figure 13-23. Vendor ID Register (VND_IDL)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | VND_IDL | |
| R-0h | | R-1h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-32. Vendor ID Register (VND_IDL) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | VND_IDL | Vendor ID low byte. Provides unique vendor identification through I2C. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.2 Vendor IDH Register (VND_IDH)

The vendor IDH register is shown in Figure 13-24 and described in Table 13-33.

**Figure 13-24. Vendor ID Register (VND_IDH)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | VND_IDH | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-33. Vendor ID Register (VND_IDH) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | VND_IDH | Vendor ID high byte. Provides unique vendor identification through I2C. |

### 13.3.2.3 Device IDL Register (DEV_IDL)

The device IDL register is shown in Figure 13-25 and described in Table 13-34.

**Figure 13-25. Device IDL Register (DEV_IDL)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DEV_IDL | |
| R-0h | | R-36h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-34. Device IDL Register (DEV_IDL) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DEV_ID | Device ID low byte. Provides unique vendor identification through I2C. |

### 13.3.2.4 Device IDH Register (DEV_IDH)

The device IDH register is shown in Figure 13-26 and described in Table 13-35.

**Figure 13-26. Device IDH Register (DEV_IDH)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DEV_IDH | |
| R-0h | | R-97h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-35. Device IDH Register (DEV_IDH) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DEV_ID | Device ID high byte. Provides unique vendor identification through I2C. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.5 Device Revision Register (DEV_REV)

The device revision register is shown in Figure 13-27 and desribed in Table 13-36.

**Figure 13-27. Device Revision Register (DEV_REV)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DEV_REV | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-36. Device Revision Register (DEV_REV) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DEV_REV | Allows distinction between revisions of same device. |

### 13.3.2.6 Software Reset Register (SRST)

The software reset register (SRST) is shown in Figure 13-28 and described in Table 13-37.

**Figure 13-28. Software Reset Register (SRST)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | FIFORST | SWRST |
| R-0h | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-37. Software Reset Register (SRST) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1 | FIFORST | | Audio FIFO reset |
| | | 0 | Normal operation |
| | | 1 | Reset (flush) audio FIFO |
| 0 | SWRST | | Software reset |
| | | 0 | Normal operation |
| | | 1 | Reset all sections, including audio FIFO but not writable registers or HDCP |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.7 System Control Register 1 (SYS_CTRL1)

The system control register 1 is shown in Figure 13-29 and described in Table 13-38.

**Figure 13-29. System Control Register 1 (SYS_CTRL1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | VSYNC | VEN | HEN | Reserved | BSEL | EDGE | PD |
| R-0h | R-x | R/W-1h | R/W-1h | R-0h | R/W-1h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; x = value is indeterminate

**Table 13-38. System Control Register 1 (SYS_CTRL1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | VSYNC | x | The current status of the VSYNC input pin. Refer to the INTR2 register for an interrupt tied to VSYNC active edge. |
| 5 | VEN | | VSYNC enable |
| | | 0 | Fixed LOW |
| | | 1 | Follow VSYNC input |
| 4 | HEN | | HSYNC enable |
| | | 0 | Fixed LOW |
| | | 1 | Follow HSYNC input |
| 3 | Reserved | 0 | Reserved |
| 2 | BSEL | | Input bus select |
| | | 0 | 12-bit Data Bus |
| | | 1 | 24-bit Data Bus |
| 1 | EDGE | | Edge select |
| | | 0 | Latch Input on Falling Edge |
| | | 1 | Latch Input on Rising Edge |
| 0 | PD | | Power down mode. Most other register values are not affected by assertion of the PD bit. |
| | | 0 | Interrupts are in power-down mode |
| | | 1 | Normal operation |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.8 System Status Register (SYS_STAT)

The system status register is shown in Figure 13-30 and described in Table 13-39.

**Figure 13-30. System Status Register (SYS_STAT)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VLOW | Reserved | | | | RSEN | HPD | P_STABLE |
| R-1h | R-0h | | | | R-x | R-X | R-x |

LEGEND: R = Read only; -*n* = value after reset; x = value is indeterminate

**Table 13-39. System Status Register (SYS_STAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | VLOW | 1 | VREF mode. Always HIGH. |
| 6-4 | Reserved | 0 | Reserved |
| 2 | RSEN | | Receiver sense (works in DC-coupled systems only)<br>RSEN is active when the TMDS link is terminated, usually into a powered-on TMDS receiver chip. An active RSEN implies an active Hot Plug Detect; as the link must also be physically connected. |
| | | 0 | No receiver is connected |
| | | 1 | Receiver is connected and powered on |
| 1 | HPD | x | Hot plug detect. The state of the hot plug detect pin can be read here. |
| 0 | P_STABLE | x | IDCK (io_pclkpin) to TMDS clock (v_ck2x) is stable and the Transmitter can send reliable data on the TMDS link. A change to the IDCK sets this bit LOW. After a subsequent LOW to HIGH transition, indicating a stable input clock, a software reset is recommended. |

### 13.3.2.9 System Control Register 3 (SYS_CTRL3)

The system control register 3 is shown in Figure 13-31 and shown in Table 13-40.

**Figure 13-31. System Control Register 3 (SYS_CTRL3)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | CTL | | Reserved |
| R-0h | | | | | | | | | | | | | R/W-0h | | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-40. System Control Register 3 (SYS_CTRL3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-3 | Reserved | Reserved |
| 2-1 | CTL | The states of these control bits are transmitted across the TMDS link during blanking times for DVI 1.0 mode only. |
| 0 | Reserved | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.10 Data Control Register (DCTL)

The data control register is shown in Figure 13-32 and described in Table 13-41.

**Figure 13-32. Data Control Register (DCTL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | VID_BLANK | AUD_MUTE | HDCP_SEL |
| R-0h | | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-41. Data Control Register (DCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2 | VID_BLANK | 0 | Normal operation (video output is not blanked) |
| | | 1 | Video output is blanked and the colors sent are those specified in registers VID_BLANK1, VID_BLANK2, and VID_BLANK3. |
| 1 | AUD_MUTE | 0 | Do not send zeroes in audio pocket |
| | | 1 | Send zeroes in audio pocket |
| 0 | HDCP_SEL | 0 | This register gets the value of the pin io_hdcp_sel.<br>When connected to 1 b0; enables firmware to take the decision whether to send unencrypted data or not. By connecting to 1 b1; HDMI Tx will be able to send ONLY encrypted data. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.11 HDCP Control Register (HDCP_CTRL)

The HDCP control register is shown in Figure 13-33 and described in Table 13-42.

**Figure 13-33. HDCP Control Register (HDCP_CTRL)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | ENC_ON | BKSV_ERR | RX_RPTR | TX_ANSTOP | CP_RESTN | RI_RDY | ENC_EN |
| R-0h | R-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-42. HDCP Control Register (HDCP_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | ENC_ON | | Encryption status |
| | | 0 | Encryption disabled or not in progress |
| | | 1 | Encryption enabled and in progress |
| 5 | BKSV_ERR | | BKSV error. To clear BKSV_ERR; the firmware must first set the TX_ANSTOP bit; then perform an authentication twice with a valid BKSV value. |
| | | 0 | No error in BKSV format |
| | | 1 | Error in BKSV format |
| 4 | RX_RPTR | | Repeater. If the HDMI Receiver is in a Repeater, write bit 4 (RX_RPTR) to 1 before beginning the authentication process. Note: This bit is written when the Source device detects an attached HDCP Repeater device. This is a necessary step in the computation of shared values in the HDCP protocol. For more information, sefer section 2.2 in the HDCP 1.0 Specification. |
| | | 0 | Single HDMI Receiver |
| | | 1 | HDMI Receiver is a Repeater |
| 3 | TX_ANSTOP | 0 | AN control. When cleared; the cipher engine is allowed to free run and the AN registers cycle through pseudo-random values. When set; the cipher engines stops and the AN register may be read and initialized in the HDCP-capable Receiver. To clear this bit; toggle the RX_RPTR bit. This bit is also cleared when the hardware is reset or BKSV_ERR is set. |
| 2 | CP_RESTN | | Content protection reset |
| | | 0 | Reset |
| | | 1 | Normal operation |
| 1 | RI_RDY | | Ri Ready |
| | | 0 | Ri first value is not ready |
| | | 1 | Ri first value is ready in the HDMI Transmitter. Cleared by a hardware reset. This bit is also cleared when the first byte of BKSV is written into the HDMI Transmitter (performed at the beginning of the next authentication process). |
| 0 | ENC_EN | | Encryption enable |
| | | 0 | Encryption is disabled |
| | | 1 | Encryption is enabled. See description of HDCP_SEL bit in register DCTL (see Section 13.3.2.10). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.12 HDCP BKSV Register (BKSV__0-BKSV__4)

The HDCP BKSV register is shown in Figure 13-34 and described in Table 13-43.

**Figure 13-34. HDCP BKSV Register (BKSV__0-BKSV__4)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | BKSV | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-43. HDCP BKSV Register (BKSV__0-BKSV__4) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | BKSV | Write : Written with the HDCP receiver key selection vector register value. Writing 5th BKSV byte triggers the authentication logic in the HDMI Transmitter. Write this byte last. |

### 13.3.2.13 HDCP AN Register (AN__0-AN__7)

The HDCP AN register is shown in Figure 13-35 and described in Table 13-44.

**Figure 13-35. HDCP AN Register (AN__0-AN__7)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AN | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-44. HDCP AN Register (AN__0-AN__7) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AN | AN value is an HDCP 64-Bit pseudo-random value. |

### 13.3.2.14 HDCP AKSV Register (AKSV__0-AKSV__4)

The HDCP AKSV registe is shown in Figure 13-36 and described in Table 13-45.

**Figure 13-36. HDCP AKSV Register (AKSV__0-AKSV__4)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AKSV | |
| R-0h | | R-x | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset; x = value is indeterminate

**Table 13-45. HDCP AKSV Register (AKSV__0-AKSV__4) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AKSV | HDCP-capable transmitter s key selection vector. Fifth AKSV byte triggers the authentication logic in the receiver. Write this byte last. |

### 13.3.2.15 HDCP Ri1 Register (RI1)

The HDCP Ri1 register is shown in Figure 13-37 and described in Table 13-46.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 13-37. HDCP Ri1 Register (RI1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RI | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-46. HDCP Ri Register (RI1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | RI1 | R1i register. The value of this register should be read and compared with the HDMI Receiver s Ri value. |

### 13.3.2.16 HDCP Ri2 Register (RI2)

The HDCP Ri2 register is shown in Figure 13-38 and described in Table 13-47.

**Figure 13-38. HDCP Ri2 Register (RI2)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RI2 | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-47. HDCP Ri2 Register (RI2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | RI2 | Ri2 register. The value of this register should be read and compared with the HDMI Receiver s Ri value. |

### 13.3.2.17 HDCP Ri 128 Compare Register (RI_128_COMP)

The HDCP Ri 128 compare register is shown in Figure 13-39 and described in Table 13-48.

**Figure 13-39. HDCP Ri 128 Compare Register (RI_128_COMP)**

| 31 | 7 | 6 | 0 |
|---|---|---|---|
| Reserved | | RI_128_COMP | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-48. HDCP Ri 128 Compare Register (RI_128_COMP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-7 | Reserved | Reserved |
| 6-0 | RI_128_COMP | Limit counter for Ri comparison. When the frame counter (I_CNT) reaches the index set in this register an RI_128 interrupt is generated. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.18 HDCP I Counter Register (I_CNT)

The HDCP I counter register is shown in Figure 13-40 and described in Table 13-49.

**Figure 13-40. HDCP I Counter Register (I_CNT)**

| 31 | 7 | 6 | 0 |
|---|---|---|---|
| Reserved | | I_CNT | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-49. HDCP I Counter Register (I_CNT) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-7 | Reserved | Reserved |
| 6-0 | I_CNT | Current value of I counter. This register counts frames from 0 to 127, then rolls over to zero. |

### 13.3.2.19 Ri Status Register (RI_STAT)

The Ri status register is shown in Figure 13-41 and described in Table 13-50.

**Figure 13-41. Ri Status Register (RI_STAT)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | RI_STARTED |
| R-0h | | R-0h |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-50. Ri Status Register (RI_STAT) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reserved |
| 0 | RI_STARTED | Ri check started status. This signal is used for handshaking between the firmware and the hardware. After the Ri check is enabled, the hardware waits for the DDC master to finish the current transaction before taking control. After this bit is set, the firmware loses the ability to use the DDC Master, unless it disables Ri Check and resets to 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.20 Ri Command Register (RI_CMD)

The Ri command register is shown in Figure 13-42 and described in Table 13-51.

**Figure 13-42. Ri Command Register (RI_CMD)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | BCAP_EN | RI_EN |
| R-0h | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-51. Ri Command Register (RI_CMD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1 | BCAP_EN | | Enable polling of the BCAP_DONE bit (in the register INTR2).<br>Note: To poll the BCAP_DONE bit; the ENC_EN (in HDPC_CTRL register) bit and the Ri_EN (in RI_CMD register) bit must be enabled on the HDMI Transmitter. |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | RI_EN | | Enable Ri check. Check bit RI_STARTED of the Ri_STAT register for firmware and hardware DDC control handshaking. |
| | | 0 | Disable |
| | | 1 | Enable |

### 13.3.2.21 Ri Line Start Register (RI_START)

The Ri line start register is shown in Figure 13-43 and described in Table 13-52.

**Figure 13-43. Ri Line Start Register (RI_START)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RI_LINE_START | |
| R-0h | | R/W-4h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-52. Ri Line Start Register (RI_START) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | RI_LINE_START | Indicates at what line within frame 127 or 0 to start the Ri Check.<br>Note that the value for this register bit represents the power of 2; the 2 LSBs are 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.22 Ri From RX Registers (Low) (RI_RX_L)

The Ri from RX registers (low) are shown in Figure 13-44 and described in Table 13-53.

**Figure 13-44. Ri From RX Registers (Low) (RI_RX_L)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RI_RX | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-53. Ri From RX Register (Low) (RI_RX_L) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | RI_RX | Ri_RX[15:8] and Ri_RX[7:0]. This value represents the HDMI Receiver s Ri value if any of the Ri Check errors occurred. |

### 13.3.2.23 Ri From RX Registers (High) (RI_RX_H)

The Ri from RX registers (high) are shown in Figure 13-45 and described in Table 13-54.

**Figure 13-45. Ri From RX Registers (High) (RI_RX_H)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RI_RX | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-54. Ri From RX Registers (High) (RI_RX_H) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | RI_RX | Ri_RX[15:8] and Ri_RX[7:0]. This value represents the HDMI Receiver s Ri value if any of the Ri Check errors occurred. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.24 Ri Debug Registers (RI_DEBUG)

The Ri debug registers are shown in Figure 13-46 and described in Table 13-55.

**Figure 13-46. Ri Debug Registers (RI_DEBUG)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RI_DBG_TRASH | RI_DBG_HOLD | Reserved | | | | | |
| R/W-0h | R/W-0h | R-0h | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-55. Ri Debug Registers (RI_DEBUG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | RI_DBG_TRASH | | Ri value trash |
| | | 0 | Continue with regular updates to Ri |
| | | 1 | Force a corruption of the Ri values |
| 6 | RI_DBG_HOLD | | Ri value hold |
| | | 0 | Continue with regular updates to Ri |
| | | 1 | Hold the Ri value steady; stop updating |
| 5-0 | Reserved | 0 | Reserved |

### 13.3.2.25 VIDEO DE Delay Register (DE_DLY)

The VIDEO DE delay register is shown in Figure 13-47 and described in Table 13-56.

**Figure 13-47. VIDEO DE Delay Register (DE_DLY)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DE_DLY[7:0] | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-56. VIDEO DE Delay Register (DE_DLY) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DE_DLY | Width of the area to the left of the active display. The unit of measure is pixels. This register should be set to the sum of (HSYNC width) + (horizontal back porch) + (horizontal left border), and is used only for DE generation.<br>Note that this 12-bit value includes four bits from register DE_CTRL (see Section 13.3.2.26). The valid range is 1-4095. 0 is invalid. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.26 VIDEO DE Control Register (DE_CTRL)

The VIDEO DE control register is shown in Figure 13-48 and described in Table 13-57.

**Figure 13-48. VIDEO DE Control Register (DE_CTRL)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | DE_GEN | VS_POL | HS_POL | DE_DLY[11:8] | | | |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-57. VIDEO DE Control Register (DE_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | DE_GEN | | Generate DE signal |
| | | 0 | Disabled |
| | | 1 | Enabled |
| 5 | VS_POL | | VSYNC polarity |
| | | 0 | Positive polarity (leading edge rises) |
| | | 1 | Negative polarity (leading edge falls) |
| | | | Set this bit to the input VSYNC polarity for the source that provides VSYNC. For embedded syncs, set this bit to the desired VSYNC polarity that is generated from the embedded sync codes. |
| 4 | HS_POL | | HSYNC polarity |
| | | 0 | Positive polarity (leading edge rises) |
| | | 1 | Negative polarity (leading edge falls) |
| | | | Set this bit to the input HSYNC polarity for the source that provides HSYNC. For embedded syncs, set this bit to the desired HSYNC polarity that is generated from the embedded sync codes. |
| 3-0 | DE_DLY | 0 | Width of the area to the left of the active display. The unit of measure is pixels. This register should be set to the sum of (HSYNC width) + (horizontal back porch) + (horizontal left border), and is used only for DE generation. |
| | | | Note that this 12-bit value includes eight bits from register DE_DLY (see Section 13.3.2.25). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.27 VIDEO DE Top Register (DE_TOP)

The VIDEO DE top register is shown in Figure 13-49 and described in Table 13-58.

**Figure 13-49. VIDEO DE Top Register (DE_TOP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DE_TOP | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-58. VIDEO DE Top Register (DE_TOP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DE_TOP | Defines the height of the area above the active display. The unit of measure is lines (HSYNC pulses). This register should be set to the sum of (VSYNC width) + (vertical back porch) + (vertical top border). The valid range is 1-127. 0 is invalid. |

### 13.3.2.28 VIDEO DE Count Register (DE_CNTL)

**Figure 13-50. VIDEO DE Count Register (DE_CNTL)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DE_CNT[7:0] | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

The VIDEO DE count register is shown in Figure 13-50 and described in Table 13-59.

**Table 13-59. VIDEO DE Count Register (DE_CNTL) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DE_CNT | Defines the width of the active display. The unit of measure is pixels. This register should be set to the desired horizontal resolution.<br>Note that this 12-bit value includes four bits from register DE_CNTH (see Section 13.3.2.29). The valid range is 1-4095. 0 is invalid. |

### 13.3.2.29 VIDEO DE Count Register (DE_CNTH)

The VIDEO DE count register is shown in Figure 13-51 and described in Table 13-60.

**Figure 13-51. VIDEO DE Count Register (DE_CNTH)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | DE_CNT[11:8] | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-60. VIDEO DE Count Register (DE_CNTH) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reserved |
| 3-0 | DE_CNT | Defines the width of the active display. The unit of measure is pixels. This register should be set to the desired horizontal resolution.<br>Note that this 12-bit value includes eight bits from register DE_CNTL (see Section 13.3.2.28). The valid range is 1-4095. 0 is invalid. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.30 VIDEO DE Line Register (DE_LINL)

The VIDEO DE line register is shown in Figure 13-52 and described in Table 13-61.

**Figure 13-52. VIDEO DE Line Register (DE_LINL)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DE_LIN[7:0] | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-61. VIDEO DE Line Register (DE_LINL) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DE_LIN | Defines the height of the active display. The unit of measure is lines (HSYNC pulses). Set this register to the desired vertical resolution. For interlaced modes; set this register to the number of lines per field; which is half the overall vertical resolution.<br>Note that this 11-bit value includes three bits from register DE_LINH_1 (see Section 13.3.2.31). The valid range is 1-2047. 0 is invalid. |

### 13.3.2.31 VIDEO DE Line Register (DE_LINH_1)

The VIDEO DE line register is shown in Figure 13-53 and described in Table 13-62.

**Figure 13-53. VIDEO DE Line Register (DE_LINH_1)**

| 31 | 3 | 2 | 0 |
|---|---|---|---|
| Reserved | | DE_LIN[10:8] | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-62. VIDEO DE Line Register (DE_LINH_1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-3 | Reserved | Reserved |
| 2-0 | DE_LIN | Defines the height of the active display. The unit of measure is lines (HSYNC pulses). Set this register to the desired vertical resolution. For interlaced modes; set this register to the number of lines per field; which is half the overall vertical resolution.<br>Note that this 11-bit value includes eight bits from register DE_LINL (see Section 13.3.2.30). The valid range is 1-2047. 0 is invalid. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.32 Video H Resolution Register (HRES_L)

The video H resolution register is shown in Figure 13-54 and described in Table 13-63.

**Figure 13-54. Video H Resolution Register (HRES_L)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | H_RES[7:0] | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-63. Video H Resolution Register (HRES_L) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | H_RES | Measures the time between two HSYNC active edges. The unit of measure is pixels.<br>Note that this 13-bit value includes five bits from register HRES_H (see Section 13.3.2.33). |

### 13.3.2.33 Video H Resolution Register (HRES_H)

The video H resolution register is shown in Figure 13-55 and described in Table 13-64.

**Figure 13-55. Video H Resolution Register (HRES_H)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | H_RES[12:8] | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-64. Video H Resolution Register (HRES_H) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reserved |
| 4-0 | H_RES | Measures the time between two HSYNC active edges. The unit of measure is pixels.<br>Note that this 13-bit value includes eight bits from register HRES_L (see Section 13.3.2.32). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.34 Video V Resolution Register (VRES_L)

The video V resolution register is shown in Figure 13-56 and described in Table 13-65.

**Figure 13-56. Video V Resolution Register (VRES_L)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | V_RES[7:0] | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-65. Video V Resolution Low Register (VRES_L) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | V_RES | Measures the time between two VSYNC active edges. The unit of measure is lines.<br>Note that this 11-bit value includes three bits from register VRES_H (see Section 13.3.2.35). |

### 13.3.2.35 Video V Resolution Register (VRES_H)

The video V resolution register is shown in Figure 13-57 and described in Table 13-66.

**Figure 13-57. Video V Resolution Register (VRES_H)**

| 31 | 3 | 2 | 0 |
|---|---|---|---|
| Reserved | | V_RES[10:8] | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-66. Video V Resolution Register (VRES_H) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-3 | Reserved | Reserved |
| 2-0 | V_RES | Measures the time between two VSYNC active edges. The unit of measure is lines.<br>Note that this 11-bit value includes eight bits from register VRES_L (see Section 13.3.2.34). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.36 Video Interlace Adjustment Register (IADJUST)

The video interface adjustment register is shown in Figure 13-58 and described in Table 13-67.

**Figure 13-58. Video Interlace Adjustment Register (IADJUST)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | DE_ADJ | F2VADJ | F2VOFST |
| R-0h | | R/W-1h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-67. Video Interlace Adjustment Register (IADJUST) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2 | DE_ADJ | | Setting this bit HIGH disables VSYNC adjustments and sets the DE generator to be more compatible with the existing transmitters. Clearing this bit enables detection circuits to locate the position of VSYNC relative to HSYNC and only include HSYNC edges that are greater than 3/4 lines from VSYNC in the line count for DE_TOP. Enable VSYNC detection when not using DE generator. |
| | | 0 | Enable VSYNC |
| | | 1 | Disable VSYNC. |
| 1 | F2VADJ | | VBIT_TO_VSYNC value (in register VBIT_TO_VSYNC) adjust enable |
| | | 0 | The VBIT_TO_VSYNC value is not adjusted. |
| | | 1 | The VBIT_TO_VSYNC value is adjusted during field 2 of an interlace frame according to the setting of the F2VOFST bit. |
| 0 | F2VOFST | | VBIT_TO_VSYNC value (in register VBIT_TO_VSYNC) offset |
| | | 0 | If F2VADJ is set, VBIT_TO_VSYNC is decremented by one during field 2 of an interlace frame. |
| | | 1 | If F2VADJ is set, VBIT_TO_VSYNC is incremented by one during field 2 of an interlace frame. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.37 Video SYNC Polarity Detection Register (POL_DETECT)

The video SYNCH polarity detection register is shown in Figure 13-59 and described in Table 13-68.

**Figure 13-59. Video SYNC Polarity Detection Register (POL_DETECT)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | I_DET | VPOL_DET | HPOL_DET |
| R-0h | | R-0h | R-0h | R-0h |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-68. Video SYNC Polarity Detection Register (POL_DETECT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2 | I_DET | | Interlace detect |
| | | 0 | Non-interlaced video |
| | | 1 | Interlaced video. This bit is set by checking for a varying VSYNC timing characteristic of interlaced modes. |
| 1 | VPOL_DET | | Detected input VSYNC polarity using internal circuit |
| | | 0 | Active high (leading edge rises) |
| | | 1 | Active low (leading edge falls) |
| 0 | HPOL_DET | | Detected input HSYNC polarity using internal circuit |
| | | 0 | Active high (leading edge rises) |
| | | 1 | Active low (leading edge falls) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.38 Video Hbit to HSYNC Register (HBIT_2HSYNC1)

The video Hbit to HSYNC register is shown in Figure 13-60 and described in Table 13-69.

**Figure 13-60. Video Hbit to HSYNC Register (HBIT_2HSYNC1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | HBIT_TO_HSYNC[7:0] | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-69. Video Hbit to HSYNC Register (HBIT_2HSYNC1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | HBIT_TO_HSYNC | Creates HSYNC pulses. Set this register to the delay from the detection of an EAV sequence (H bit change from 1 to 0) to the active edge of HSYNC. The unit of measure is pixels.<br>Note that this 10-bit value includes two bits from register HBIT_2HSYNC2 (see Section 13.3.2.39). The valid range is 1-1023. 0 is invalid. |

### 13.3.2.39 Video Hbit to HSYNC Register (HBIT_2HSYNC2)

The video Hbit to HSYNC register is shown in Figure 13-61 and described in Table 13-70.

**Figure 13-61. Video Hbit to HSYNC Register (HBIT_2HSYNC2)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | HBIT_TO_HSYNC[9:8] | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-70. Video Hbit to HSYNC Register (HBIT_2HSYNC2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-2 | Reserved | Reserved |
| 1-0 | HBIT_TO_HSYNC | Creates HSYNC pulses. Set this register to the delay from the detection of an EAV sequence (H bit change from 1 to 0) to the active edge of HSYNC. The unit of measure is pixels.<br>Note that this 10-bit value includes eight bits from register HBIT_2HSYNC1 (see Section 13.3.2.38). The valid range is 1-1023. 0 is invalid. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.40 Video Field2 HSYNC Offset Register (FLD2_HS_OFSTL)

The video field2 HSYNC offset register is shown in Figure 13-62 and described in Table 13-71.

**Figure 13-62. Video Field2 HSYNC Offset Register (FLD2_HS_OFSTL)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | FIELD2_OFST[7:0] | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-71. Video Field2 HSYNC Offset Register (FLD2_HS_OFSTL) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | FIELD2_OFST | Determines VSYNC pixel offset for the odd field of an interlaced source. Set this to half the number of pixels/line.<br>Note that this 12-bit value includes four bits from register FLD2_HS_OFSTH (see Section 13.3.2.41). The valid range is 1-4095. 0 is invalid. |

### 13.3.2.41 Video Field2 HSYNC Offset Register (FLD2_HS_OFSTH)

**Figure 13-63. Video Field2 HSYNC Offset Register (FLD2_HS_OFSTH)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | FIELD2_OFST[11:8] | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

The video field2 HSYNC offset register is shown in Figure 13-63 and described in Table 13-72.

**Table 13-72. Video Field2 HSYNC Offset Register (FLD2_HS_OFSTH) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reserved |
| 3-0 | FIELD2_OFST | Determines VSYNC pixel offset for the odd field of an interlaced source. Set this to half the number of pixels/line.<br>Note that this 12-bit value includes eight bits from register FLD2_HS_OFSTL (see Section 13.3.2.40). The valid range is 1-4095. 0 is invalid. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.42 Video HSYNC Length Register (HWIDTH1)

The video HSYNC length register is shown in Figure 13-64 and described in Table 13-73.

**Figure 13-64. Video HSYNC Length Register (HWIDTH1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | HWIDTH[7:0] | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-73. Video HSYNC Length Register (HWIDTH1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | HWIDTH | Sets the width of the HSYNC pulses. Set this register to the desired HSYNC pulse width. The unit of measure is pixels.<br>Note that this 10-bit value includes two bits from register HWIDTH2 (see Section 13.3.2.43). The valid range is 1-1023. 0 is invalid. |

### 13.3.2.43 Video HSYNC Length Register (HWIDTH2)

The video HSYNC length register is shown in Figure 13-65 and described in Table 13-74.

**Figure 13-65. Video HSYNC Length Register (HWIDTH2)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | HWIDTH[9:8] | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-74. Video HSYNC Length Register (HWIDTH2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-2 | Reserved | Reserved |
| 1-0 | HWIDTH | Sets the width of the HSYNC pulses. Set this register to the desired HSYNC pulse width. The unit of measure is pixels.<br>Note that this 10-bit value includes eight bits from register HWIDTH1 (see Section 13.3.2.42). The valid range is 1-1023. 0 is invalid. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.44 Video Vbit to VSYNC Register (VBIT_TO_VSYNC)

The video Vbit to VSYNC register is shown in Figure 13-66 and desribed in Table 13-75.

**Figure 13-66. Video Vbit to VSYNC Register (VBIT_TO_VSYNC)**

| 31 | 6 | 5 | 0 |
|---|---|---|---|
| Reserved | | VBIT_TO_VSYNC | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-75. Video Vbit to VSYNC Register (VBIT_TO_VSYNC) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-6 | Reserved | Reserved |
| 5-0 | VBIT_TO_VSYNC | Sets the delay from the detection of V bit changing from 1 to 0 in an EAV sequence; to the asserting edge of VSYNC. The unit of measure is lines. The valid range is 1-63. 0 is invalid. |

### 13.3.2.45 Video VSYNC Length Register (VWIDTH)

The video VSYNC length register is shown in Figure 13-67 and described in Table 13-76.

**Figure 13-67. Video VSYNC Length Register (VWIDTH)**

| 31 | 6 | 5 | 0 |
|---|---|---|---|
| Reserved | | VWIDTH | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-76. Video VSYNC Length Register (VWIDTH) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-6 | Reserved | Reserved |
| 5-0 | VWIDTH | Sets the width of VSYNC pulse. The unit of measure is lines. The valid range is 1-63. 0 is invalid. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.46 Video Control Register (VID_CTRL)

The video control register is shown in Figure 13-68 and described in Table 13-77.

**Figure 13-68. Video Control Register (VID_CTRL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | IFPOL | Reserved | EXTN | CSCSEL | Reserved | | ICLK | |
| R-0h | | R/W-0h | R-0h | R/W-0h | R/W-0h | R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-77. Video Control Register (VID_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | IFPOL | | Invert field polarity. This bit is used when the 656 Flag bit is opposite the standard polarity for Field1 and Field2. Inverting the field polarity causes the sync extraction to format HSYNC and VSYNC properly based on the F bit. In embedded sync mode, the HDMI Transmitter does not detect even from odd field, except based on the setting of the F bit. With explicit syncs, the HDMI transmitter encodes HSYNC and VSYNC across the HDMI/TMDS link regardless of field sequence. |
| | | 0 | Do not invert field bit |
| | | 1 | Invert field bit |
| 6 | Reserved | 0 | Reserved |
| 5 | EXTN | | Extended bit mode |
| | | 0 | All 8-bit input modes |
| | | 1 | All 12-bit 4:2:2 input modes. For 4:2:2 inputs wider than 8 bits but less than 12 bits; the unused bits should be cleared to 0. |
| 4 | CSCSEL | | Color space conversion standard select |
| | | 0 | BT.601 conversion |
| | | 1 | BT.709 conversion |
| 3-2 | Reserved | 0 | Reserved |
| 1-0 | ICLK | | Clock mode<br>If the DEMUX bit in the VID_MODE register is cleared to 0, set the ICLK bit and the pixel replication field of the AVI v2 data byte 5 to the same value.<br>If the DEMUX bit in the VID_MODE register is set to 1, set the pixel replication field of the AVI v2 data byte 5 to the next higher pixel replication rate. For example, if DEMUX = 1 and ICLK = 1, set the pixel replication field of AVI v2 data byte 5 to 3h. |
| | | 0 | Pixel data is not replicated |
| | | 1h | Pixels are replicated once (each sent twice) |
| | | 2h | Reserved |
| | | 3h | Pixels are replicated 4 times (each sent four times) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.47 Video Action Enable Register (VID_ACEN)

The video action enable register is shown in Figure 13-69 and described in Table 13-78.

**Figure 13-69. Video Action Enable Register (VID_ACEN)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WIDE_BUS | | Reserved | CLIP_CS_ID | RANGE_CLIP | RGB_2_YCBCR | RANGE_CMPS | DOWN_SMPL |
| R/W-0h | | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-78. Video Action Enable Register (VID_ACEN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-6 | WIDE_BUS | | Identifies the number of bits per input video channel |
| | | 0 | 8 bits per channel or 24-bit bus mode |
| | | 1h | 10 bits per channel or 30-bit bus mode |
| | | 2h | 12 bits per channel or 36-bit bus mode |
| | | 3h | Reserved |
| 5 | Reserved | 0 | Reserved |
| 4 | CLIP_CS_ID | | Identifies the output color space on the link - used by the Clipper block to determine which way to clip |
| | | 0 | Output color space is RGB 1 |
| | | 1 | Output color space is YCbCr |
| 3 | RANGE_CLIP | | Enable range clip from 16 to 235 (RGB and Y)/ 240 (CbCr) |
| | | 0 | Disable |
| | | 1 | Enable |
| 2 | RGB_2_YCBCR | | Enable RGB to YCbCr color-space converter |
| | | 0 | Disable |
| | | 1 | Enable |
| 1 | RANGE_CMPS | | Enable range compress 0-255 to 16-234 |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | DOWN_SMPL | | Enable down sampler 4:4:4 to 4:2:2 |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.48 Video Mode1 Register (VID_MODE)

The video mode1 register is shown in Figure 13-70 and described in Table 13-79.

**Figure 13-70. Video Mode1 Register (VID_MODE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DITHER_MODE | | DITHER | RANGE | CSC | UPSMP | DEMUX | SYNCEX |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-79. Video Mode1 Register (VID_MODE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-6 | DITHER_MODE | | Identifies the number of bits per output video channel |
| | | 0 | Dither to 8 bits |
| | | 1h | Dither to 10 bits |
| | | 2h | Dither to 12 bits |
| | | 3h | Reserved |
| 5 | DITHER | | Dither enable |
| | | 0 | Dither disabled; the video output is truncated to the output width specified in the DITHER_MODE bit. |
| | | 1 | Dither enabled; the video output is dithered to the output width specified in the DITHER_MODE bit. |
| 4 | RANGE | | Data range 16–235 to 0–255 expansion |
| | | | When this bit is set, the HDMI transmitter expands the range of pixel data values from 16-235 into the full 8-bit range of 0-255. This is suitable for translating input YCbCr data into output RGB data in PC modes that use the complete range. The HDMI Specification allows one non-CEA-861D mode in the first and only 18-byte descriptor of the Sink's EDID 1.3. This is the native resolution of the Sink, which may be RGB. It may be a standard PC resolution (XGA, SXGA, WXGA, and so on), or a specific native resolution. In these cases (or for a Sink with the Type B HDMI connector, which allows multiple PC modes), when the HDMI transmitter receives YCbCr data, the data must be expanded to full range for outputting RGB full-range modes. |
| | | 0 | Disable |
| | | 1 | Enable |
| 3 | CSC | | YcbCr to RGB color space conversion |
| | | 0 | Disable |
| | | 1 | Enable |
| 2 | UPSMP | | Upsampling 4:2:2 to 4:4:4 |
| | | 0 | Disable |
| | | 1 | Enable |
| 1 | DEMUX | | One- to two-data-channel demux |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | SYNCEX | | Embedded sync extraction |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.49 Video Blanking Register (VID_BLANK1)

The video blanking register is shown in Figure 13-71 and described in Table 13-80.

**Figure 13-71. Video Blanking Register (VID_BLANK1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | VID_BLANK1 | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-80. Video Blanking Registers (VID_BLANK1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | VID_BLANK1 | Defines the video blanking value for Channel 1 (Blue). |

### 13.3.2.50 Video Blanking Register (VID_BLANK2)

The video blanking register is shown in Figure 13-72 and described in Table 13-81.

**Figure 13-72. Video Blanking Register (VID_BLANK2)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | VID_BLANK2 | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-81. Video Blanking Register (VID_BLANK2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | VID_BLANK2 | Defines the video blanking value for Channel 2 (Green). |

### 13.3.2.51 Video Blanking Register (VID_BLANK3)

The video blanking register is shown in Figure 13-73 and described in Table 13-82.

**Figure 13-73. Video Blanking Register (VID_BLANK3)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | VID_BLANK3 | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-82. Video Blanking Register (VID_BLANK3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | VID_BLANK3 | Defines the video blanking value for Channel 3 (Red). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.52 Deep Color Header Register (DC_HEADER)

The deep color header register is shown in Figure 13-74 and described in Table 13-83.

**Figure 13-74. Deep Color Header Register (DC_HEADER)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DC_HEADER | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-83. Deep Color Header Register (DC_HEADER) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DC_HEADER | The least-significant byte of the deep color header that sends the TMDS dynamic phase once per frame. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.53 Video Mode2 Register (VID_DITHER)

The video mode2 register is shown in Figure 13-75 and described in Table 13-84.

**Figure 13-75. Video Mode2 Register (VID_DITHER)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | M_D2 | UP2 | 422_EN | D_BC_EN | D_GC_EN | D_RC_EN | DRD |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-84. Video Mode2 Register (VID_DITHER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | M_D2 | | Dither + round option |
| | | 0 | Disable the function |
| | | 1 | Enable the function |
| 5 | UP2 | | Dither + 2'b10 option |
| | | 0 | Disable the function |
| | | 1 | Enable the function |
| 4 | 422_EN | | Enable Mode 4:2:2 for dithering and clipping |
| | | 0 | Disable the function |
| | | 1 | Enable the function |
| 3 | D_BC_EN | | Enable adding random number on Blue channel data. |
| | | 0 | Disable the function |
| | | 1 | Enable the function |
| 2 | D_GC_EN | | Enable adding random number on Green channel data. |
| | | 0 | Disable the function |
| | | 1 | Enable the function |
| 1 | D_RC_EN | | Enable adding random number on Red channel data. |
| | | 0 | Disable the function |
| | | 1 | Enable the function |
| 0 | DRD | | Dither round: Add random number + 2b10 then truncate the LSB 2-bit. |
| | | 0 | Disable the function |
| | | 1 | Enable the function |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.54 RGB_2_xvYCC Control Register (RGB2XVYCC_CT)

The RGB_2_xvYCC control register is shown in Figure 13-76 and described in Table 13-85.

**Figure 13-76. RGB_2_xvYCC Control Register (RGB2XVYCC_CT)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | XV_CO_OV | XV_FUS | XV_EN |
| R-0h | | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-85. RGB_2_xvYCC control Register (RGB2XVYCC_CT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2 | XV_CO_OV | | Override internal CSC coefficients with register 51 to XX values |
| | | 0 | Disable the function |
| | | 1 | Enable the function |
| 1 | XV_FUS | | xvYCC fullscale mode |
| | | 0 | Disable the function |
| | | 1 | Enable the function |
| 0 | XV_EN | | xvYCC enable |
| | | 0 | Disable the function |
| | | 1 | Enable the function |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.55 RGB_2_xvYCC Conversion R_2_Y Register (R2Y_COEFF_LOW)

The RGB_2_xvYCC conversion R_2_Y register is shown in Figure 13-77 and described in Table 13-86.

**Figure 13-77. RGB_2_xvYCC Conversion R_2_Y Register (R2Y_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | R2YCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-86. RGB_2_xvYCC Conversion R_2_Y Register (R2Y_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | R2YCOEFF_L | RGB to xvYCC conversion. R to Y coefficient lower byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.56 RGB_2_xvYCC Conversion R_2_Y Register (R2Y_COEFF_UP)

The RGB_2_xvYCC conversion R_2_Y register is shown in Figure 13-78 and described in Table 13-87.

**Figure 13-78. RGB_2_xvYCC Conversion R_2_Y Register (R2Y_COEFF_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | R2YCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-87. RGB_2_xvYCC Conversion R_2_Y Register (R2Y_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | R2YCOEFF_H | RGB to xvYCC conversion. R to Y coefficient upper byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.57 RGB_2_xvYCC Conversion G_2_Y Register (G2Y_COEFF_LOW)

The RGB_2_xvYCC conversion G_2_Y register is shown in Figure 13-79 and described in Table 13-88.

**Figure 13-79. RGB_2_xvYCC Conversion G_2_Y Register (G2Y_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | G2YCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-88. RGB_2_xvYCC Conversion G_2_Y Register (G2Y_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | G2YCOEFF_L | RGB to xvYCC conversion. G to Y coefficient lower byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.58 RGB_2_xvYCC Conversion G_2_Y Register (G2Y_COEFF_UP)

The RGB_2_xvYCC conversion G_2_Y register is shown in Figure 13-80 and described in Table 13-89.

**Figure 13-80. RGB_2_xvYCC Conversion G_2_Y Register (G2Y_COEFF_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | G2YCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-89. RGB_2_xvYCC Conversion G_2_Y Register (G2Y_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | G2YCOEFF_H | RGB to xvYCC conversion. G to Y coefficient upper byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.59 RGB_2_xvYCC Conversion B_2_Y Register (B2Y_COEFF_LOW)

The RGB_2_xvYCC conversion B_2_Y register is shown in Figure 13-81 and described in Table 13-90.

**Figure 13-81. RGB_2_xvYCC Conversion B_2_Y Register (B2Y_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | B2YCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-90. RGB_2_xvYCC Conversion B_2_Y Register (B2Y_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | B2YCOEFF_L | RGB to xvYCC conversion. B to Y coefficient lower byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.60 RGB_2_xvYCC Conversion B_2_Y Register (B2Y_COEFF_UP)

The RGB_2_xvYCC conversion B_2_Y register is shown in Figure 13-82 and described in Table 13-91.

**Figure 13-82. RGB_2_xvYCC Conversion B_2_Y Register (B2Y_COEFF_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | B2YCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-91. RGB_2_xvYCC Conversion B_2_Y Register (B2Y_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | B2YCOEFF_H | RGB to xvYCC conversion. B to Y coefficient upper byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.61 RGB_2_xvYCC Conversion R_2_Cb Register (R2CB_COEFF_LOW)

The RGB_2_xvYCC conversion R_2_Cb register is shown in Figure 13-83 and described in Table 13-92.

**Figure 13-83. RGB_2_xvYCC Conversion R_2_Cb Register (R2CB_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | R2CBCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-92. RGB_2_xvYCC Conversion R_2_Cb Register (R2CB_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | R2CBCOEFF_L | RGB to xvYCC conversion. R to Cb coefficient lower byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.62 RGB_2_xvYCC Conversion R_2_Cb Register (R2CB_COEFF_UP)

The RGB_2_xvYCC conversion R_2_Cb register is shown in Figure 13-84 and described in Table 13-93.

**Figure 13-84. RGB_2_xvYCC Conversion R_2_Cb Register (R2CB_COEFF_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | R2CBCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-93. RGB_2_xvYCC Conversion R_2_Cb Register (R2CB_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | R2CBCOEFF_H | RGB to xvYCC conversion. R to Cb coefficient upper byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.63 RGB_2_xvYCC Conversion G_2_Cb Register (G2CB_COEFF_LOW)

The RGB_2_xvYCC conversion G_2_Cb register is shown in Figure 13-85 and described in Table 13-94.

**Figure 13-85. RGB_2_xvYCC Conversion G_2_Cb Register (G2CB_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | G2CBCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-94. RGB_2_xvYCC Conversion G_2_Cb Register (G2CB_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | G2CBCOEFF_L | RGB to xvYCC conversion. G to Cb coefficient lower byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.64 RGB_2_xvYCC Conversion G_2_Cb Register (G2CB_COEFF_UP)

The RGB_2_xvYCC conversion G_2_Cb register is shown in Figure 13-86 and described in Table 13-95.

**Figure 13-86. RGB_2_xvYCC Conversion G_2_Cb Register (G2CB_COEFF_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | G2CBCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-95. RGB_2_xvYCC Conversion G_2_Cb Register (G2CB_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | G2CBCOEFF_H | RGB to xvYCC conversion. G to Cb coefficient upper byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.65 RGB_2_xvYCC Conversion B_2_Cb Register (B2CB_COEFF_LOW)

The RGB_2_xvYCC conversion G_2_Cb register is shown in Figure 13-87 and described in Table 13-96.

**Figure 13-87. RGB_2_xvYCC Conversion B_2_Cb Register (B2CB_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | B2CBCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-96. RGB_2_xvYCC Conversion B_2_Cb Register (B2CB_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | B2CBCOEFF_L | RGB to xvYCC conversion. B to Cb coefficient lower byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.66 RGB_2_xvYCC Conversion B_2_Cb Register (B2CB_COEFF_UP)

The RGB_2_xvYCC conversion G_2_Cb register is shown in Figure 13-88 and described in Table 13-97.

**Figure 13-88. RGB_2_xvYCC Conversion B_2_Cb Register (B2CB_COEFF_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | B2CBCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-97. RGB_2_xvYCC Conversion B_2_Cb Register (B2CB_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | B2CBCOEFF_H | RGB to xvYCC conversion. B to Cb coefficient upper byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.67 RGB_2_xvYCC Conversion R_2_Cr Register (R2CR_COEFF_LOW)

The RGB_2_xvYCC conversion R_2_Cr register is shown in Figure 13-89 and described in Table 13-98.

**Figure 13-89. RGB_2_xvYCC Conversion R_2_Cr Register (R2CR_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | R2CRCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-98. RGB_2_xvYCC Conversion R_2_Cr Register (R2CR_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | R2CRCOEFF_L | RGB to xvYCC conversion. R to Cr coefficient lower byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.68 RGB_2_xvYCC Conversion R_2_Cr Register (R2CR_COEFF_UP)

The RGB_2_xvYCC conversion R_2_Cr register is shown in Figure 13-90 and described in Table 13-99.

**Figure 13-90. RGB_2_xvYCC Conversion R_2_Cr Register (R2CR_COEFF_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | R2CRCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-99. RGB_2_xvYCC Conversion R_2_Cr Register (R2CR_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | R2CRCOEFF_H | RGB to xvYCC conversion. R to Cr coefficient upper byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.69 RGB_2_xvYCC Conversion G_2_Cr Register (G2CR_COEFF_LOW)

The RGB_2_xvYCC conversion G_2_Cr register is shown in Figure 13-91 and described in Table 13-100.

**Figure 13-91. RGB_2_xvYCC Conversion G_2_Cr Register (G2CR_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | G2CRCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-100. RGB_2_xvYCC Conversion G_2_Cr Register (G2CR_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | G2CRCOEFF_L | RGB to xvYCC conversion. G to Cr coefficient lower byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.70 RGB_2_xvYCC Conversion G_2_Cr Register (G2CR_COEFF_UP)

The RGB_2_xvYCC conversion G_2_Cr register is shown in Figure 13-92 and described in Table 13-101.

**Figure 13-92. RGB_2_xvYCC Conversion G_2_Cr Register (G2CR_COEFF_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | G2CRCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-101. RGB_2_xvYCC Conversion G_2_Cr Register (G2CR_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | G2CRCOEFF_H | RGB to xvYCC conversion. G to Cr coefficient upper byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.71 RGB_2_xvYCC Conversion B_2_Cr Register (B2CR_COEFF_LOW)

The RGB_2_xvYCC conversion B_2_Cr register is shown in Figure 13-93 and described in Table 13-102.

**Figure 13-93. RGB_2_xvYCC Conversion B_2_Cr Register (B2CR_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | B2CRCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-102. RGB_2_xvYCC Conversion B_2_Cr Register (B2CR_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | B2CRCOEFF_L | RGB to xvYCC conversion. B to Cr coefficient lower byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.72 RGB_2_xvYCC Conversion B_2_Cr Register (B2CR_COEFF_UP)

The RGB_2_xvYCC conversion B_2_Cr register is shown in Figure 13-94 and described in Table 13-103.

**Figure 13-94. RGB_2_xvYCC Conversion B_2_Cr Register (B2CR_COEFF_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | B2CRCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-103. RGB_2_xvYCC Conversion B_2_Cr Register (B2CR_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | B2CRCOEFF_H | RGB to xvYCC conversion. B to Cr coefficient upper byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.73 RGB_2_xvYCC RGB Input Offset Register (RGB_OFFSET_LOW)

The RGB_2_xvYCC RGB input offset register is shown in Figure 13-95 and described in Table 13-104.

**Figure 13-95. RGB_2_xvYCC RGB Input Offset Register (RGB_OFFSET_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RGB_OFFS_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-104. RGB_2_xvYCC RGB Input Offset Register (RGB_OFFSET_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | RGB_OFFS_L | Input RGB offset value lower byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.74 RGB_2_xvYCC RGB Input Offset Register (RGB_OFFSET_UP)

The RGB_2_xvYCC RGB input offset register is shown in Figure 13-96 and described in Table 13-105.

**Figure 13-96. RGB_2_xvYCC RGB Input Offset Register (RGB_OFFSET_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RGB_OFFS_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-105. RGB_2_xvYCC RGB Input Offset Register (RGB_OFFSET_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | RGB_OFFS_H | Input RGB offset value upper byte (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.75 RGB_2_xvYCC Conversion Y Output Offset Register (Y_OFFSET_LOW)

The RGB_2_xvYCC conversion Y output offset register is shown in Figure 13-97 and described in Table 13-106.

**Figure 13-97. RGB_2_xvYCC Conversion Y Output Offset Register (Y_OFFSET_LOW)**

| 31 | 7 | 6 | 0 |
|---|---|---|---|
| Reserved | | Y_OFFS_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-106. RGB_2_xvYCC Conversion Y Output Offset Register (Y_OFFSET_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-7 | Reserved | Reserved |
| 6-0 | Y_OFFS_L | Output Y offset value lower 7 bits (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.76 RGB_2_xvYCC Conversion Y Output Offset Register (Y_OFFSET_UP)

The RGB_2_xvYCC conversion Y output offset register is shown in Figure 13-98 and described in Table 13-107.

**Figure 13-98. RGB_2_xvYCC Conversion Y Output Offset Register (Y_OFFSET_UP)**

| 31 | 7 | 6 | 0 |
|---|---|---|---|
| Reserved | | Y_OFFS_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-107. RGB_2_xvYCC Conversion Y Output Offset Register (Y_OFFSET_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-7 | Reserved | Reserved |
| 6-0 | Y_OFFS_H | Output Y offset value upper 7 bits (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.77 RGB_2_xvYCC Conversion CbCr Output Offset Register (CBCR_OFFSET_LOW)

The RGB_2_xvYCC conversion CbCr output offset register is shown in Figure 13-99 and described in Table 13-108.

**Figure 13-99. RGB_2_xvYCC Conversion CbCr Output Offset Register (CBCR_OFFSET_LOW)**

| 31 | 7 | 6 | 0 |
|---|---|---|---|
| Reserved | | CBCR_OFFS_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-108. RGB_2_xvYCC Conversion CbCr Output Offset Register (CBCR_OFFSET_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-7 | Reserved | Reserved |
| 6-0 | CBCR_OFFS_L | Output CbCr offset value lower 7 bits (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.78 RGB_2_xvYCC Conversion CbCr Output Offset Register (CBCR_OFFSET_UP)

The RGB_2_xvYCC conversion CbCr output offset register is shown in Figure 13-100 and described in Table 13-109.

**Figure 13-100. RGB_2_xvYCC Conversion CbCr Output Offset Register (CBCR_OFFSET_UP)**

| 31 | 7 | 6 | 0 |
|---|---|---|---|
| Reserved | | CBCR_OFFS_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-109. RGB_2_xvYCC Conversion CbCr Output Offset Register (CBCR_OFFSET_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-7 | Reserved | Reserved |
| 6-0 | CBCR_OFFS_H | Output CbCr offset value upper 7 bits (override internal CSC value when XV_CO_OV = 1 in register RGB_2_XVYCC_CT) |

### 13.3.2.79 Interrupt State Register (INTR_STATE)

The interrupt state register is shown in Figure 13-101 and described in Table 13-110.

**Figure 13-101. Interrupt State Register (INTR_STATE)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | INTR |
| R-0h | | R-0h |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-110. Interrupt State Register (INTR_STATE) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reserved |
| 0 | INTR | Interrupt State. When an interrupt is asserted; this bit is set to 1. The polarity of the INT output signal is set using this bit and the POLARITY bit in the INT_CTRL register. Only INTR1, INTR2, INTR3, and INTR4 bits with matching set bits in INT_UNMASK can contribute to setting the INTR bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.80 Interrupt Source Register (INTR1)

The interrupt source register is shown in Figure 13-102 and described in Table 13-111.

**Figure 13-102. Interrupt Source Register (INTR1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SOFT | HPD | RSEN | DROP_SAMPLE | BIP_HASE_ERR | RI_128 | OVER_RUN | UNDER_RUN |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-111. Interrupt Source Register (INTR1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7 | SOFT | Software Induced Interrupt. Allows the firmware to generate an interrupt directly. |
| 6 | HPD | Monitor detect interrupt. Asserted if hot plug detect has changed state. The HDMI transmitter signals a change in the connectivity to a Sink, either unplug or plug. HDMI specifies that hot plug be active only when the Sink s EDID is ready to be read and that hot plug be toggled any time there is a change in connectivity downstream of an attached repeater. |
| 5 | RSEN | Receiver sense interrupt asserted if RSEN has changed. This interrupt is set whenever $V_{CC}$ is applied to or removed from the attached HDMI receiver chip. A receiver with multiple input ports can also disconnect the TMDS termination to the unused port, which triggers this RSEN interrupt. |
| 4 | DROP_SAMPLE | New preamble forced to drop sample (S/PDIF input only). If the HDMI transmitter detects an 8-bit preamble in the S/PDIF input stream before the subframe has been captured, this interrupt is set. A S/PDIF input that stops signaling or a flat line condition can create such a premature preamble. |
| 3 | BIP_HASE_ERR | Input S/PDIF stream has bi-phase error. This can occur when there is noise or an Fs rate change on the S/PDIF input. |
| 2 | RI_128 | Input counted past frame count threshold set in RI_128_COMP register. This interrupt occurs when the count written to register RI_128_COMP is matched by the VSYNC (frame) counter in the HDMI transmitter. It should trigger the firmware to perform a link integrity check. Such a match occurs every 128 frames. |
| 1 | OVER_RUN | Audio FIFO overflow. This interrupt occurs if the audio FIFO overflows when more samples are written into it than are drawn out across the HDMI link. Such a condition can occur from a transient change in the Fs or pixel clock rate. |
| 0 | UNDER_RUN | Audio FIFO underflow. Similar to OVER_RUN, this interrupt occurs when the audio FIFO empties. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.81 Interrupt Source Register (INTR2)

The interrupt source register is shown in Figure 13-103 and described in Table 13-112.

**Figure 13-103. Interrupt Source Register (INTR2)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BCAP_DONE | SPDIF_PAR | ENC_DIS | PREAM_ERR | CTS_CHG | ACR_OVR | TCLK_STBL | VSYNC_REC |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-x | R-0h |

LEGEND: R = Read only; -*n* = value after reset; x = value is indeterminate

**Table 13-112. Interrupt Source Register (INTR2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7 | BCAP_DONE | If set, this interrupt detected that the FIFORDY bit is set to 1. |
| 6 | SPDIF_PAR | S/PDIF parity error. The S/PDIF stream includes a parity (P) bit at the end of each sub-frame. An interrupt occurs if the calculated parity does not match the state of this bit. |
| 5 | ENC_DIS | The ENC_EN bit (in register HDCP_CTRL) changed from 1 to 0. This interrupt occurs if encryption is turned off. |
| 4 | PREAM_ERR | This condition is the opposite of the condition that causes DROP_SAMPLE (in register INTR1). This interrupt occurs if a preamble is expected but not found when decoding the S/PDIF stream. |
| 3 | CTS_CHG | Change in ACR CTS value. This interrupt occurs when the change is of an unexpected magnitude. Such an interrupt should be expected when changing Fs or pixel clock frequency. |
| 2 | ACR_OVR | ACR packet overwrite. This interrupt occurs if the HDMI transmitter puts a NCTS packet into the queue before the previous NCTS packet has been sent. This can happen if very long active data times do not allow for sufficient NCTS packet bandwidth. For all CEA- 861D modes, no ACR_OVR interrupt should occur. |
| 1 | TCLK_STBL | Whenever IDCK changes, there is a temporary instability in the internal clocking. This interrupt is set when the internal clocking has stabilized. |
| 0 | VSYNC_REC | Asserted when VSYNC active edge is recognized. It is useful for triggering firmware actions that occur during vertical blanking. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.82 Interrupt Source Register (INTR3)

The interrupt source register is shown in Figure 13-104 and described in Table 13-113.

**Figure 13-104. Interrupt Source Register (INTR3)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RI_ERR_3 | RI_ERR_2 | RI_ERR_1 | RI_ERR_0 | DDC_CMD_DONE | DDC_FIFO_HALF | DDC_FIFO_FULL | DDC_FIFO_EMPTY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-113. Interrupt Source Register (INTR3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7 | RI_ERR_3 | Ri and Ri do not match during frame 127 (ICNT .1). |
| 6 | RI_ERR_2 | Ri and Ri do not match during frame 0 (ICNT). |
| 5 | RI_ERR_1 | Ri did not change between frame 127 and 0. |
| 4 | RI_ERR_0 | Ri not read within one frame. |
| 3 | DDC_CMD_DONE | DDC command is complete. |
| 2 | DDC_FIFO_HALF | DDC FIFO is half full. |
| 1 | DDC_FIFO_FULL | DDC FIFO is full. |
| 0 | DDC_FIFO_EMPTY | DDC FIFO is empty. Reset value is 0, but can be set after reset since the FIFO is empty. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.83 Interrupt Source Register (INTR4)

The interrupt source register is shown in Figure 13-105 and described in Table 13-114.

**Figure 13-105. Interrupt Source Register (INTR4)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | REG_INTR4_STAT3 | REG_INTR4_STAT2 | REG_INTR4_STAT1 | DSD_INVALID |
| R-0h | | | | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear, write 0 has no effect; -*n* = value after reset

**Table 13-114. Interrupt Source Register (INTR4) Field Descriptions**

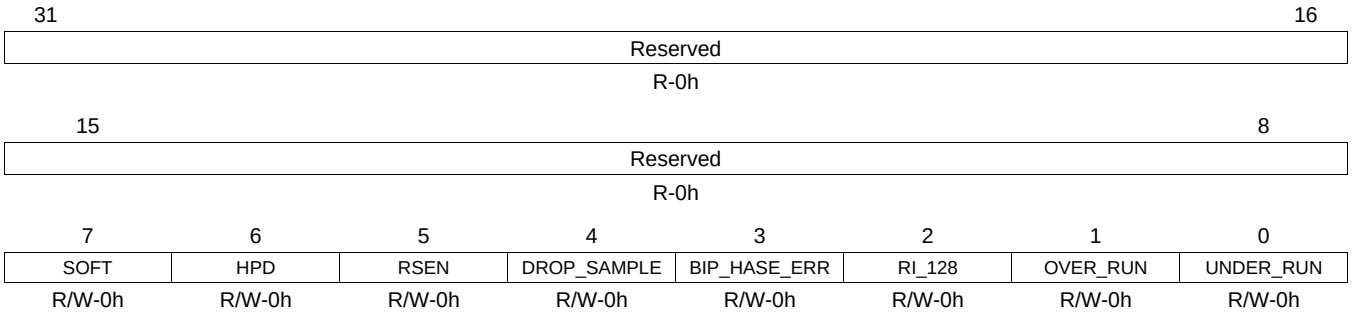
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | REG_INTR4_STAT3 | | CEC interrupt. |
| | | 0 | No interrupt occurred |
| | | 1 | Interrupt asserted |
| 2 | REG_INTR4_STAT2 | | For each interrupt bit: |
| | | 0 | No interrupt occurred |
| | | 1 | Interrupt asserted |
| 1 | REG_INTR4_STAT1 | | For each interrupt bit: |
| | | 0 | No interrupt occurred |
| | | 1 | Interrupt asserted |
| 0 | DSD_INVALID | 0 | DSD stream got invalid sequence: more then 24 bits of the same value. Asserted if set to 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.84 Interrupt Unmask Register (INT_UNMASK1)

The interrupt unmask register is shown in Figure 13-106 and described in Table 13-115.

**Figure 13-106. Interrupt Unmask Register (INT_UNMASK1)**




LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-115. Interrupt Unmask Register (INT_UNMASK1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7 | SOFT | Software Induced Interrupt. Allows the firmware to generate an interrupt directly. |
| 6 | HPD | Monitor detect interrupt. Asserted if hot plug detect has changed state. The HDMI transmitter signals a change in the connectivity to a Sink, either unplug or plug. HDMI specifies that hot plug be active only when the Sink s EDID is ready to be read and that hot plug be toggled any time there is a change in connectivity downstream of an attached repeater. |
| 5 | RSEN | Receiver sense interrupt asserted if RSEN has changed. This interrupt is set whenever $V_{CC}$ is applied to or removed from the attached HDMI receiver chip. A receiver with multiple input ports can also disconnect the TMDS termination to the unused port, which triggers this RSEN interrupt. |
| 4 | DROP_SAMPLE | New preamble forced to drop sample (S/PDIF input only). If the HDMI transmitter detects an 8-bit preamble in the S/PDIF input stream before the subframe has been captured, this interrupt is set. A S/PDIF input that stops signaling or a flat line condition can create such a premature preamble. |
| 3 | BIP_HASE_ERR | Input S/PDIF stream has bi-phase error. This can occur when there is noise or an Fs rate change on the S/PDIF input. |
| 2 | RI_128 | Input counted past frame count threshold set in RI_128_COMP register. This interrupt occurs when the count written to register RI_128_COMP is matched by the VSYNC (frame) counter in the HDMI transmitter. It should trigger the firmware to perform a link integrity check; such a match occurs every 128 frames. |
| 1 | OVER_RUN | Audio FIFO overflow. This interrupt occurs if the audio FIFO overflows when more samples are written into it than are drawn out across the HDMI link. Such a condition can occur from a transient change in the Fs or pixel clock rate. |
| 0 | UNDER_RUN | Audio FIFO underflow. Similar to OVER_RUN. This interrupt occurs when the audio FIFO empties. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.85 Interrupt Unmask Register (INT_UNMASK2)

The interrupt unmask register is shown in Figure 13-107 and described in Table 13-116.

**Figure 13-107. Interrupt Unmask Register (INT_UNMASK2)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BCAP_DONE | SPDIF_PAR | ENC_DIS | PREAM_ERR | CTS_CHG | ACR_OVR | TCLK_STBL | VSYNC_REC |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-116. Interrupt Unmask Register (INT_UNMASK2) Field Descriptions**

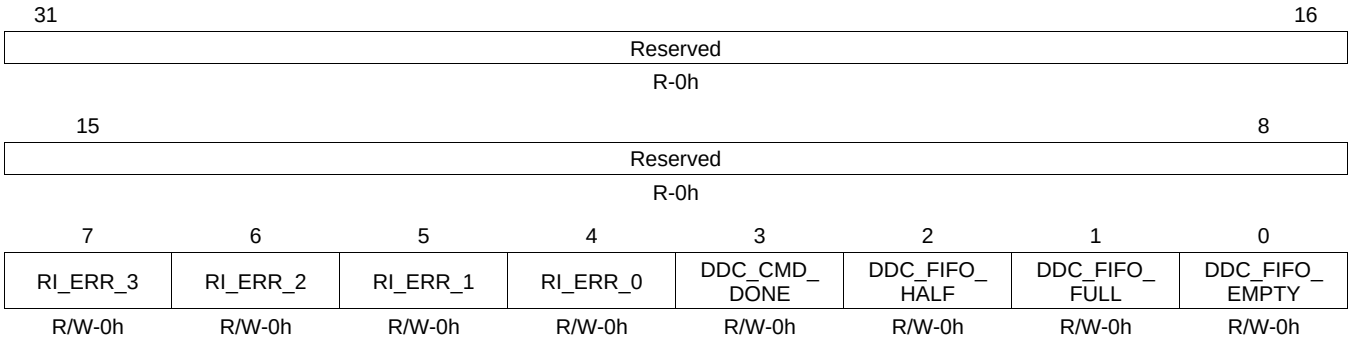
| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7 | BCAP_DONE | If set, this interrupt detected that the FIFORDY bit is set to 1. |
| 6 | SPDIF_PAR | S/PDIF parity error. The S/PDIF stream includes a parity (P) bit at the end of each sub-frame. An interrupt occurs if the calculated parity does not match the state of this bit. |
| 5 | ENC_DIS | The ENC_EN bit (in register HDCP_CTRL) changed from 1 to 0. This interrupt occurs if encryption is turned off. |
| 4 | PREAM_ERR | This condition is the opposite of the condition that causes DROP_SAMPLE (in register INTR1). This interrupt occurs if a preamble is expected but not found when decoding the S/PDIF stream. |
| 3 | CTS_CHG | Change in ACR CTS value. This interrupt occurs when the change is of an unexpected magnitude. Such an interrupt should be expected when changing Fs or pixel clock frequency. |
| 2 | ACR_OVR | ACR packet overwrite. This interrupt occurs if the HDMI transmitter puts a NCTS packet into the queue before the previous NCTS packet has been sent. This can happen if very long active data times do not allow for sufficient NCTS packet bandwidth. For all CEA- 861D modes, no ACR_OVR interrupt should occur. |
| 1 | TCLK_STBL | TCLK_STABLE. Whenever IDCK changes, there is a temporary instability in the internal clocking. This interrupt is set when the internal clocking has stabilized. |
| 0 | VSYNC_REC | Asserted when VSYNC active edge is recognized, it is useful for triggering firmware actions that occur during vertical blanking. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.86 Interrupt Unmask Register (INT_UNMASK3)

The interrupt unmask register is shown in Figure 13-108 and described in Table 13-117.

**Figure 13-108. Interrupt Unmask Register (INT_UNMASK3)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-117. Interrupt Unmask Register (INT_UNMASK3) Field Descriptions**

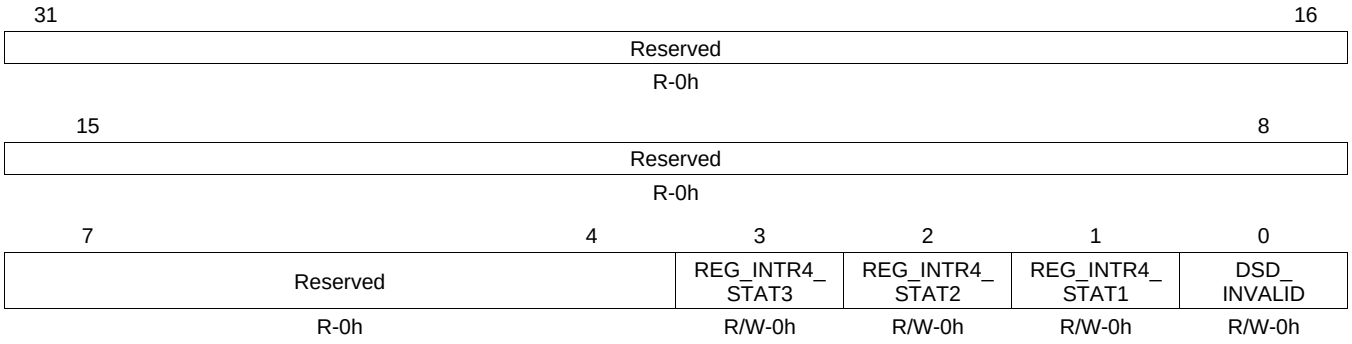
| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7 | RI_ERR_3 | Ri and Ri do not match during frame 127 (ICNT .1). |
| 6 | RI_ERR_2 | Ri and Ri do not match during frame 0 (ICNT). |
| 5 | RI_ERR_1 | Ri did not change between frame 127 and 0. |
| 4 | RI_ERR_0 | Ri not read within one frame. |
| 3 | DDC_CMD_DONE | DDC command is complete. |
| 2 | DDC_FIFO_HALF | DDC FIFO is half full. |
| 1 | DDC_FIFO_FULL | DDC FIFO is full. |
| 0 | DDC_FIFO_EMPTY | DDC FIFO is empty. Reset value is 0, but can be set after reset since the FIFO Is empty. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.87 Interrupt Unmask Register (INT_UNMASK4)

The interrupt unmask register is shown in Figure 13-109 and described in Table 13-118.

**Figure 13-109. Interrupt Unmask Register (INT_UNMASK4)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-118. Interrupt Unmask Register (INT_UNMASK4) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | REG_INTR4_STAT3 | | CEC interrupt |
| | | 0 | No interrupt occurred |
| | | 1 | Interrupt asserted |
| 2 | REG_INTR4_STAT2 | | For each interrupt bit: |
| | | 0 | No interrupt occurred |
| | | 1 | Interrupt asserted |
| 1 | REG_INTR4_STAT1 | | For each interrupt bit: |
| | | 0 | No interrupt occurred |
| | | 1 | Interrupt asserted |
| 0 | DSD_INVALID | 0 | DSD stream got invalid sequence: more then 24 bits of the same value. Asserted if set to 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.88 Interrupt Control Register (INT_CTRL)

The interrupt control register is shown in Figure 13-110 and described in Table 13-119.

**Figure 13-110. Interrupt Control Register (INT_CTRL)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | SOFT_INTR | OPEN_DRAIN | POLARITY | Reserved |
| R-0h | | | | R/W-0h | R/W-0h | R/W-1h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-119. Interrupt Control Register (INT_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | SOFT_INTR | | Set software interrupt |
| | | 0 | Clear interrupt |
| | | 1 | Set interrupt |
| 2 | OPEN_DRAIN | | INT pin output type |
| | | 0 | Push/Pull |
| | | 1 | Open Drain pin |
| 1 | POLARITY | | INT pin assertion level |
| | | 0 | Assert HIGH |
| | | 1 | Assert LOW |
| 0 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.89 xvYCC_2_RGB Control Register (XVYCC2RGB_CTL)

The xvYCC_2_RGB control register is shown in Figure 13-111 and described in Table 13-120.

**Figure 13-111. xvYCC_2_RGB Control Register (XVYCC2RGB_CTL)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | EXP_ONLY | BYP_ALL | SW_OVR | FULLRANGE | XVYCCSEL |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-120. xvYCC_2_RGB Control Register (XVYCC2RGB_CTL) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reserved |
| 4 | EXP_ONLY | When set to 1, indicates the internal CSC will be bypassed and only range expansion is on. |
| 3 | BYP_ALL | When set to 1, indicates all the functions will be bypassed. |
| 2 | SW_OVR | Software over ride. When turned on, all coefficients and offsets are coming from registers not internal hardware decision. |
| 1 | FULLRANGE | xvYCC full-range expansion enable |
| 0 | XVYCCSEL | Indicates the source. When set to 1, is xvYCC; when cleared to 0, is YcbCr. It can be configured by firmware. Under auto video configure (AVC) mode, this control comes from HDMI packet decoding. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.90 xvYCC_2_RGB Conversion Y_2_R Register (Y2R_COEFF_LOW)

The xvYCC_2_RGB conversion Y_2_R register is shown in Figure 13-112 and described in Table 13-121.

**Figure 13-112. xvYCC_2_RGB Conversion Y_2_R Register (Y2R_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | Y2RCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-121. xvYCC_2_RGB Conversion Y_2_R Register (Y2R_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | Y2RCOEFF_L | xvYCC to RGB conversion. Y to R coefficient lower byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |

### 13.3.2.91 xvYCC_2_RGB Conversion Y_2_R Register (Y2R_COEFF_UP)

The xvYCC_2_RGB conversion Y_2_R register is shown in Figure 13-113 and described in Table 13-122.

**Figure 13-113. xvYCC_2_RGB Conversion Y_2_R Register (Y2R_COEFF_UP)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | Y2RCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-122. xvYCC_2_RGB Conversion Y_2_R Register (Y2R_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reserved |
| 4-0 | Y2RCOEFF_H | xvYCC to RGB conversion. Y to R coefficient upper byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.92 xvYCC_2_RGB Conversion Cr_2_R Register (CR2R_COEFF_LOW)

The xvYCC_2_RGB conversion Cr_2_R register is shown in Figure 13-114 and described in Table 13-123.

**Figure 13-114. xvYCC_2_RGB Conversion Cr_2_R Register (CR2R_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CR2RCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-123. xvYCC_2_RGB Conversion Cr_2_R Register (CR2R_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CR2RCOEFF_L | xvYCC to RGB conversion. Cr to R coefficient lower byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |

### 13.3.2.93 xvYCC_2_RGB Conversion Cr_2_R Register (C2R2R_COEFF_UP)

The xvYCC_2_RGB conversion Cr_2_R register is shown in Figure 13-115 and described in Table 13-124.

**Figure 13-115. xvYCC_2_RGB Conversion Cr_2_R Register (C2R2R_COEFF_UP)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | CR2RCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-124. xvYCC_2_RGB Conversion Cr_2_R Register (C2R2R_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reserved |
| 4-0 | CR2RCOEFF_H | xvYCC to RGB conversion. Cr to R coefficient upper byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.94 xvYCC_2_RGB Conversion Cb_2_B Register (CB2B_COEFF_LOW)

The xvYCC_2_RGB conversion Cb_2_B register is shown in Figure 13-116 and described in Table 13-125.

**Figure 13-116. xvYCC_2_RGB Conversion Cb_2_B Register (CB2B_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CB2BCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-125. xvYCC_2_RGB Conversion Cb_2_B Register (CB2B_COEFF_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CB2BCOEFF_L | xvYCC to RGB conversion. Cb to B coefficient lower byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |

### 13.3.2.95 xvYCC_2_RGB Conversion Cb_2_B Register (CB2B_COEFF_UP)

The xvYCC_2_RGB Conversion Cb_2_B Register is shown in Figure 13-117 and described in Table 13-126.

**Figure 13-117. xvYCC_2_RGB Conversion Cb_2_B Register (CB2B_COEFF_UP)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | CB2BCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-126. xvYCC_2_RGB Conversion Cb_2_B Register (CB2B_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reserved |
| 4-0 | CB2BCOEFF_H | xvYCC to RGB conversion. Cb to B coefficient upper byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.96 xvYCC_2_RGB Conversion Cr_2_G Register (CR2G_COEFF_LOW)

The xvYCC_2_RGB conversion Cr_2_G register is shown in Figure 13-118 and described in Table 13-127.

**Figure 13-118. xvYCC_2_RGB Conversion Cr_2_G Register (CR2G_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CR2GCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-127. CR2G_COEFF_LOW Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CR2GCOEFF_L | xvYCC to RGB conversion. Cr to G coefficient lower byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |

### 13.3.2.97 xvYCC_2_RGB Conversion Cr_2_G Register (CR2G_COEFF_UP)

The xvYCC_2_RGB conversion Cr_2_G register is shown in Figure 13-119 and described in Table 13-128.

**Figure 13-119. xvYCC_2_RGB Conversion Cr_2_G Register (CR2G_COEFF_UP)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | CR2GCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-128. xvYCC_2_RGB Conversion Cr_2_G Register (CR2G_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reserved |
| 4-0 | CR2GCOEFF_H | xvYCC to RGB conversion. Cr to G coefficient upper byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.98 xvYCC_2_RGB Conversion Cb_2_G Register (CB2G_COEFF_LOW)

The xvYCC_2_RGB conversion Cb_2_G register is shown in Figure 13-120 and described in Table 13-129.

**Figure 13-120. xvYCC_2_RGB Conversion Cb_2_G Register (CB2G_COEFF_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CB2GCOEFF_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-129. xvYCC_2_RGB Conversion Cb_2_G Register (CB2G_COEFF_LOW)**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CB2GCOEFF_L | xvYCC to RGB conversion. Cb to G coefficient lower byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |

### 13.3.2.99 xvYCC_2_RGB Conversion Cb_2_G Register (CB2G_COEFF_UP)

The xvYCC_2_RGB conversion Cb_2_G register is shown in Figure 13-121 and described in Table 13-130.

**Figure 13-121. xvYCC_2_RGB Conversion Cb_2_G Register (CB2G_COEFF_UP)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | CB2GCOEFF_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-130. xvYCC_2_RGB Conversion Cb_2_G Register (CB2G_COEFF_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reserved |
| 4-0 | CB2GCOEFF_H | xvYCC to RGB conversion. Cb to G coefficient upper byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.100 xvYCC_2_RGB Conversion Y Offset Register (YOFFSET1_LOW)

The xvYCC_2_RGB conversion Y offset register is shown in Figure 13-122 and described in Table 13-131.

**Figure 13-122. xvYCC_2_RGB Conversion Y Offset Register (YOFFSET1_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | YOFFS1_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-131. xvYCC_2_RGB Conversion Y Offset Register (YOFFSET1_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | YOFFS1_L | xvYCC2RGB Y offset coefficient lower byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |

### 13.3.2.101 xvYCC_2_RGB Conversion Y Offset Register (YOFFSET1_UP)

The xvYCC_2_RGB conversion Y offset register is shown in Figure 13-123 and described in Table 13-132.

**Figure 13-123. xvYCC_2_RGB Conversion Y Offset Register (YOFFSET1_UP)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | YOFFS1_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-132. xvYCC_2_RGB Conversion Y Offset Register (YOFFSET1_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reserved |
| 3-0 | YOFFS1_H | xvYCC2RGB Y offset coefficient upper byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.102 xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_LOW)

The xvYCC_2_RGB conversion offset1 register is shown in Figure 13-124 and described in Table 13-133.

**Figure 13-124. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | OFFS1_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-133. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | OFFS1_L | xvYCC2RGB offset1 coefficient lower byte. Offset for RGB channel before right shifting, which is subtractive if software override is on (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |

### 13.3.2.103 xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_MID)

The xvYCC_2_RGB conversion offset1 register is shown in Figure 13-125 and described in Table 13-134.

**Figure 13-125. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_MID)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | OFFS1_M | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-134. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_MID) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | OFFS1_M | xvYCC2RGB offset1 coefficient mid byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |

### 13.3.2.104 xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_UP)

The xvYCC_2_RGB conversion offset1 register is shown in Figure 13-126 and described in Table 13-135.

**Figure 13-126. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_UP)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | OFFS1_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-135. xvYCC_2_RGB Conversion Offset1 Register (OFFSET1_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | OFFS1_H | xvYCC2RGB offset1 coefficient upper byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.105 xvYCC_2_RGB Conversion Offset2 Register (OFFSET2_LOW)

The xvYCC_2_RGB conversion offset2 register is shown in Figure 13-127 and described in Table 13-136.

**Figure 13-127. xvYCC_2_RGB Conversion Offset2 Register (OFFSET2_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | OFFS2_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-136. xvYCC_2_RGB Conversion Offset2 Register (OFFSET2_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | OFFS2_L | xvYCC2RGB offset2 coefficient lower byte. Offset for RGB channel before right shifting, which is subtractive if software override is on (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |

### 13.3.2.106 xvYCC_2_RGB Conversion Offset2 Register (OFFSET2_UP)

The xvYCC_2_RGB conversion offset2 register is shown in Figure 13-128 and described in Table 13-137.

**Figure 13-128. xvYCC_2_RGB Conversion Offset2 Register (OFFSET2_UP)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | OFFS2_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-137. xvYCC_2_RGB Conversion Offset2 Register (OFFSET2_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reserved |
| 3-0 | OFFS2_H | xvYCC2RGB offset2 coefficient upper byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.107 xvYCC_2_RGB Conversion DC Level Register (DCLEVEL_LOW)

The xvYCC_2_RGB conversion DC level register is shown in Figure 13-129 and described in Table 13-138.

**Figure 13-129. xvYCC_2_RGB Conversion DC Level Register (DCLEVEL_LOW)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DC_LEV_L | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-138. xvYCC_2_RGB Conversion DC Level Register (DCLEVEL_LOW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DC_LEV_L | xvYCC2RGB DC level coefficient lower byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |

### 13.3.2.108 xvYCC_2_RGB Conversion DC Level Register (DCLEVEL_UP)

The xvYCC_2_RGB conversion DC level register is shown in Figure 13-130 and described in Table 13-139.

**Figure 13-130. xvYCC_2_RGB Conversion DC Level Register (DCLEVEL_UP)**

| 31 | 6 | 5 | 0 |
|---|---|---|---|
| Reserved | | DC_LEV_H | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-139. xvYCC_2_RGB Conversion DC Level Register (DCLEVEL_UP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-6 | Reserved | Reserved |
| 5-0 | DC_LEV_H | xvYCC2RGB DC level coefficient upper byte (override internal CSC value when SW_OVR = 1 in register XVYCC2RGB_CTL) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.109 DDC I2C Manual Register (DDC_MAN)

The DDC I2C manual register is shown in Figure 13-131 and described in Table 13-140.

**Figure 13-131. DDC I2C Manual Register (DDC_MAN)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MAN_OVR | Reserved | MAN_SDA | MAN_SCL | Reserved | | IO_SCL | IO_SDA |
| R/W-0 | R-0h | R/W-0h | R/W-0h | R-0h | | R-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-140. DDC I2C Manual Register (DDC_MAN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | MAN_OVR | | Manual override of SCL and SDA output |
| | | 0 | Normal operation |
| | | 1 | Override port with MAN_SCL and MAN_SDA states |
| 6 | Reserved | 0 | Reserved |
| 5 | MAN_SDA | 0 | Manual SDA output |
| 4 | MAN_SCL | 0 | Manual SCL output |
| 3-2 | Reserved | 0 | Reserved |
| 1 | IO_SCL | 0 | DDC SCL input state |
| 0 | IO_SDA | 0 | DDC SDA input state |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.110 DDC I2C Target Slave Address Register (DDC_ADDR)

The DDC I2C target slave address register (DDC_ADDR) is shown in Figure 13-132 and described in Table 13-141.

**Figure 13-132. DDC I2C Target Slave Address Register (DDC_ADDR)**

| 31 | 8 | 7 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | DDC_ADDR | | Reserved |
| R-0h | | R/W-0h | | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-141. DDC I2C Target Slave Address Register (DDC_ADDR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-1 | DDC_ADDR | DDC device address |
| 0 | Reserved | Reserved |

### 13.3.2.111 DDC I2C Target Segment Address Register (DDC_SEGM)

The DDC I2C target segment address register is shown in Figure 13-133 and described in Table 13-142.

**Figure 13-133. DDC I2C Target Segment Address Register (DDC_SEGM)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DDC_SEGM | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-142. DDC I2C Target Segment Address Register (DDC_SEGM) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DDC_SEGM | DDC segment address |

### 13.3.2.112 DDC I2C Target Offset Address Register (DDC_OFFSET)

The DDC I2C target offset address register is shown in Figure 13-134 and described in Table 13-143.

**Figure 13-134. DDC I2C Target Offset Address Register (DDC_OFFSET)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DDC_OFFSET | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-143. DDC I2C Target Offset Address Register (DDC_OFFSET) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DDC_OFFSET | DDC offset address |

### 13.3.2.113 DDC I2C Data Count Register (DDC_COUNT1)

The DDC I2C data count register is shown in Figure 13-135 and described in Table 13-144.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 13-135. DDC I2C Data Count Register (DDC_COUNT1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DDC_COUNT1 | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-144. DDC I2C Data Count Register (DDC_COUNT1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DDC_COUNT | The total number of bytes to be read from the slave or written to the slave before a stop bit is sent on the DDC bus. For example, if the HDCP KSV FIFO length is 635 bytes (127 devices × 5 bytes/KSV), the DDC_COUNT must be 27Bh. |

### 13.3.2.114 DDC I2C Data Count Register (DDC_COUNT2)

The DDC I2C data count register is shown in Figure 13-136 and described in Table 13-145.

**Figure 13-136. DDC I2C Data Count Register (DDC_COUNT2)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DDC_COUNT2 | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-145. DDC I2C Data Count Register (DDC_COUNT2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DDC_COUNT | The total number of bytes to be read from the slave or written to the slave before a Stop bit is sent on the DDC bus. For example, if the HDCP KSV FIFO length is 635 bytes (127 devices × 5 bytes/KSV), the DDC_COUNT must be 27Bh. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.115 DDC I2C Status Register (DDC_STATUS)

The DDC I2C Status Register (DDC_STATUS) is shown in Figure 13-137 and described in Table 13-146.

**Figure 13-137. DDC I2C Status Register (DDC_STATUS)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | BUS_LOW | NO_ACK | IN_PROG | FIFO_FULL | FIFO_EMP | FRD_USE | FWT_USE |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-1h | R-0h | R-0h |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-146. DDC I2C Status Register (DDC_STATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | BUS_LOW | | I2C bus is LOW |
| | | 0 | I2C bus is not pulled low by an external device |
| | | 1 | I2C transaction did not start because I2C bus is pulled LOW by an external device |
| 5 | NO_ACK | | HDMI transmitter did not receive an ACK |
| | | 0 | HDMI transmitter did receive an ACK |
| | | 1 | HDMI transmitter did not receive an ACK from slave device during address or data write |
| 4 | IN_PROG | | DDC operation in progress |
| | | 0 | DDC operation is not in progress |
| | | 1 | DDC operation in progress |
| 3 | FIFO_FULL | | DDC FIFO full |
| | | 0 | DDC FIFO is not full |
| | | 1 | DDC FIFO is full |
| 2 | FIFO_EMP | | DDC FIFO empty |
| | | 0 | DDC FIFO is not empty |
| | | 1 | DDC FIFO is empty |
| 1 | FRD_USE | | DDC FIFO read in use |
| | | 0 | DDC FIFO read is not in use |
| | | 1 | DDC FIFO read is in use |
| 0 | FWT_USE | | DDC FIFO write in use |
| | | 0 | DDC FIFO write is not in use |
| | | 1 | DDC FIFO write is in use |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.116 DDC I2C Command Register (DDC_CMD)

The DDC I2C command register is shown in Figure 13-138 and described in Table 13-147.

**Figure 13-138. DDC I2C Command Register (DDC_CMD)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | DDC_FLT_EN | SDA_DEL_EN | DDC_CMD | | | |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-147. DDC I2C Command Register (DDC_CMD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5 | DDC_FLT_EN | | Enable the DCC delay. A DDC delay is inserted into the SDA line to create a 300 ns delay for the falling edge of the DDC SDA signal to avoid an erroneous I2C START condition. The real start condition must have a setup time of 600 ns so that this delay of 300 ns does not remove the real START condition. Filtering is done using a ring oscillator. |
| | | 0 | Enable |
| | | 1 | Disable |
| 4 | SDA_DEL_EN | | Enable 3 ns glitch filtering on the DDC clock and data line. Filtering is done using a ring oscillator. |
| | | 0 | Enable |
| | | 1 | Disable |
| 3-0 | DDC_CMD | | DDC command<br>Writing to this register immediately initiates the I2C transaction on the DDC bus. Note: The clear FIFO command resets the FIFO read and write pointers to zero. Data formerly loaded into the FIFO cannot be re-read after a clear FIFO, as the FIFO will be empty. Other command codes are reserved and may cause the DDC bus to hang if used. The clock SCL command resets any I2C devices on the DDC lines. This should be initiated once before initiating the DDC commands. |
| | | 0 | Current address read with no ACK on last byte |
| | | 2h | Sequential read with no ACK on last byte |
| | | 4h | Enhanced DDC read with no ACK on last byte |
| | | 6h | Sequential write ignoring ACK on last byte |
| | | 7h | Sequential write requiring ACK on last byte |
| | | 9h | Clear FIFO |
| | | Ah | Clock SCL |
| | | Fh | Abort transaction |
| | | | All other values are reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.117 DDC I2C Data Register (DDC_DATA)

The DDC I2C data register is shown in Figure 13-139 and described in Table 13-148.

**Figure 13-139. DDC I2C Data Register (DDC_DATA)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DDC_DATA | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-148. DDC I2C Data Register (DDC_DATA) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | DDC_DATA | DDC data input |

### 13.3.2.118 DDC I2C FIFO Count Register (DDC_FIFOCNT)

The DDC I2C FIFO count register is shown in Figure 13-140 and described in Table 13-149.

**Figure 13-140. DDC I2C FIFO Count Register (DDC_FIFOCNT)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | DDC_FIFOCNT | |
| R-0h | | R-0h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-149. DDC I2C FIFO Count Register (DDC_FIFOCNT) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reserved |
| 4-0 | DDC_FIFOCNT | FIFO data byte count (the number of bytes in the FIFO). The DDC FIFO size is 16. The maximum value for DDC_FIFOCNT is 10h. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.119 ROM Status Register (EPST)

The ROM status register is shown in Figure 13-141 and described in Table 13-150.

**Figure 13-141. ROM Status Register (EPST)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | BIST2_ERR | BIST1_ERR | Reserved | | | CRC_ERR | CMDD |
| R-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-150. ROM Status Register (EPST) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | BIST2_ERR | | BIST self-authentication test 2 error |
| | | 0 | No error |
| | | 1 | BIST self-authentication test 2 error |
| 5 | BIST1_ERR | | BIST self-authentication test 1 error |
| | | 0 | No error |
| | | 1 | BIST self-authentication test 1 error |
| 4-2 | Reserved | 0 | Reserved |
| 1 | CRC_ERR | | CRC error |
| | | 0 | No error |
| | | 1 | CRC error |
| 0 | CMDD | | Command done |
| | | 0 | After reset. Set to 1 once the KSV keys are read (when osclk is running). |
| | | 1 | Command done (last operation completed successfully). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.2.120 ROM Command Register (EPCM)

The ROM command register is shown in Figure 13-142 and described in Table 13-151.

**Figure 13-142. ROM Command Register (EPCM)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | LD_KSV | EPCM | | | | |
| R-0h | | R/W-0h | R/W-0h | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-151. ROM Command Register (EPCM) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5 | LD_KSV | | Load KSV enable |
| | | 0 | Write 0 before enabling again |
| | | 1 | Enable loading of KSV from OTP |
| 4-0 | EPCM | | Command. Before writing a new value into this register, verify that the previous command is complete by checking the CMDD bit in the EPST register. |
| | | 3h | Run all BIST tests |
| | | 4h | Run only CRC test |
| | | 8h | Run only BIST self authentication test 1 (16-bit CRC to verify OTP contents) |
| | | 10h | Run only BIST self authentication test 2 (2 pass authentication to verify the HDCP cipher engine) |
| | | | All other values are reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.3 HDMI IP Core Gamut Registers

Table 13-152 lists the HDMI IP core gamut registers.

**Table 13-152. HDMI IP Core Gamut Registers**

| Address Offset | Acronym | Register Name |
|---|---|---|
| 00h | GAMUT_HEADER1 | Gamut Metadata Register |
| 04h | GAMUT_HEADER2 | Gamut Metadata Register |
| 08h | GAMUT_HEADER3 | Gamut Metadata Register |
| 0Ch-78h | GAMUT_DBYTE__0 - GAMUT_DBYTE__27 | Gamut Metadata Registers |

#### 13.3.3.1 Gamut Metadata Register (GAMUT_HEADER1)

The gamut metadata register is shown in Figure 13-143 and described in Table 13-153.

**Figure 13-143. Gamut Metadata Register (GAMUT_HEADER1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | HEADER1 | |
| R-0h | | R/W-Ah | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-153. Gamut Metadata Register (GAMUT_HEADER1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | HEADER1 | Gamut metadata header information |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.3.2 Gamut Metadata Register (GAMUT_HEADER2)

The gamut metadata register is shown in Figure 13-144 and described in Table 13-154.

**Figure 13-144. Gamut Metadata Register (GAMUT_HEADER2)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0h | |

| 7 | 6 | 4 | 3 | 0 |
|---|---|---|---|---|
| NEXT_FIELD | GBD_PROFILE | | AFF_GAM_SEQ_NUM | |
| R/W-0h | R/W-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-154. Gamut Metadata Register (GAMUT_HEADER2) Field Descriptions**

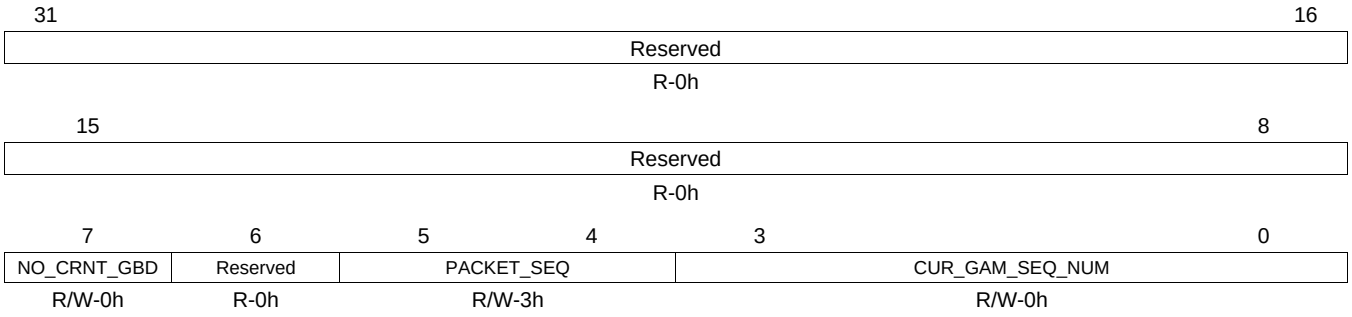
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | NEXT_FIELD | 0 | Indicates that the GBD will be effective on the next video field. NEXT_FIELD should be set even if the GBD is already effective (for example, Current = Affected) |
| 6-4 | GBD_PROFILE | | Transmission profile number. Values from 4h-7h are reserved. |
| | | 0 | P0 |
| | | 1h | P1 |
| | | 2h | P2 |
| | | 3h | P3 |
| 3-0 | AFF_GAM_SEQ_NUM | 0 | Indicates which video fields are relevant for this metadata. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.3.3 Gamut Metadata Register (GAMUT_HEADER3)

The gamut metadata register is shown in Figure 13-145 and described in Table 13-155.

**Figure 13-145. Gamut Metadata Register (GAMUT_HEADER3)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

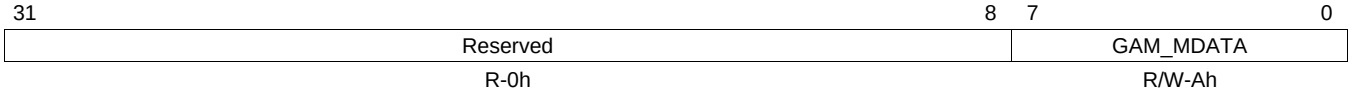
**Table 13-155. Gamut Metadata Register (GAMUT_HEADER3) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | NO_CRNT_GBD | 0 | Indicates no gamut metadata available for currently transmitted video. When set to 1, CUR_GAM_SEQ_NUM shall be ignored by the sink. |
| 6 | Reserved | 0 | Reserved |
| 5-4 | PACKET_SEQ | | Indicates the position of current packet. |
| | | 0 | Intermediate packet |
| | | 1h | First packet |
| | | 2h | Last packet |
| | | 3h | Only packet in sequence |
| 3-0 | CUR_GAM_SEQ_NUM | 0 | Indicates the gamut number of the currently transmitted video stream. |

### 13.3.3.4 Gamut Metadata Registers (GAMUT_DBYTE__0-GAMUT_DBYTE__27)

The gamut metadata registers are shown in Figure 13-146 and described in Table 13-156.

**Figure 13-146. Gamut Metadata Registers (GAMUT_DBYTE__0-GAMUT_DBYTE__27)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-156. Gamut Metadata Registers (GAMUT_DBYTE__0-GAMUT_DBYTE__27) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | GAM_MDATA | Gamut metadata data bytes |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4 HDMI IP Core Audio Video Registers

Table 13-157 lists the HDMI IP core audio video registers.

**Table 13-157. HDMI IP Core Audio Video Registers**

| Address Offset | Acronym | Register Name |
|---|---|---|
| 04h | ACR_CTRL | ACR Control Register |
| 08h | FREQ_SVAL | ACR Audio Frequency Register |
| 0Ch | N_SVAL1 | ACR N Software Value Register |
| 10h | N_SVAL2 | ACR N Software Value Register |
| 14h | N_SVAL3 | ACR N Software Value Register |
| 18h | CTS_SVAL1 | ACR CTS Software Value Register |
| 1Ch | CTS_SVAL2 | ACR CTS Software Value Register |
| 20h | CTS_SVAL3 | ACR CTS Software Value Register |
| 24h | CTS_HVAL1 | ACR CTS Hardware Value Register |
| 28h | CTS_HVAL2 | ACR CTS Hardware Value Register |
| 2Ch | CTS_HVAL3 | ACR CTS Hardware Value Register |
| 50h | AUD_MODE | Audio In Mode Register |
| 54h | SPDIF_CTRL | Audio In S/PDIF Control Register |
| 60h | HW_SPDIF_FS | Audio In S/PDIF Extracted Fs and Length Register |
| 64h | SWAP_I2S | Audio In I2S Channel Swap Register |
| 6Ch | SPDIF_ERTH | Audio Error Threshold Register |
| 70h | I2S_IN_MAP | Audio In I2S Data In Map Register |
| 74h | I2S_IN_CTRL | Audio In I2S Control Register |
| 78h | I2S_CHST0 | Audio In I2S Channel Status Registers |
| 7Ch | I2S_CHST1 | Audio In I2S Channel Status Registers |
| 80h | I2S_CHST2 | Audio In I2S Channel Status Registers |
| 84h | I2S_CHST4 | Audio In I2S Channel Status Registers |
| 88h | I2S_CHST5 | Audio In I2S Channel Status Registers |
| 8Ch | ASRC | Audio Sample Rate Conversion Register |
| 90h | I2S_IN_LEN | Audio I2S Input Length Register |
| BCh | HDMI_CTRL | HDMI Control Register |
| C0h | AUDO_TXSTAT | Audio Path Status Register |
| CCh | AUD_PAR_BUSCLK_1 | Audio Input Data Rate Adjustment Register |
| D0h | AUD_PAR_BUSCLK_2 | Audio Input Data Rate Adjustment Register |
| D4h | AUD_PAR_BUSCLK_3 | Audio Input Data Rate Adjustment Register |
| F0h | TEST_TXCTRL | Test Control Register |
| F4h | DPD | Diagnostic Power Down Register |
| F8h | PB_CTRL1 | Packet Buffer Control 1 Register |
| FCh | PB_CTRL2 | Packet Buffer Control 2 Register |
| 100h | AVI_TYPE | Packet Registers |
| 104h | AVI_VERS | Packet Registers |
| 108h | AVI_LEN | Packet Registers |
| 10Ch | AVI_CHSUM | Packet Registers |
| 110h-148h | AVI_DBYTE_0 - AVI_DBYTE_14 | Packet Registers |
| 180h | SPD_TYPE | SPD InfoFrame Registers |
| 184h | SPD_VERS | SPD InfoFrame Registers |
| 188h | SPD_LEN | SPD InfoFrame Registers |
| 18Ch | SPD_CHSUM | SPD InfoFrame Registers |
| 190h-1F8h | SPD_DBYTE_0 - SPD_DBYTE_26 | SPD InfoFrame Registers |
| 200h | AUDIO_TYPE | Audio InfoFrame Registers |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 13-157. HDMI IP Core Audio Video Registers (continued)**

| Address Offset | Acronym | Register Name |
|---|---|---|
| 204h | AUDIO_VERS | Audio InfoFrame Registers |
| 208h | AUDIO_LEN | Audio InfoFrame Registers |
| 20Ch | AUDIO_CHSUM | Audio InfoFrame Registers |
| 210h-234h | AUDIO_DBYTE_0 - AUDIO_DBYTE_9 | Audio InfoFrame Registers |
| 280h | MPEG_TYPE | MPEG InfoFrame Registers |
| 284h | MPEG_VERS | MPEG InfoFrame Registers |
| 288h | MPEG_LEN | MPEG InfoFrame Registers |
| 28Ch | MPEG_CHSUM | MPEG InfoFrame Registers |
| 290h-2F8h | MPEG_DBYTE_0 - MPEG_DBYTE_26 | MPEG InfoFrame Registers |
| 300h-378h | GEN_DBYTE_0 - GEN_DBYTE_30 | Generic Packet Registers |
| 37Ch | CP_BYTE1 | General Control Packet Register |
| 380h-3F8h | GEN2_DBYTE_0 - GEN2_DBYTE_30 | Generic Packet 2 Registers |
| 3FCh | CEC_ADDR_ID | CEC Slave ID Register |

### 13.3.4.1 ACR Control Register (ACR_CTRL)

The ACR control register is shown in Figure 13-147 and described in Table 13-158.

**Figure 13-147. ACR Control Register (ACR_CTRL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | NCTSPKT_EN | CTS_SEL |
| R-0h | | R/W-1h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-158. ACR Control Register (ACR_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1 | NCTSPKT_EN | | CTS request enable |
| | | 0 | N/CTS packet is disabled |
| | | 1 | N/CTS packet is enabled |
| 0 | CTS_SEL | | CTS source select |
| | | 0 | Send hardware-updated CTS value in N/CTS packet (recommended) |
| | | 1 | Send software-updated CTS value in N/CTS packet (for diagnostic use) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.2 ACR Audio Frequency Register (FREQ_SVAL)

The ACR audio frequency register is shown in Figure 13-148 and described in Table 13-159.

**Figure 13-148. ACR Audio Frequency Register (FREQ_SVAL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 3 | 2 | 0 |
|---|---|---|---|
| Reserved | | MCLK_CONF | |
| R-0h | | R/W-1h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-159. ACR Audio Frequency Register (FREQ_SVAL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2-0 | MCLK_CONF | | MCLK input mode. The HDMI transmitter uses these bits to divide the MCLK input to produce CTS values according to the 128 × Fs formula. The MCLK to Fs ratio is for the input Fs, not the down-sampled output Fs (see Section 13.3.4.25). |
| | | 0 | MCLK is 128 × Fs |
| | | 1h | MCLK is 256 × Fs |
| | | 2h | MCLK is 384 × Fs |
| | | 3h | MCLK is 512 × Fs |
| | | 4h | MCLK is 768 × Fs |
| | | 5h | MCLK is 1024 × Fs |
| | | 6h | MCLK is 1152 × Fs |
| | | 7h | MCLK is 192 × Fs |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.3 ACR N Software Value Register 1 (N_SVAL1)

The ACR N software value register is shown in Figure 13-149 and described in Table 13-160.

**Figure 13-149. ACR N Software Value Register 1 (N_SVAL1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | N_SVAL | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-160. ACR N Software Value Register 1 (N_SVAL1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | N_SVAL | N value for audio clock regeneration method, a 20-bit value. This must be written to the registers to create the correct divisor for audio clock regeneration. Only values greater than 0 are valid. This register must be written after a hardware reset. |

### 13.3.4.4 ACR N Software Value Register 2 (N_SVAL2)

The ACR N software value register is shown in Figure 13-150 and described in Table 13-161.

**Figure 13-150. ACR N Software Value Register 2 (N_SVAL2)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | N_SVAL | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-161. ACR N Software Value Register 2 (N_SVAL2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | N_SVAL | N value for audio clock regeneration method, a 20-bit value. This must be written to the registers to create the correct divisor for audio clock regeneration. Only values greater than 0 are valid. This register must be written after a hardware reset. |

### 13.3.4.5 ACR N Software Value Register 3 (N_SVAL3)

**Figure 13-151. ACR N Software Value Register 3 (N_SVAL3)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | N_SVAL | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-162. ACR N Software Value Register 3 (N_SVAL3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reserved |
| 3-0 | N_SVAL | N Value for audio clock regeneration method; a 20-bit value. This must be written to the registers to create the correct divisor for audio clock regeneration. Only values greater than 0 are valid. This register must be written after a hardware reset. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.6 ACR CTS Software Value Register 1 (CTS_SVAL1)

**Figure 13-152. ACR CTS Software Value Register 1 (CTS_SVAL1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CTS_SVAL | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-163. ACR CTS Software Value Register 1 (CTS_SVAL1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CTS_SVAL | CTS Value for audio clock regeneration method; a 20-bit value. Diagnostic use and applied only when the CTS_SEL bit (in register ACR_CTRL) is set to 1. |

### 13.3.4.7 ACR CTS Software Value Register 2 (CTS_SVAL2)

**Figure 13-153. ACR CTS Software Value Register 2 (CTS_SVAL2)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CTS_SVAL | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-164. ACR CTS Software Value Register 2 (CTS_SVAL2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CTS_SVAL | CTS Value for audio clock regeneration method; a 20-bit value. Diagnostic use and applied only when the CTS_SEL bit (in register ACR_CTRL) is set to 1. |

### 13.3.4.8 ACR CTS Software Value Register 3 (CTS_SVAL3)

**Figure 13-154. ACR CTS Software Value Register 3 (CTS_SVAL3)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | CTS_SVAL | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-165. ACR CTS Software Value Register 3 (CTS_SVAL3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reserved |
| 3-0 | CTS_SVAL | CTS Value for audio clock regeneration method; a 20-bit value. Diagnostic use and applied only when the CTS_SEL bit (in register ACR_CTRL) is set to 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.9 ACR CTS Hardware Value Register 1 (CTS_HVAL1)

**Figure 13-155. ACR CTS Hardware Value Register 1 (CTS_HVAL1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CTS_HVAL | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-166. ACR CTS Hardware Value Register 1 (CTS_HVAL1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CTS_HVAL | CTS Value for audio clock regeneration method; a 20-bit value. This value is measured and stored here by the hardware when MCLK is active and N is valid. |

### 13.3.4.10 ACR CTS Hardware Value Register 2 (CTS_HVAL2)

**Figure 13-156. ACR CTS Hardware Value Register 2 (CTS_HVAL2)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CTS_HVAL | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-167. ACR CTS Hardware Value Register 2 (CTS_HVAL2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CTS_HVAL | CTS Value for audio clock regeneration method; a 20-bit value. This value is measured and stored here by the hardware when MCLK is active and N is valid. |

### 13.3.4.11 ACR CTS Hardware Value Register 3 (CTS_HVAL3)

**Figure 13-157. ACR CTS Hardware Value Register 3 (CTS_HVAL3)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | CTS_HVAL | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-168. ACR CTS Hardware Value Register 3 (CTS_HVAL3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reserved |
| 3-0 | CTS_HVAL | CTS Value for audio clock regeneration method; a 20-bit value. This value is measured and stored here by the hardware when MCLK is active and N is valid. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.12 Audio In Mode Register (AUD_MODE)

**Figure 13-158. Audio In Mode Register (AUD_MODE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SD3_EN | SD2_EN | SD1_EN | SD0_EN | DSD_EN | AUD_PAR_EN | SPDIF_EN | AUD_EN |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-169. Audio In Mode Register (AUD_MODE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | SD3_EN | | I2S input channel 3 |
| | | 0 | Disable |
| | | 1 | Enable |
| 6 | SD2_EN | | I2S input channel 2 |
| | | 0 | Disable |
| | | 1 | Enable |
| 5 | SD1_EN | | I2S input channel 1 |
| | | 0 | Disable |
| | | 1 | Enable |
| 4 | SD0_EN | | I2S input channel 0 |
| | | 0 | Disable |
| | | 1 | Enable |
| 3 | DSD_EN | | Direct Stream Digital Audio enable |
| | | 0 | Disable |
| | | 1 | Enable |
| 2 | AUD_PAR_EN | | Parallel audio enable |
| | | 0 | Disable |
| | | 1 | Enable |
| 1 | SPDIF_EN | | S/PDIF input stream enable |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | AUD_EN | | Audio input stream enable |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.13 Audio In S/PDIF Control Register (SPDIF_CTRL)

**Figure 13-159. Audio In S/PDIF Control Register (SPDIF_CTRL)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | NOAUDIO | Reserved | FS_OVERRIDE | Reserved |
| R-0h | | | | R-0h | R-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-170. Audio In S/PDIF Control Register (SPDIF_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | NOAUDIO | | No S/PDIF audio |
| | | 0 | Detected change on the S/PDIF input |
| | | 1 | No change detected on the S/PDIF input |
| 2 | Reserved | 0 | Reserved |
| 1 | FS_OVERRIDE | | S/PDIF input stream override |
| | | 0 | Use input S/PDIF stream s detected FS |
| | | 1 | Use software FS in I2S_CHST4 register |
| 0 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.14 Audio In S/PDIF Extracted Fs and Length Register (HW_SPDIF_FS)

**Figure 13-160. Audio In S/PDIF Extracted Fs and Length Register (HW_SPDIF_FS)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| HW_SPDIF_LEN | | | HW_MAXLEN | HW_SPDIF_FS | | | |
| R-0h | | | R-0h | R-0h | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-171. Audio In S/PDIF Extracted Fs and Length Register (HW_SPDIF_FS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-5 | HW_SPDIF_LEN | | Channel status bits 33 to 35 (bit 33 = LSB). Combines with HW_MAXLEN bit to indicate sample size. |
| | | | **When HW_MAXLEN = 0 (20 bits):** |
| | | 0 | Reserved |
| | | 1h | 16 bits |
| | | 2h | 18 bits |
| | | 3h | Reserved |
| | | 4h | 19 bits |
| | | 5h | 20 bits |
| | | 6h | 17 bits |
| | | 7h | Reserved |
| | | | **When HW_MAXLEN = 1 (24 bits):** |
| | | 0 | Reserved |
| | | 1h | 20 bits |
| | | 2h | 22 bits |
| | | 3h | Reserved |
| | | 4h | 23 bits |
| | | 5h | 24 bits |
| | | 6h | 21 bits |
| | | 7h | Reserved |
| 4 | HW_MAXLEN | | Maximum sample length (channel status bit 32) |
| | | 0 | 20 bits |
| | | 1 | 24 bits |
| 3-0 | HW_SPDIF_FS | 0-Fh | Set to the FS extracted from the S/PDIF input channel status bits 24-27. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.15 Audio In I2S Channel Swap Register (SWAP_I2S)

**Figure 13-161. Audio In I2S Channel Swap Register (SWAP_I2S)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SWCH3 | SWCH2 | SWCH1 | SWCH0 | RSVDRW | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-9h | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

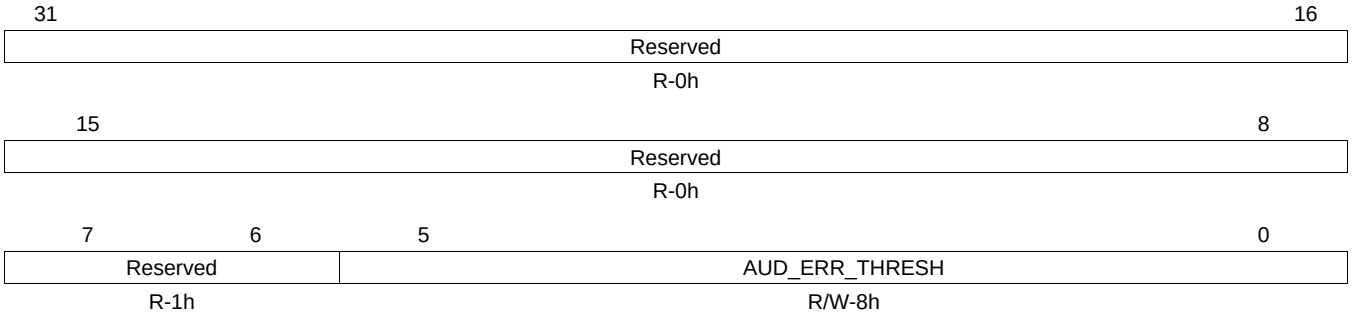
**Table 13-172. Audio In I2S Channel Swap Register (SWAP_I2S) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | SWCH3 | | Swap left-right channels for I2S Channel 3 |
| | | 0 | Do not swap left and right |
| | | 1 | Swap left and right |
| 6 | SWCH2 | | Swap left-right channels for I2S Channel 2 |
| | | 0 | Do not swap left and right |
| | | 1 | Swap left and right |
| 5 | SWCH1 | | Swap left-right channels for I2S Channel 1 |
| | | 0 | Do not swap left and right |
| | | 1 | Swap left and right |
| 4 | SWCH0 | | Swap left-right channels for I2S Channel 0 |
| | | 0 | Do not swap left and right |
| | | 1 | Swap left and right |
| 3-0 | RSVDRW | 9h | Reserved. Do not modify. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.16 Audio Error Threshold Register (SPDIF_ERTH)

**Figure 13-162. Audio Error Threshold Register (SPDIF_ERTH)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-173. Audio Error Threshold Register (SPDIF_ERTH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-6 | Reserved | 1 | Reserved |
| 5-0 | AUD_ERR_THRESH | 8h | Specifies the error threshold level. The frame is marked as invalid if the number of bi-phase mark encoding errors in the audio stream exceeds this threshold level during frame decoding. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.17 Audio In I2S Data In Map Register (I2S_IN_MAP)

**Figure 13-163. Audio In I2S Data In Map Register (I2S_IN_MAP)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FIFO3_MAP | | FIFO2_MAP | | FIFO1_MAP | | FIFO0_MAP | |
| R/W-3h | | R/W-2h | | R/W-1h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-174. Audio In I2S Data In Map Register (I2S_IN_MAP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-6 | FIFO3_MAP | | Channel map to FIFO3 (for HDMI Layout 1). |
| | | 0 | Map SD0 to FIFO3 |
| | | 1h | Map SD1 to FIFO3 |
| | | 2h | Map SD2 to FIFO3 |
| | | 3h | Map SD3 to FIFO3 |
| 5-4 | FIFO2_MAP | | Channel map to FIFO2 (for HDMI Layout 1). |
| | | 0 | Map SD0 to FIFO2 |
| | | 1h | Map SD1 to FIFO2 |
| | | 2h | Map SD2 to FIFO2 |
| | | 3h | Map SD3 to FIFO2 |
| 3-2 | FIFO1_MAP | | Channel map to FIFO1 (for HDMI Layout 1). |
| | | 0 | Map SD0 to FIFO1 |
| | | 1h | Map SD1 to FIFO1 |
| | | 2h | Map SD2 to FIFO1 |
| | | 3h | Map SD3 to FIFO1 |
| 1-0 | FIFO0_MAP | | Channel map to FIFO0 (for HDMI Layout 1). |
| | | 0 | Map SD0 to FIFO0 |
| | | 1h | Map SD1 to FIFO0 |
| | | 2h | Map SD2 to FIFO0 |
| | | 3h | Map SD3 to FIFO0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.18 Audio In I2S Control Register (I2S_IN_CTRL)

**Figure 13-164. Audio In I2S Control Register (I2S_IN_CTRL)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| HBRA_ON | SCK_EDGE | CBIT_ORDER | VBIT | I2S_WS | I2S_JUST | I2S_DIR | I2S_SHIFT |
| R/W-0h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R/W-0h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-175. Audio In I2S Control Register (I2S_IN_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | HBRA_ON | | High Bit Rate Audio On. All of the I2S control bits will apply to the control of the High Bit Rate Audio. |
| | | 0 | Input stream is not high bit rate |
| | | 1 | Input stream is high bit rate |
| 6 | SCK_EDGE | | SCK sample edge |
| | | 0 | Sample edge is falling; SD3-SD0 and WS source should change state on the rising edge of SCK |
| | | 1 | Sample clock is rising; SD3-SD0 and WS source should change state on the falling edge of SCK |
| 5 | CBIT_ORDER | 0 | This bit should be set to 1 for High Bit Rate Audio. |
| 4 | VBIT | | V bit value |
| | | 0 | PCM |
| | | 1 | Compressed |
| 3 | I2S_WS | | WS polarity |
| | | 0 | Left polarity when WS is LOW |
| | | 1 | Left polarity when WS is HIGH |
| 2 | I2S_JUST | | SD justify |
| | | 0 | Data is left-justified |
| | | 1 | Data is right-justified |
| 1 | I2S_DIR | | SD direction |
| | | 0 | MSB shifted first |
| | | 1 | LSB shifted first |
| 0 | I2S_SHIFT | | WS to SD first bit shift |
| | | 0 | First bit shift (refer to the Philips Specification) |
| | | 1 | No shift |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.19 Audio In I2S Channel Status Register 0 (I2S_CHST0)

**Figure 13-165. Audio In I2S Channel Status Register 0 (I2S_CHST0)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | I2S_CHST0 | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-176. I2S_CHST0 Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | I2S_CHST0 | Channel Status Byte 0 |

### 13.3.4.20 Audio In I2S Channel Status Register 1 (I2S_CHST1)

**Figure 13-166. Audio In I2S Channel Status Register 0 (I2S_CHST1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | I2S_CHST1 | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-177. I2S_CHST1 Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | I2S_CHST1 | Channel Status Byte 1: Category Code |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.21 Audio In I2S Channel Status Register 2 (I2S_CHST2)

**Figure 13-167. Audio In I2S Channel Status Register 2 (I2S_CHST2)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| I2S_CHAN_NUM | | | | I2S_SRC_NUM | | | |
| R/W-0h | | | | R/W-0h | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-178. Audio In I2S Channel Status Register 2 (I2S_CHST2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-4 | I2S_CHAN_NUM | Channel Status Byte 2: Source Number |
| 3-0 | I2S_SRC_NUM | Channel Status Byte 2: Source Number |

### 13.3.4.22 Audio In I2S Channel Status Register 3 (I2S_CHST3)

**Figure 13-168. Audio In I2S Channel Status Register 3 (I2S_CHST3)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| I2S_CHAN_NUM | | | | I2S_SRC_NUM | | | |
| R/W-0h | | | | R/W-0h | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-179. Audio In I2S Channel Status Register 3 (I2S_CHST3) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-4 | I2S_CHAN_NUM | Channel Status Byte 2: Source Number |
| 3-0 | I2S_SRC_NUM | Channel Status Byte 2: Source Number |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.23 Audio In I2S Channel Status Register 4 (I2S_CHST4)

**Figure 13-169. Audio In I2S Channel Status Register 4 (I2S_CHST4)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CLK_ACCUR | | | | SW_SPDIF_FS | | | |
| R/W-0h | | | | R/W-Fh | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

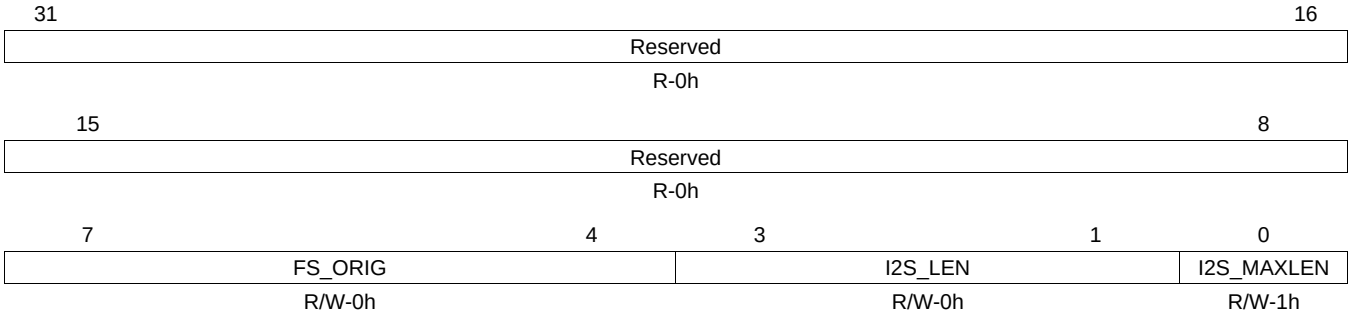
**Table 13-180. Audio In I2S Channel Status Register 4 (I2S_CHST4) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-4 | CLK_ACCUR | Clock Accuracy |
| 3-0 | SW_SPDIF_FS | Sampling frequency as set by software |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.24 Audio In I2S Channel Status Register 5 (I2S_CHST5)

**Figure 13-170. Audio In I2S Channel Status Register 5 (I2S_CHST5)**




LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

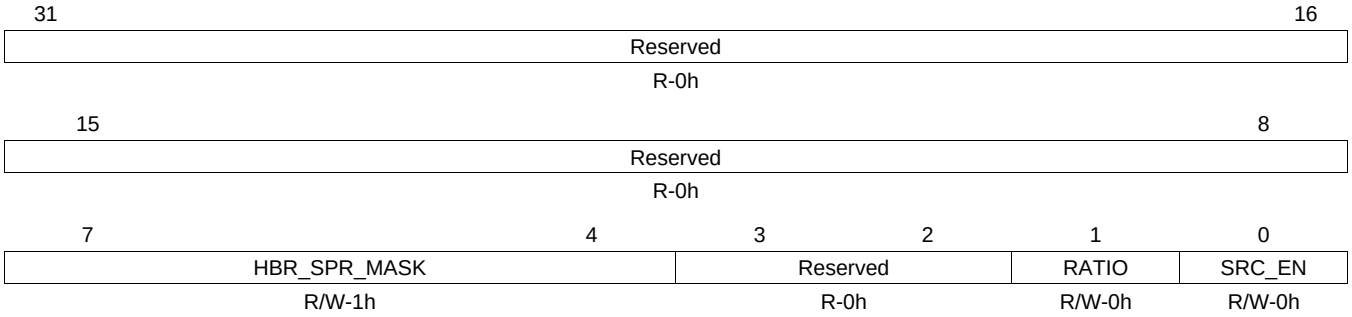
**Table 13-181. Audio In I2S Channel Status Register 5 (I2S_CHST5) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-4 | FS_ORIG | 0 | Original Fs |
| 3-1 | I2S_LEN | | Audio sample word length. Defined in bits with I2S_MAXLEN. |
| | | | **When I2S_MAXLEN = 0 (20 bits):** |
| | | 0 | Reserved |
| | | 1h | 16 bits |
| | | 2h | 18 bits |
| | | 3h | Reserved |
| | | 4h | 19 bits |
| | | 5h | 20 bits |
| | | 6h | 17 bits |
| | | 7h | Reserved |
| | | | **When I2S_MAXLEN = 1 (24 bits):** |
| | | 0 | Reserved |
| | | 1h | 20 bits |
| | | 2h | 22 bits |
| | | 3h | Reserved |
| | | 4h | 23 bits |
| | | 5h | 24 bits |
| | | 6h | 21 bits |
| | | 7h | Reserved |
| 0 | I2S_MAXLEN | | Maximum audio sample word length. |
| | | 0 | 20 bits |
| | | 1 | 24 bits |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.25 Audio Sample Rate Conversion Register (ASRC)

**Figure 13-171. Audio Sample Rate Conversion Register (ASRC)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-182. Audio Sample Rate Conversion Register (ASRC) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-4 | HBR_SPR_MASK | | Mask for the sample present and flat bit of the High Bit Rate Audio header. Each bit masks out one of the subpacket sample present bits. |
| | | 0 | Mask out |
| | | 1 | Unmask bits 7-4 must be programmed to 0 when HBRA mode is selected |
| 3-2 | Reserved | 0 | Reserved |
| 1 | RATIO | | Sample rate down-conversion ratio when the SRC_EN bit is set to 1. |
| | | 0 | Down-sample is 2-to-1 |
| | | 1 | Down-sample is 4-to-1 |
| 0 | SRC_EN | | Audio sample rate conversion |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.26 Audio I2S Input Length Register (I2S_IN_LEN)

**Figure 13-172. Audio I2S Input Length Register (I2S_IN_LEN)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| HDR_PKT_ID | | | | IN_LENGTH | | | |
| R/W-9h | | | | R/W-Bh | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

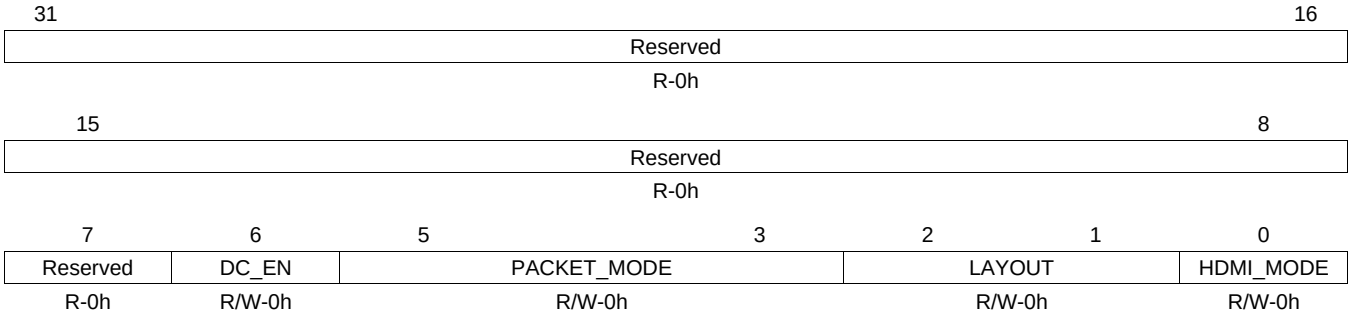
**Table 13-183. Audio I2S Input Length Register (I2S_IN_LEN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-4 | HDR_PKT_ID | 9h | The ID of the High Bit Rate Audio packet header |
| 3-0 | IN_LENGTH | | Number of valid bits in the input I2S stream. Used for the extraction of the I2S data from the input stream. |
| | | 0-1h | Reserved |
| | | 2h | 16 bits |
| | | 3h | Reserved |
| | | 4h | 18 bits |
| | | 5h | 22 bits |
| | | 6h-7h | Reserved |
| | | 8h | 19 bits |
| | | 9h | 23 bits |
| | | Ah | 20 bits |
| | | Bh | 24 bits |
| | | Ch | 17 bits |
| | | Dh | 21 bits |
| | | Eh-Fh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.27 HDMI Control Register (HDMI_CTRL)

**Figure 13-173. HDMI Control Register (HDMI_CTRL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-184. HDMI Control Register (HDMI_CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | DC_EN | | Deep-color packet enable.<br>The following data is sent in data byte 1 of the packet: 7 = Reserved, 6 = PP2, 5 = PP1, 4 = PP0, 3 = PP_valid, 2 = CD2, 1 = CD1, 0 = CD0, The CD bits indicate deep-color mode (defined in the same register bits 5:3). The PP bits indicate the fragment s phase related information that comes from the hardware state machine. |
| | | 0 | Do not send deep-color related information in the packet to the HDMI Receiver. |
| | | 1 | Send deep-color related information in the packet to the HDMI Receiver. |
| 5-3 | PACKET_MODE | | Specifies the number of bits per pixel sent to the paketizer. The firmware must program 24 bits per pixel (8 bits per pixel; no packing) for initialization. |
| | | 0-3h | Reserved |
| | | 4h | 24 bits per pixel (8 bits per pixel; no packing) |
| | | 5h | 30 bits per pixel (10 bits per pixel pack to 8 bits) |
| | | 6h | 36 bits per pixel (12 bits per pixel pack to 8 bits) |
| | | 7h | 48 bits per pixel (16 bits per pixel; no packing) |
| 2-1 | LAYOUT | | Audio packet header layout indicator. |
| | | 0 | Layout 0 (2-channel) |
| | | 1h | Layout 1 (up to 8 channels) |
| | | 2h-3h | Reserved |
| 0 | HDMI_MODE | | HDMI Mode |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.28 Audio Path Status Register (AUDO_TXSTAT)

**NOTE:** MUTE is not set immediately when the SETAVM bit in register CP_BYTE1 is written. After writing SETAVM to 1, MUTE is set after the Control Packet is transmitted in HDMI mode. In DVI mode, MUTE is set at the start of VSYNC. MUTE is cleared when a Control Packet with CLRAVM = 1 is sent, or (in DVI mode) at the start of VSYNC after CLRAVM has been written to 1. Note that the combinations {SETAVM, CLRAVM} = {0,0} and {1,1} are not supported by HDMI. Such a packet is not compliant.

**Figure 13-174. Audio Path Status Register (AUDO_TXSTAT)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0h | |

| 7 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | MUTE | NPACKET_EN_VS_HIGH | NPACKET_EN |
| R-0h | | R-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-185. Audio Path Status Register (AUDO_TXSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2 | MUTE | | General Control Packet mute status |
| | | 0 | No packet with SETAVM = 1 has been sent |
| | | 1 | A packet with SETAVM = 1 has been sent |
| 1 | NPACKET_EN_VS_HIGH | 0 | Enables null packet flooding only when VSync is high. |
| 0 | NPACKET_EN | 0 | Enables null packet flooding all the time. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.29 Audio Input Data Rate Adjustment Register 1 (AUD_PAR_BUSCLK_1)

**Figure 13-175. Audio Input Data Rate Adjustment Register 1 (AUD_PAR_BUSCLK_1)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AUD_PAR_BUSCLK_1 | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-186. Audio Input Data Rate Adjustment Register 1 (AUD_PAR_BUSCLK_1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AUD_PAR_BUSCLK_1 | Decimal part of adjustment parameter |

### 13.3.4.30 Audio Input Data Rate Adjustment Register 2 (AUD_PAR_BUSCLK_2)

**Figure 13-176. Audio Input Data Rate Adjustment Register 2 (AUD_PAR_BUSCLK_2)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AUD_PAR_BUSCLK_2 | |
| R-0h | | R/W-8h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-187. AUD_PAR_BUSCLK_2 Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AUD_PAR_BUSCLK_2 | Lower byte of integer part of parameter |

### 13.3.4.31 Audio Input Data Rate Adjustment Register 3 (AUD_PAR_BUSCLK_3)

**Figure 13-177. Audio Input Data Rate Adjustment Register 3 (AUD_PAR_BUSCLK_3)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AUD_PAR_BUSCLK_3 | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

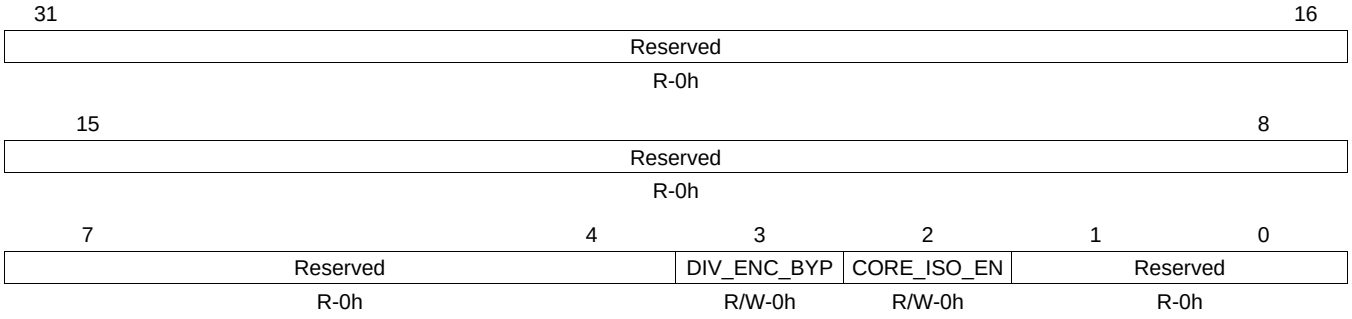
**Table 13-188. AUD_PAR_BUSCLK_3 Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AUD_PAR_BUSCLK_3 | Upper byte of integer part of parameter |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.32 Test Control Register (TEST_TXCTRL)

**Figure 13-178. Test Control Register (TEST_TXCTRL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-189. Test Control Register (TEST_TXCTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | DIV_ENC_BYP | | DVI encoder bypass |
| | | 0 | Normal operation |
| | | 1 | Bypass DVI encoder logic. Never set this bit. Always use the DVI encoder. |
| 2 | CORE_ISO_EN | | TMDS Core Isolation Enable |
| | | 0 | Normal operation |
| | | 1 | Input pins mixed to TMDS Tx core; to emulate discrete Tx. |
| 1-0 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.33 Diagnostic Power Down Register (DPD)

**Figure 13-179. Diagnostic Power Down Register (DPD)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VID_BYP_EN | Reserved | | | TCLKPHZ | PDIDCK | PDOSC | PDTOT |
| R/W-0h | R-0h | | | R/W-1h | R/W-1h | R/W-1h | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-190. Diagnostic Power Down Register (DPD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | VID_BYP_EN | | Enable bypath of the video path. The CORE_ISO_EN bit (in register TEST_TXCTRL) must be set and the HDCP cypher must be disabled (HDCP_CTRL.EWNC_EN = 0). |
| | | 0 | Disable |
| | | 1 | Enable |
| 6-4 | Reserved | 0 | Reserved |
| 3 | TCLKPHZ | | Selects the TCLK phase |
| | | 0 | Default phase; the same as TMDS core 1 |
| | | 1 | Invert TCLK; change the phase 180 degrees |
| 2 | PDIDCK | | Power down IDCK input |
| | | 0 | Power down, gate off IDCK signal to disable all IDCK-based logic |
| | | 1 | Normal operation |
| 1 | PDOSC | | Power down internal oscillator. This disables the I2C port to the internal ROM (disabling loading), halts all interrupt updates, and disables the Master DDC block. |
| | | 0 | Power down |
| | | 1 | Normal operation |
| 0 | PDTOT | | Power down total |
| | | 0 | Power down everything; INT source is RSEN |
| | | 1 | Normal operation |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.34 Packet Buffer Control 1 Register (PB_CTRL1)

**Figure 13-180. Packet Buffer Control 1 Register (PB_CTRL1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MPEG_EN | MPEG_RPT | AUD_EN | AUD_RPT | SPD_EN | SPD_RPT | AVI_EN | AVI_RPT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-191. Packet Buffer Control 1 Register (PB_CTRL1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | MPEG_EN | | Enable MPEG InfoFrame transmission |
| | | 0 | Disable |
| | | 1 | Enable |
| 6 | MPEG_RPT | | Repeat MPEG InfoFrame transmission |
| | | 0 | Disable (send once after enable bit is set) |
| | | 1 | Enable (send in every VBLANK period) |
| 5 | AUD_EN | | Enable Audio InfoFrame transmission |
| | | 0 | Disable |
| | | 1 | Enable |
| 4 | AUD_RPT | | Repeat Audio InfoFrame transmission |
| | | 0 | Disable (send once after enable bit is set) |
| | | 1 | Enable (send in every VBLANK period) |
| 3 | SPD_EN | | Enable General Control Packet transmission |
| | | 0 | Disable |
| | | 1 | Enable |
| 2 | SPD_RPT | | Repeat SPD InfoFrame transmission |
| | | 0 | Disable (send once after enable bit is set) |
| | | 1 | Enable (send in every VBLANK period) |
| 1 | AVI_EN | | Enable AVI InfoFrame transmission |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | AVI_RPT | | Repeat AVI InfoFrame transmission |
| | | 0 | Disable (send once after enable bit is set) |
| | | 1 | Enable (send in every VBLANK period) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.35 Packet Buffer Control 2 Register (PB_CTRL2)

**Figure 13-181. Packet Buffer Control 2 Register (PB_CTRL2)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| GAM_EN | GAM_RPT | GEN2_EN | GEN2_RPT | CP_EN | CP_RPT | GEN_EN | GEN_RPT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-192. Packet Buffer Control 2 Register (PB_CTRL2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | GAM_EN | | Enable Gamut Metadata InfoFrame transmission on HDMI. |
| | | 0 | Disable |
| | | 1 | Enable |
| 6 | GAM_RPT | | Repeat Gamut Metadata InfoFrame Packet data each frame. |
| | | 0 | Disable (send once after enable bit is set) |
| | | 1 | Enable (send in every VBLANK period) |
| 5 | GEN2_EN | | Enable Generic 2 Packet transmission |
| | | 0 | Disable |
| | | 1 | Enable |
| 4 | GEN2_RPT | | Repeat Generic 2 Packet transmission |
| | | 0 | Disable (send once after enable bit is set) |
| | | 1 | Enable (send in every VBLANK period) |
| 3 | CP_EN | | Enable General Control Packet transmission |
| | | 0 | Disable |
| | | 1 | Enable |
| 2 | CP_RPT | | Repeat General Control Packet transmission |
| | | 0 | Disable (send once after enable bit is set) |
| | | 1 | Enable (send in every VBLANK period) |
| 1 | GEN_EN | | Enable Generic Packet transmission |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | GEN_RPT | | Repeat Generic Packet transmission |
| | | 0 | Disable (send once after enable bit is set) |
| | | 1 | Enable (send in every VBLANK period) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.36 AVI InfoFrame Register (AVI_TYPE)

**Figure 13-182. AVI InfoFrame Register (AVI_TYPE)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AVI_TYPE | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-193. AVI InfoFrame Register (AVI_TYPE) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AVI_TYPE | AVI InfoFrame Type Code. AVI_HDR[7:0] |

### 13.3.4.37 AVI InfoFrame Register (AVI_VERS)

**Figure 13-183. AVI InfoFrame Register (AVI_VERS)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AVI_VERS | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-194. AVI InfoFrame Register (AVI_VERS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AVI_VERS | AVI InfoFrame Version Code. AVI_HDR[15:8] |

### 13.3.4.38 AVI InfoFrame Register (AVI_LEN)

**Figure 13-184. AVI InfoFrame Register (AVI_LEN)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AVI_LEN | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-195. AVI InfoFrame Register (AVI_LEN) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AVI_LEN | AVI InfoFrame Length. AVI_HDR[23:16] |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.39 AVI InfoFrame Register (AVI_CHSUM)

**Figure 13-185. AVI InfoFrame Register (AVI_CHSUM)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AVI_CHSUM | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-196. AVI InfoFrame Register (AVI_CHSUM) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AVI_CHSUM | AVI InfoFrame Checksum. AVI_HDR[31:24] |

### 13.3.4.40 AVI InfoFrame Registers (AVI_DBYTE_0-AVI_DBYTE_14)

**Figure 13-186. AVI InfoFrame Registers (AVI_DBYTE_0-AVI_DBYTE_14)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AVI_DATA | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-197. AVI InfoFrame Registers (AVI_DBYTE_0-AVI_DBYTE_14) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AVI_DATA | AVI InfoFrame Data Bytes. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.41 SPD InfoFrame Register (SPD_TYPE)

**Figure 13-187. SPD InfoFrame Register (SPD_TYPE)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | SPD_TYPE | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-198. SPD InfoFrame Register (SPD_TYPE) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | SPD_TYPE | SPD InfoFrame Type Code. SPD_HDR[7:0] |

### 13.3.4.42 SPD InfoFrame Register (SPD_VERS)

**Figure 13-188. SPD InfoFrame Register (SPD_VERS)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | SPD_VERS | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-199. SPD InfoFrame Register (SPD_VERS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | SPD_VERS | SPD InfoFrame Version Code. SPD_HDR[15:8] |

### 13.3.4.43 SPD InfoFrame Register (SPD_LEN)

**Figure 13-189. SPD InfoFrame Register (SPD_LEN)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | SPD_LEN | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-200. SPD InfoFrame Register (SPD_LEN) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | SPD_LEN | SPD InfoFrame Length. SPD_HDR[23:16] |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.44 SPD InfoFrame Register (SPD_CHSUM)

**Figure 13-190. SPD InfoFrame Register (SPD_CHSUM)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | SPD_CHSUM | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-201. SPD InfoFrame Register (SPD_CHSUM) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | SPD_CHSUM | SPD InfoFrame Checksum. SPD_HDR[31:24] |

### 13.3.4.45 SPD InfoFrame Registers (SPD_DBYTE_0-SPD_DBYTE_26)

**Figure 13-191. SPD InfoFrame Registers (SPD_DBYTE_0-SPD_DBYTE_26)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | SPD_DATA | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-202. SPD InfoFrame Registers (SPD_DBYTE_0-SPD_DBYTE_26) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | SPD_DATA | SPD InfoFrame Data Bytes. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.46 Audio InfoFrame Register (AUDIO_TYPE)

**Figure 13-192. Audio InfoFrame Register (AUDIO_TYPE)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AUDIO_TYPE | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-203. Audio InfoFrame Register (AUDIO_TYPE) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AUDIO_TYPE | AUDIO InfoFrame Type Code. AUDIO_HDR[7:0] |

### 13.3.4.47 Audio InfoFrame Register (AUDIO_VERS)

**Figure 13-193. Audio InfoFrame Register (AUDIO_VERS)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AUDIO_VERS | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-204. Audio InfoFrame Register (AUDIO_VERS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AUDIO_VERS | AUDIO InfoFrame Version Code. AUDIO_HDR[15:8] |

### 13.3.4.48 Audio InfoFrame Register (AUDIO_LEN)

**Figure 13-194. Audio InfoFrame Register (AUDIO_LEN)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AUDIO_LEN | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-205. Audio InfoFrame Register (AUDIO_LEN) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AUDIO_LEN | AUDIO InfoFrame Length. AUDIO_HDR[23:16] |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.49 Audio InfoFrame Register (AUDIO_CHSUM)

**Figure 13-195. Audio InfoFrame Register (AUDIO_CHSUM)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AUDIO_CHSUM | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-206. Audio InfoFrame Register (AUDIO_CHSUM) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AUDIO_CHSUM | AUDIO InfoFrame Checksum. AUDIO_HDR[31:24] |

### 13.3.4.50 Audio InfoFrame Registers (AUDIO_DBYTE_0-AUDIO_DBYTE_9)

**Figure 13-196. Audio InfoFrame Registers (AUDIO_DBYTE_0-AUDIO_DBYTE_9)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | AUDIO_DATA | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-207. Audio InfoFrame Registers (AUDIO_DBYTE_0-AUDIO_DBYTE_9) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | AUDIO_DATA | AUDIO InfoFrame Data Bytes |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.51 MPEG InfoFrame Register (MPEG_TYPE)

**Figure 13-197. MPEG InfoFrame Register (MPEG_TYPE)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | MPEG_TYPE | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-208. MPEG InfoFrame Register (MPEG_TYPE) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | MPEG_TYPE | MPEG InfoFrame Type Code. MPEG_HDR[7:0] |

### 13.3.4.52 MPEG InfoFrame Register (MPEG_VERS)

**Figure 13-198. MPEG InfoFrame Register (MPEG_VERS)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | MPEG_VERS | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-209. MPEG InfoFrame Register (MPEG_VERS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | MPEG_VERSN | MPEG InfoFrame Version Code. MPEG_HDR[15:8] |

### 13.3.4.53 MPEG InfoFrame Register (MPEG_LEN)

**Figure 13-199. MPEG InfoFrame Register (MPEG_LEN)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | MPEG_LEN | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-210. MPEG InfoFrame Register (MPEG_LEN) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | MPEG_LEN | MPEG InfoFrame Length. MPEG_HDR[23:16] |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.54 MPEG InfoFrame Register (MPEG_CHSUM)

**Figure 13-200. MPEG InfoFrame Register (MPEG_CHSUM)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | MPEG_CHSUM | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-211. MPEG InfoFrame Register (MPEG_CHSUM) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | MPEG_CHSUM | MPEG InfoFrame Checksum. MPEG_HDR[31:24] |

### 13.3.4.55 MPEG InfoFrame Registers (MPEG_DBYTE_0-MPEG_DBYTE_26)

**Figure 13-201. MPEG InfoFrame Registers (MPEG_DBYTE_0-MPEG_DBYTE_26)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | MPEG_DATA | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-212. MPEG InfoFrame Registers (MPEG_DBYTE_0-MPEG_DBYTE_26) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | MPEG_DATA | MPEG InfoFrame Data Bytes |

### 13.3.4.56 Generic Packet Registers (GEN_DBYTE_0-GEN_DBYTE_30)

**Figure 13-202. Generic Packet Registers (GEN_DBYTE_0-GEN_DBYTE_30)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | GEN_DATA | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-213. Generic Packet Registers (GEN_DBYTE_0-GEN_DBYTE_30) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | GEN_DATA | Generic Packet Data Bytes |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.57 General Control Packet Register (CP_BYTE1)

**Figure 13-203. General Control Packet Register (CP_BYTE1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CLRAVM | Reserved | | | SETAVM |
| R-0h | | | R/W-0h | R-0h | | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-214. General Control Packet Register (CP_BYTE1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reserved |
| 4 | CLRAVM | Clear AV Mute flag |
| 3-1 | Reserved | Reserved |
| 0 | SETAVM | Set AV Mute flag. When the AVMUTE flag is set, the HDMI Transmitter sends a General Control Packet on the TMDS link to inform the Sink that the data may be incorrect. The HDMI Transmitter sends blanklevel data for all video packets and 00 for all audio packet data. When the AVMUTE flag is set, the Sink assumes that no valid data is being received. Optionally, the Sink can apply a mute function to the audio data and/or a blank function to the video data. |

### 13.3.4.58 Generic Packet 2 Registers (GEN2_DBYTE_0-GEN2_DBYTE_30)

**Figure 13-204. Generic Packet 2 Registers (GEN2_DBYTE_0-GEN2_DBYTE_30)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| GEN2_DATA | | | | | | | |
| R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-215. Generic Packet 2 Registers (GEN2_DBYTE_0-GEN2_DBYTE_30) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | GEN2_DATA | Generic Packet 2 Data Bytes |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.4.59 CEC Slave ID Register (CEC_ADDR_ID)

**Figure 13-205. CEC Slave ID Register (CEC_ADDR_ID)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CEC_ID | |
| R-0h | | R/W-C0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-216. CEC Slave ID Register (CEC_ADDR_ID) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CEC_ID | CEC I2C slave address ID |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5 HDMI IP Core CEC Registers

Table 13-217 lists the HDMI IP core CEC registers.

**Table 13-217. HDMI IP Core CEC Registers**

| Address Offset | Acronym | Register Name |
|---|---|---|
| 00h | CEC_DEV_ID | CEC Device ID Register |
| 04h | CEC_SPEC | CEC Specification Register |
| 08h | CEC_SUFF | CEC Specification Suffix Register |
| 0Ch | CEC_FW | CEC Firmware Revision Register |
| 10h | CEC_DBG_0 | CEC Debug Register 0 |
| 14h | CEC_DBG_1 | CEC Debug Register 1 |
| 18h | CEC_DBG_2 | CEC Debug Register 2 |
| 1Ch | CEC_DBG_3 | CEC Debug Register 3 |
| 20h | CEC_TX_INIT | CEC Tx Initialization Register |
| 24h | CEC_TX_DEST | CEC Tx Destination Register |
| 38h | CEC_SETUP | CEC Setup Register |
| 3Ch | CEC_TX_COMMAND | CEC Tx Command Register |
| 40h-78h | CEC_TX_OPERAND_0 - CEC_TX_OPERAND_14 | CEC Tx Operand Registers |
| 7Ch | CEC_TRANSMIT_DATA | CEC Transmit Data Register |
| 88h | CEC_CA_7_0 | CEC Capture ID0 Register |
| 8Ch | CEC_CA_15_8 | CEC Capture ID0 Register |
| 90h | CEC_INT_ENABLE_0 | CEC Interrupt Enable Register 0 |
| 94h | CEC_INT_ENABLE_1 | CEC Interrupt Enable Register 1 |
| 98h | CEC_INT_STATUS_0 | CEC Interrupt Status Register 0 |
| 9Ch | CEC_INT_STATUS_1 | CEC Interrupt Status Register 1 |
| B0h | CEC_RX_CONTROL | CEC RX Control Register |
| B4h | CEC_RX_COUNT | CEC Rx Count Register |
| B8h | CEC_RX_CMD_HEADER | CEC Rx Command Header Register |
| BCh | CEC_RX_COMMAND | CEC Rx Command Register |
| C0h-F8h | CEC_RX_OPERAND_0 - CEC_RX_OPERAND_14 | CEC Rx Operand Registers |

#### 13.3.5.1 CEC Device ID Register (CEC_DEV_ID)

**Figure 13-206. CEC Device ID Register (CEC_DEV_ID)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CEC_DEV_ID | |
| R-0h | | R-CCh | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-218. CEC Device ID Register (CEC_DEV_ID) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CEC_DEV_ID | ID of CEC device |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.2 CEC Specification Register (CEC_SPEC)

**Figure 13-207. CEC Specification Register (CEC_SPEC)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CEC_REL | | | | CEC_REV | | | |
| R-1h | | | | R-3h | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-219. CEC Specification Register (CEC_SPEC) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-4 | CEC_REL | CEC Specification major release |
| 3-0 | CEC_REV | CEC Specification minor release |

### 13.3.5.3 CEC Specification Suffix Register (CEC_SUFF)

**Figure 13-208. EC Specification Suffix Register (CEC_SUFF)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SUB_SYS | Reserved | | | CEC_SUFF | | | |
| R-1h | R-0h | | | R-0h | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-220. EC Specification Suffix Register (CEC_SUFF) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | SUB_SYS | | Subsystem. |
| | | 0 | Firmware |
| | | 1 | Hardware |
| 6-4 | Reserved | 0 | Reserved |
| 3-0 | CEC_SUFF | 0 | CEC Specification Suffix (0 = a for rev 1.2a) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.4 CEC Firmware Revision Register (CEC_FW)

**Figure 13-209. CEC Firmware Revision Register (CEC_FW)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FW_REL_ID | | | | FW_REV_ID | | | |
| R-1h | | | | R-4h | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-221. CEC Firmware Revision Register (CEC_FW) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-4 | FW_REL_ID | Firmware Release ID |
| 3-0 | FW_REV_ID | Firmware Revision ID |

### 13.3.5.5 CEC Debug Register 0 (CEC_DBG_0)

**Figure 13-210. CEC Debug Register 0 (CEC_DBG_0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | STB_LOW_P |
| R-0h | | | | | | | R-0h |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-222. CEC Debug Register 0 (CEC_DBG_0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | STB_LOW_P | Start Bit Low Period. Measured in units of 250 µs. Expected range is 3.5 ms (Eh) to 3.9 ms (0Fh). |

### 13.3.5.6 CEC Debug Register 1 (CEC_DBG_1)

**Figure 13-211. CEC Debug Register 1 (CEC_DBG_1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | STB_DUR_P |
| R-0h | | | | | | | R-0h |

LEGEND: R = Read only; -*n* = value after reset

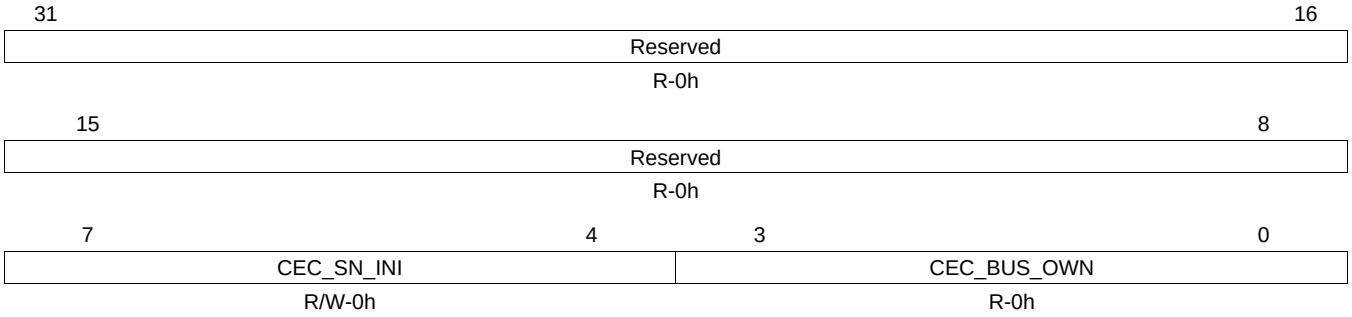
**Table 13-223. CEC Debug Register 1 (CEC_DBG_1) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | STB_DUR_P | Start Bit Duration Period. Measured in units of 250 µs. Expected range is 4.3 ms (11h) to 4.7 ms (12h). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.7 CEC Debug Register 2 (CEC_DBG_2)

**Figure 13-212. CEC Debug Register 2 (CEC_DBG_2)**




LEGEND: R = Read only; -*n* = value after reset

**Table 13-224. CEC Debug Register 2 (CEC_DBG_2) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-4 | CEC_SN_INI | CEC Snoop Initiator. Values 0 to Fh. |
| 3-0 | CEC_BUS_OWN | Current CEC bus owner. Values 0 to Fh. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.8 CEC Debug Register 3 (CEC_DBG_3)

**Figure 13-213. CEC Debug Register 3 (CEC_DBG_3)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FL_FIF | FR_RT_CNT | | | Reserved | INV_ACK | ACKN_HEAD | CEC_SN |
| R/W-0h | R-0h | | | R-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-225. CEC Debug Register 3 (CEC_DBG_3) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | FL_FIF | | Flush Tx FIFO |
| | | 0 | No |
| | | 1 | Yes, self-resetting bit |
| 6-4 | FR_RT_CNT | 0 | Frame Retransmit Count. Values 0 to 5. |
| 3 | Reserved | 0 | Reserved |
| 2 | INV_ACK | | Invert ACK to Broadcast Commands |
| | | 0 | No |
| | | 1 | Yes |
| 1 | ACKN_HEAD | | ACK/NACK Header Block |
| | | 0 | ACK |
| | | 1 | NACK |
| 0 | CEC_SN | | CEC snoop |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.9 CEC Tx Initialization Register (CEC_TX_INIT)

**Figure 13-214. CEC Tx Initialization Register (CEC_TX_INIT)**

| 31 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | CEC_INIT_ID | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-226. CEC Tx Initialization Register (CEC_TX_INIT) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-4 | Reserved | Reserved |
| 3-0 | CEC_INIT_ID | CEC Initiator ID |

### 13.3.5.10 CEC Tx Destination Register (CEC_TX_DEST)

**Figure 13-215. CEC Tx Destination Register (CEC_TX_DEST)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0h | |

| 7 | 6 | 4 | 3 | 0 |
|---|---|---|---|---|
| CEC_SD_POLL | Reserved | | CEC_DEST_ID | |
| R/W-0h | R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-227. CEC Tx Destination Register (CEC_TX_DEST) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | CEC_SD_POLL | | Generate a polling message |
| | | 0 | No |
| | | 1 | Yes, self-resetting bit |
| 6-4 | Reserved | 0 | Reserved |
| 3-0 | CEC_DEST_ID | 0 | CEC Destination ID. Identifies the target device for the command. Must be written before writing the corresponding CEC command. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.11 CEC Setup Register (CEC_SETUP)

**Figure 13-216. CEC Setup Register (CEC_SETUP)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | CEC_PTRH | Reserved | CEC_FORCE_NON_CALIB | CEC_CAL_EN | CEC_CAL_SW |
| R-0h | | | R/W-1h | R-0h | R/W-1h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-228. CEC Setup Register (CEC_SETUP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-5 | Reserved | Reserved |
| 4 | CEC_PTRH | CEC pass-thru register |
| 3 | Reserved | Reserved |
| 2 | CEC_FORCE_NON_CALIB | Must be set to 1 if calibration is not needed (If a 2 MHz crystal clock is available) |
| 1 | CEC_CAL_EN | CEC calibration enable register (self clearing) |
| 0 | CEC_CAL_SW | Must be set to 1 for 10 ms in case of calibration |

### 13.3.5.12 CEC Tx Command Register (CEC_TX_COMMAND)

**Figure 13-217. CEC Tx Command Register (CEC_TX_COMMAND)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | CEC_TX_COM | | | | | | | |
| R-0h | | | | | | | | R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-229. CEC Tx Command Register (CEC_TX_COMMAND) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CEC_TX_COM | CEC Tx Command |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.13 CEC Tx Operand Registers (CEC_TX_OPERAND_0-CEC_TX_OPERAND_14)

**Figure 13-218. CEC Tx Operand Registers (CEC_TX_OPERAND_0-CEC_TX_OPERAND_14)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CEC_TX_COM | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-230. CEC Tx Operand Registers (CEC_TX_OPERAND_0-CEC_TX_OPERAND_14) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CEC_TX_COM | CEC Tx Operand |

### 13.3.5.14 CEC Transmit Data Register (CEC_TRANSMIT_DATA)

**Figure 13-219. CEC Transmit Data Register (CEC_TRANSMIT_DATA)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0h | |

| 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | TX_BFR_ACC | TX_AUT_CALC | TRA_CEC_CMD | TX_CNT | |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-231. CEC Transmit Data Register (CEC_TRANSMIT_DATA) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | TX_BFR_ACC | | Read back internal buffer contents from 8Fh–9Eh. |
| | | 0 | No |
| | | 1 | Yes |
| 5 | TX_AUT_CALC | | Auto-Calculate TX_CNT and send. |
| | | 0 | No |
| | | 1 | Yes |
| 4 | TRA_CEC_CMD | | Send CEC Command and TX_CNT Operands. |
| | | 0 | No |
| | | 1 | Yes |
| 3-0 | TX_CNT | | Transmit Byte Count. Selects the number of CEC_TX_OPERAND bytes to send with the command. |
| | | 0 | No operands |
| | | 1h-Fh | 1 operand to 15 operands |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.15 CEC Capture ID0 Register (CEC_CA_7_0)

**Figure 13-220. CEC Capture ID0 Register (CEC_CA_7_0)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CEC_CAP_ID | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-232. CEC Capture ID0 Register (CEC_CA_7_0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CEC_CAP_ID | The CEC Capture ID register is separate from the CEC Initiator ID. It selects the received commands that will be captured in the receive FIFO and acknowledged. If the command destination matches one of the bits set in this register or is a Broadcast cycle; it will be captured. |

### 13.3.5.16 CEC Capture ID0 Register (CEC_CA_15_8)

**Figure 13-221. CEC Capture ID0 Register (CEC_CA_15_8)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CEC_CAP_ID | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-233. CEC Capture ID0 Register (CEC_CA_15_8) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CEC_CAP_ID | The CEC Capture ID register is separate from the CEC Initiator ID. It selects the received commands that will be captured in the receive FIFO and acknowledged. If the command destination matches one of the bits set in this register or is a Broadcast cycle; it will be captured. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.17 CEC Interrupt Enable Register 0 (CEC_INIT_ENABLE_0)

**Figure 13-222. CEC Interrupt Enable Register 0 (CEC_INIT_ENABLE_0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | CEC_INTR1_MASK5 | Reserved | | CEC_INTR1_MASK2 | CEC_INTR1_MASK1 | CEC_INTR1_MASK0 |
| R-0h | | R/W-0h | R-0h | | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-234. CEC Interrupt Enable Register 0 (CEC_INIT_ENABLE_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5 | CEC_INTR1_MASK5 | | Tx: Transmit Buffer Full/Empty Change event |
| | | 0 | Disable |
| | | 1 | Enable |
| 4-3 | Reserved | 0 | Reserved |
| 2 | CEC_INTR1_MASK2 | | Transmitter FIFO Empty Event |
| | | 0 | Disable |
| | | 1 | Enable |
| 1 | CEC_INTR1_MASK1 | | Receiver FIFO Not Empty Event |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | CEC_INTR1_MASK0 | | Command Being Received Event |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.18 CEC Interrupt Enable Register 1 (CEC_INIT_ENABLE_1)

**Figure 13-223. CEC Interrupt Enable Register 1 (CEC_INIT_ENABLE_1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CEC_INTR2_MASK3 | CEC_INTR2_MASK2 | CEC_INTR2_MASK1 | CEC_INTR2_MASK0 |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-235. CEC Interrupt Enable Register 1 (CEC_INIT_ENABLE_1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | CEC_INTR2_MASK3 | | Rx FIFO Overrun Error Event |
| | | 0 | Disable |
| | | 1 | Enable |
| 2 | CEC_INTR2_MASK2 | | Short Pulse Detected Event |
| | | 0 | Disable |
| | | 1 | Enable |
| 1 | CEC_INTR2_MASK1 | | Frame Retransmit Count Exceeded Event |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | CEC_INTR2_MASK0 | | Start Bit Irregularity Event |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.19 CEC Interrupt Status Register 0 (CEC_INIT_STATUS_0)

**Figure 13-224. CEC Interrupt Status Register 0 (CEC_INIT_STATUS_0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CEC_LSTAT | TFIF_BFULL | CEC_INTR1_STAT4 | Reserved | | CEC_INTR1_STAT2 | CEC_INTR1_STAT1 | CEC_INTR1_STAT0 |
| R-0h | R-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-236. CEC Interrupt Status Register 0 (CEC_INIT_STATUS_0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | CEC_LSTAT | | CEC line current state |
| | | 0 | Low |
| | | 1 | High |
| 6 | TFIF_BFULL | | Tx FIFO Transmit Buffer Full |
| | | 0 | No |
| | | 1 | Yes |
| 5 | CEC_INTR1_STAT4 | | Tx: Transmit Buffer Full/Empty Change Event Pending |
| | | 0 | No |
| | | 1 | Yes |
| 4-3 | Reserved | 0 | Reserved |
| 2 | CEC_INTR1_STAT2 | | Transmitter FIFO Empty Event Pending |
| | | 0 | No |
| | | 1 | Yes |
| 1 | CEC_INTR1_STAT1 | | Receiver FIFO Not Empty Event Pending |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | CEC_INTR1_STAT0 | | Command Being Received Event Pending |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.20 CEC Interrupt Status Register 1 (CEC_INIT_STATUS_1)

**Figure 13-225. CEC Interrupt Status Register 1 (CEC_INIT_STATUS_1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CEC_INTR2_STAT3 | CEC_INTR2_STAT2 | CEC_INTR2_STAT1 | CEC_INTR2_STAT0 |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

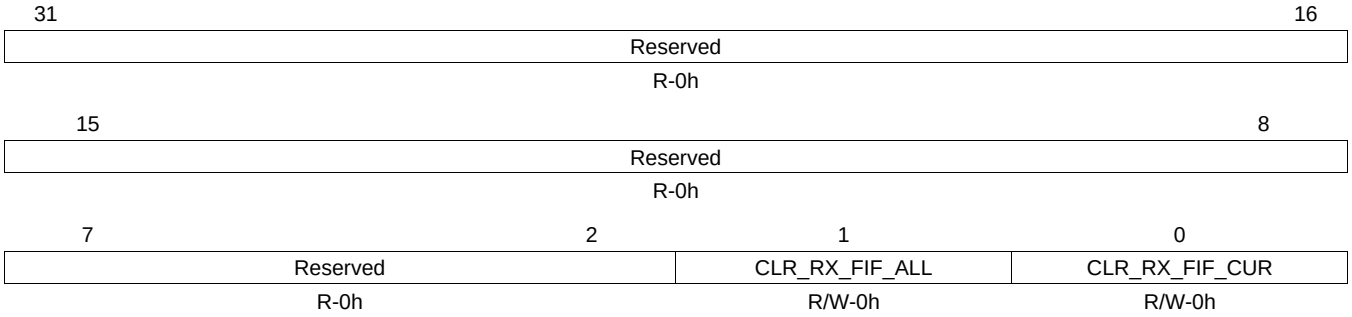
**Table 13-237. CEC_INIT_STATUS1 Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | CEC_INTR2_STAT3 | | Rx FIFO Overrun Error Event |
| | | 0 | Disable |
| | | 1 | Enable |
| 2 | CEC_INTR2_STAT2 | | Short Pulse Detected Event |
| | | 0 | Disable |
| | | 1 | Enable |
| 1 | CEC_INTR2_STAT1 | | Frame Retransmit Count Exceeded Event |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | CEC_INTR2_STAT0 | | Start Bit Irregularity Event |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.21 CEC RX Control Register (CEC_RX_CONTROL)

**Figure 13-226. CEC RX Control Register (CEC_RX_CONTROL)**



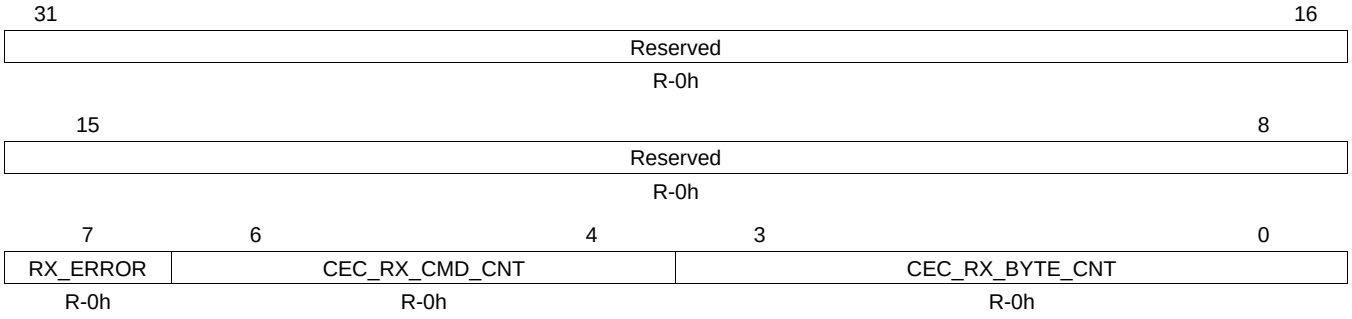

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-238. CEC RX Control Register (CEC_RX_CONTROL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1 | CLR_RX_FIF_ALL | | Clear all frames from Rx FIFO |
| | | 0 | No |
| | | 1 | Yes, self-resetting bit |
| 0 | CLR_RX_FIF_CUR | | Clear Current Frame from Rx FIFO |
| | | 0 | No |
| | | 1 | Yes, self-resetting bit |

### 13.3.5.22 CEC Rx Count Register (CEC_RX_COUNT)

**Figure 13-227. CEC Rx Count Register (CEC_RX_COUNT)**




LEGEND: R = Read only; -*n* = value after reset

**Table 13-239. CEC Rx Count Register (CEC_RX_COUNT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | RX_ERROR | | Error associated with this message |
| | | 0 | No |
| | | 1 | Yes |
| 6-4 | CEC_RX_CMD_CNT | 0 | CEC Receive FIFO Frame Count. Returns the number of frames awaiting reading in the FIFO; 0-3. |
| 3-0 | CEC_RX_BYTE_CNT | 0 | CEC Receive Byte Count. Returns the number of operands in the current frame. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.23 CEC Rx Command Header Register (CEC_RX_CMD_HEADER)

**Figure 13-228. CEC Rx Command Header Register (CEC_RX_CMD_HEADER)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CEC_RX_INIT | | | | CEC_RX_DEST | | | |
| R-0h | | | | R-0h | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 13-240. CEC Rx Command Header Register (CEC_RX_CMD_HEADER) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-4 | CEC_RX_INIT | CEC Initiator ID - identifies the initiator of the current frame. |
| 3-0 | CEC_RX_DEST | CEC Destination ID - identifies the intended target of the current frame. |

### 13.3.5.24 CEC Rx Command Register (CEC_RX_COMMAND)

**Figure 13-229. CEC Rx Command Register (CEC_RX_COMMAND)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | CEC_RX_COM | | | | | | | |
| R-0h | | | | | | | | R/W-0h | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-241. CEC Rx Command Register (CEC_RX_COMMAND) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CEC_RX_COM | CEC Rx Command |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.5.25 CEC Rx Operand Registers (CEC_RX_OPERAND_0-CEC_RX_OPERAND_14)

**Figure 13-230. CEC Rx Operand Registers (CEC_RX_OPERAND_0-CEC_RX_OPERAND_14)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CEC_RX_OP | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 13-242. CEC Rx Operand Registers (CEC_RX_OPERAND_0-CEC_RX_OPERAND_14) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7-0 | CEC_RX_OP | CEC Rx Operand |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.6 HDMI PHY Registers

Table 13-243lists the HDMI PHY registers.

**Table 13-243. HDMI PHY Registers**

| Address Offset | Acronym | Register Name |
|---|---|---|
| 04h | TMDS_CNTL2 | TMDS Control Register |
| 08h | TMDS_CNTL3 | TMDS Control Register |
| 0Ch | BIST_CNTL | BIST Control Register |
| 20h | TMDS_CNTL9 | TMDS Control Register |

#### 13.3.6.1 TMDS Control Register 2 (TMDS_CNTL2)

**Figure 13-231. TMDS Control Register 2 (TMDS_CNTL2)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | OE | TERM_EN | Reserved | | Clkdetect | Rsen |
| R-0h | | R/W-0h | R/W-0h | R-2h | | R-x | R-x |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; x = value is indeterminate

**Table 13-244. TMDS Control Register 2 (TMDS_CNTL2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5 | OE | | Output enable control |
| | | 0 | Output drivers of TMDS outputs are switched off |
| | | 1 | Output drivers are switched on |
| 4 | TERM_EN | | Source termination enable control |
| | | 0 | Bus termination is switched off |
| | | 1 | Bus termination is active; driver must also be enabled |
| 3-2 | Reserved | 2h | Reserved |
| 1 | Clkdetect | | Clock detector output |
| | | 0 | If clock < 2.5 MHz |
| | | 1 | If clock > 2.5 MHz |
| 0 | Rsen | | Receiver sense output |
| | | 0 | When receiver is disconnected |
| | | 1 | When receiver is connected |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.6.2 TMDS Control Register 3 (TMDS_CNTL3)

**Figure 13-232. TMDS Control Register 3 (TMDS_CNTL3)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | clkmult_ctl | | dpcolor_ctl | | pdb |
| R-5h | | | R/W-1h | | R/W-0h | | R/W-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-245. TMDS Control Register 3 (TMDS_CNTL3) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 5h | Reserved |
| 4-3 | clkmult_ctl | | Clock multiplication factor control. Use this setting when pixel repetition is needed. Output clock "m_clkout_dig" depends on the dpcolor_ctl and clkmult_ctl bits.<br>For example, if the input clock frequency p_clkin is 30 MHz:<br>dpcolor_ctl = 2h and clkmult_ctl = 3h, m_clkout_dig = 30 × 1.5 × 4 = 180 MHz;<br>dpcolor_ctl = 1h and clkmult_ctl = 3h, m_clkout_dig = 30 × 1.25 × 4 = 150 MHz;<br>dpcolor_ctl = 0 and clkmult_ctl = 3h, m_clkout_dig = 30 × 1.0 × 4 = 120 MHz. |
| | | 0 | 0.5× |
| | | 1h | 1× |
| | | 2h | 2× |
| | | 3h | 4× |
| 2-1 | dpcolor_ctl | | Deep color mode control |
| | | 0 | 8 bit/channel |
| | | 1h | 10 bit/channel |
| | | 2h | 12 bit/channel |
| | | 3h | Reserved |
| 0 | pdb | | Powerdown control |
| | | 0 | PHY will be power down |
| | | 1 | No power down |


Copyright © 2011–2016, Texas Instruments Incorporated

### 13.3.6.3 BIST Control Register (BIST_CNTL)

**Figure 13-233. BIST Control Register (BIST_CNTL)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | enc_byp | Reserved | |
| R-8h | | | | | R/W-0h | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-246. BIST Control Register (BIST_CNTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 8h | Reserved |
| 2 | enc_byp | | DVI encoder bypass. Always bypass the DVI encoder (set this bit to 1). |
| | | 0 | Data goes through the DVI encoder |
| | | 1 | Bypass the DVI encoder |
| 1-0 | Reserved | 0 | Reserved |

### 13.3.6.4 TMDS Control Register 9 (TMDS_CNTL9)

**Figure 13-234. TMDS Control Register 9 (TMDS_CNTL9)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ten_bit_bypass | Reserved | | | | | | |
| R/W-0h | R-0h | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 13-247. TMDS Control Register 9 (TMDS_CNTL9) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-8 | Reserved | Reserved |
| 7 | ten_bit_bypass | Always set this bit to 1. |
| 6-0 | Reserved | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

# ARM Interrupt Controller (AINTC)

This chapter discusses the ARM interrupt controller.

**Topic** **Page**

14.1 Introduction ..... 2864
14.2 Architecture ..... 2867
14.3 Basic Programming Model ..... 2870
14.4 Registers ..... 2879


Copyright © 2011–2016, Texas Instruments Incorporated

## 14.1 Introduction

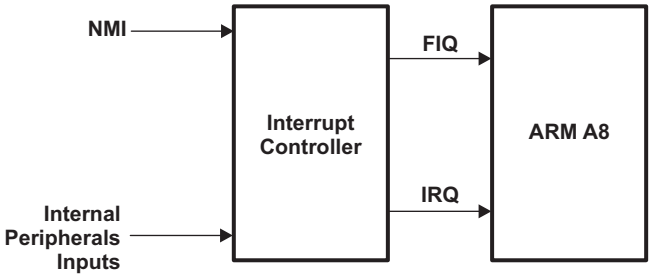
### 14.1.1 Overview

The Host ARM Interrupt Controller (AINTC) is responsible for prioritizing all service requests from the system peripherals and generating either nIRQ or nFIQ to the host. The type of the interrupt (nIRQ or nFIQ) and the priority of the interrupt inputs are programmable. However, FIQ can only be used on secure devices, and is not available on GP devices. It has the capability to handle up to 128 requests which can be steered/prioritized as ARM Cortex-A8 nFIQ or nIRQ interrupt requests. The general features of the AINTC are:

- Up to 128 level sensitive interrupts inputs
- Individual priority for each interrupt input
- Each interrupt can be steered to nFIQ or nIRQ
- Independent priority sorting for nFIQ and nIRQ

Figure 14-1 shows the internal interrupt scheme.

**Figure 14-1. Interrupt Controller**




### 14.1.2 Limitations

FIQ is only available on secure devices, and cannot be used on GP devices.

### 14.1.3 Functional Block Diagram

The interrupt controller processes incoming interrupts by masking and priority sorting to produce the interrupt signals for the processor to which it is attached. Figure 14-2 shows the top-level view of interrupt processing.

### 14.1.4 Environment

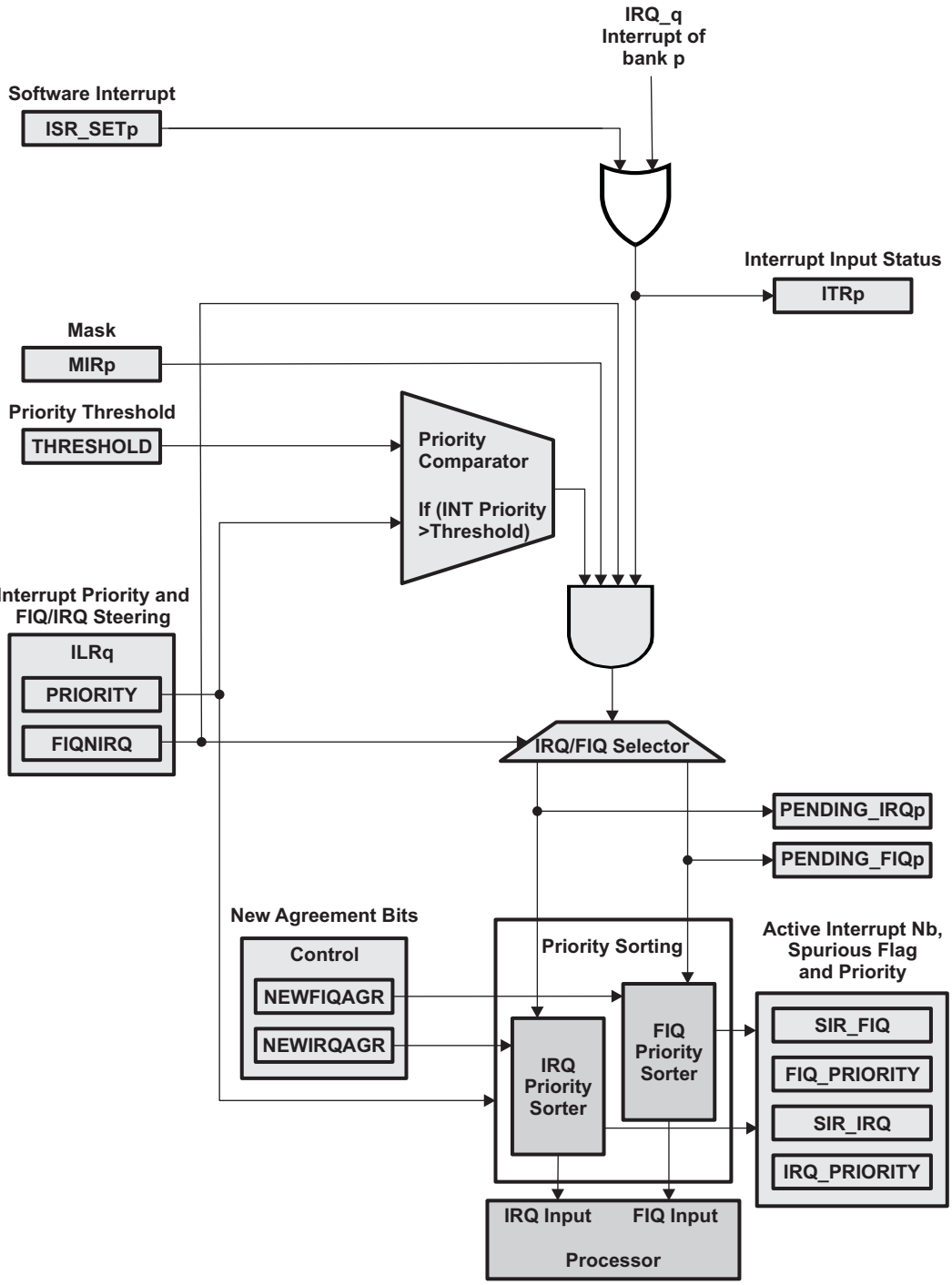
The sources of interrupt for AINTC are:

- External interrupts through NMI pin. Note that interrupts from external peripherals of SOC need to be routed through GPIO.
  - NMI is treated like other interrupts and can be masked. In general, software should configure the NMI input as the highest priority interrupt
- Interrupt from peripherals inside the SOC and ARM Cortex-A8.

All the unmasked interrupts can generate a wake up to the interrupt controller.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 14-2. Interrupt Controller Block Diagram**





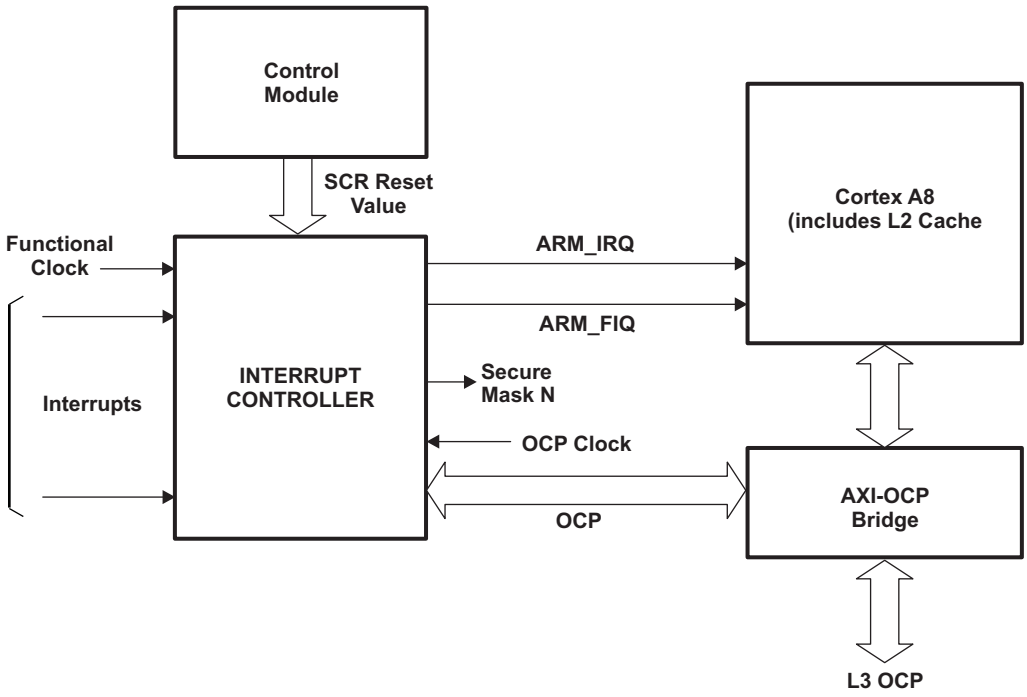
Copyright © 2011–2016, Texas Instruments Incorporated

### 14.1.5 ARM A8—Interrupt Controller Integration

The interrupt controller (INTC) is the interface between the many incoming interrupts and the two interrupt inputs of the ARM Cortex-A8. The two interrupts inputs to ARM are FIQ and IRQ. Figure 14-3 shows the integration of the interrupt controller in the ARM Cortex-A8 subsystem.

The ARM Cortex-A8 subsystem INTC is directly connected to the ARM Cortex-A8 by an ARM peripheral port. Consequently, the ARM Cortex-A8 subsystem INTC is accessible and visible only by the ARM Cortex-A8.

**Figure 14-3. ARM A8 Subsystem INTC Integration**





Copyright © 2011–2016, Texas Instruments Incorporated

## 14.2 Architecture

This section discusses the architecture of the interrupt controller.

### 14.2.1 Clocking and Reset

#### 14.2.1.1 ARM A8 INTC Clocks

The interrupt controller runs at half the rate of the ARM Cortex-A8 functional clock. The interface clock, used for register access, runs at the rate of the interconnect bus clock (equal to the rate of the ARM Cortex-A8 interface clock).

#### 14.2.1.2 Hardware and Software Reset

Table 14-1 lists the ARM Cortex-A8 subsystem INTC resets.

**Table 14-1. Interrupt Controller Resets**

| Type | Name | Source | Activation |
|---|---|---|---|
| Hardware | ARM Cortex-A8 top-level Reset | PRCM | Active low |
| Software | SOFTRESET | INTCPS_SYSCONFIG[1]SOFTRESET bit, write 1 to clear. Read returns 0. | Active high |

### 14.2.2 Interrupt Request Lines

Table 14-2 lists the incoming and outgoing interrupt lines of the ARM Cortex-A8 INTC.

For mapping of interrupt from various modules to the ARM Cortex-A8 INTC, refer to your device-specific data manual. Check availability of modules and features in a particular device. Ensure that interrupts of unavailable modules and features are masked in ARM Cortex-A8 subsystem.

**Table 14-2. Interrupt Controller Interrupt Inputs and Outputs**

| Type | Number | Name | Mapping | Comment |
|---|---|---|---|---|
| Interrupt request inputs | Up to 128 | Refer to data manual | Refer to data manual | Inputs to ARM Cortex-A8 INTC module, source from various modules. |
| Interrupt request outputs | 2 | INTC_FIQ<br>INTC_IRQ | INTC_FIQ<br>INTC_IRQ | FIQ is fast interrupt to ARM<br>IRQ is normal input to ARM Cortex-A8 |

### 14.2.3 Interrupt Processing

#### 14.2.3.1 Input Selection

The INTC supports only level-sensitive incoming interrupt detection. A peripheral asserting an interrupt maintains it until software has handled the interrupt and instructed the peripheral to deassert the interrupt. A software interrupt is generated if the corresponding bit in the INTCPS_ISR_SETn register is set (register bank number: n = [0,1,2,3] for the MPU subsystem INTC, 128 incoming interrupt lines are supported). The software interrupt clears when the corresponding bit in the INTCPS_ISR_CLEARn register is written. Typical use of this feature is software debugging.


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.2.3.2 Masking

#### *14.2.3.2.1 Individual Masking*

Detection of interrupts on each incoming interrupt line can be enabled or disabled independently by the INTCPS_MIRn interrupt mask register. In response to an unmasked incoming interrupt, the INTC can generate one of two types of interrupt requests to the processor:

- IRQ: normal interrupt request
- FIQ: fast interrupt request

The type of interrupt request is determined by the INTCPS_ILRm[0] FIQNIRQ bit (m= [0,127]). The current incoming interrupt status before masking is readable from the INTCPS_ITRn register. After masking and IRQ/FIQ selection, and before priority sorting is done, the interrupt status is readable from the INTCPS_PENDING_IRQn and INTCPS_PENDING_FIQn registers.

#### *14.2.3.2.2 Priority Masking*

To enable faster processing of high-priority interrupts, a programmable priority masking threshold is provided (the INTCPS_THRESHOLD[7:0] PRIORITYTHRESHOLD field). This priority threshold allows preemption by higher priority interrupts; all interrupts of lower or equal priority than the threshold are masked. However, priority 0 can never be masked by this threshold; a priority threshold of 0 is treated the same way as priority 1. PRIORITY and PRIORITYTHRESHOLD fields values can be set between 0 and 7Fh; 0 is the highest priority and 7Fh is the lowest priority. When priority masking is not necessary, a priority threshold value of FFh disables the priority threshold mechanism. This value is also the reset default for backward compatibility with previous versions of the INTC.

### 14.2.3.3 Priority Sorting

A priority level (0 being the highest) is assigned to each incoming interrupt line. Both the priority level and the interrupt request type are configured by the INTCPS_ILRm register. If more than one incoming interrupt with the same priority level and interrupt request type occur simultaneously, the highest-numbered interrupt is serviced first. When one or more unmasked incoming interrupts are detected, the INTC separates between IRQ and FIQ using the corresponding INTCPS_ILRm[0] FIQNIRQ bit. The result is placed in INTCPS_PENDING_IRQn or INTCPS_PENDING_FIQn. If no other interrupts are currently being processed, INTC asserts IRQ/FIQ and starts the priority computation. Priority sorting for IRQ and FIQ can execute in parallel. Each IRQ/FIQ priority sorter determines the highest priority interrupt number. Each priority number is placed in the corresponding INTCPS_SIR_IRQ[6:0] ACTIVEIRQ field or INTCPS_SIR_FIQ[6:0] ACTIVEFIQ field. The value is preserved until the corresponding INTCPS_CONTROL NEWIRQAGR or NEWFIQAGR bit is set. Once the interrupting peripheral device has been serviced and the incoming interrupt deasserted, you must write to the appropriate NEWIRQAGR or NEWFIQAGR bit to indicate to the INTC the interrupt has been handled. If there are any pending unmasked incoming interrupts for this interrupt request type, the INTC restarts the appropriate priority sorter; otherwise, the IRQ or FIQ interrupt line is deasserted.


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.2.4 Module Power Saving

The INTC provides an auto-idle function in its three clock domains:

- Interface clock
- Functional clock
- Synchronizer clock

The interface clock auto-idle power-saving mode is enabled if the INTCPS_SYSCONFIG[0] AUTOIDLE bit is set to 1. When this mode is enabled and there is no activity on the bus interface, the interface clock is disabled internally to the module, thus reducing power consumption. When there is new activity on the bus interface, the interface clock restarts without any latency penalty. After reset, this mode is disabled, by default.

The functional clock auto-idle power-saving mode is enabled if the INTCPS_IDLE[0] FUNCIDLE bit is cleared to 0. When this mode is enabled and there is no active interrupt (IRQ or FIQ interrupt being processed or generated) or no pending incoming interrupt, the functional clock is disabled internally to the module, thus reducing power consumption. When a new unmasked incoming interrupt is detected, the functional clock restarts and the INTC processes the interrupt. If this mode is disabled, the interrupt latency is reduced by one cycle. After reset, this mode is enabled, by default.

The synchronizer clock allows external asynchronous interrupts to be resynchronized before they are masked. The synchronizer input clock has an auto-idle power-saving mode enabled if the INTCPS_IDLE[1] TURBO bit is set to 1. If the auto-idle mode is enabled, the standby power is reduced, but the IRQ or FIQ interrupt latency increases from four to six functional clock cycles. This feature can be enabled dynamically according to the requirements of the device. After reset, this mode is disabled, by default.

### 14.2.5 Error Handling

The following accesses will cause error:

- Privilege violation (attempt to access PROTECTION register in user mode or any register in user mode if Protection bit is set).
- Unsupported commands.

The following will NOT cause any error response:

- Access to a non-decoded address.
- Write to a read-only register.

### 14.2.6 Interrupt Latency

The IRQ/FIQ interrupt generation takes four INTC functional clock cycles (plus or minus one cycle) if the INTCPS_IDLE[1] TURBO bit is cleared to 0. If the TURBO bit is set to 1, the interrupt generation takes six cycles, but power consumption is reduced while waiting for an interrupt. These latencies can be reduced by one cycle by disabling functional clock auto-idle (INTCPS_IDLE[0] FUNCIDLE bit set to 1), but power consumption is increased, so the benefit is minimal. To minimize interrupt latency when an unmasked interrupt occurs, the IRQ or FIQ interrupt is generated before priority sorting completion. The priority sorting takes 10 functional clock cycles, which is less than the minimum number of cycles required for the MPU to switch to the interrupt context after reception of the IRQ or FIQ event.

Any read of the INTCPS_SIR_IRQ or INTCPS_SIR_FIQ register during the priority sorting process stalls until priority sorting is complete and the relevant register is updated. However, the delay between the interrupt request being generated and the interrupt service routine being executed is such that priority sorting always completes before the INTCPS_SIR_IRQ or INTCPS_SIR_FIQ register is read.


Copyright © 2011–2016, Texas Instruments Incorporated

## 14.3 Basic Programming Model

### 14.3.1 Initialization Sequence

1. Program the INTCPS_SYSCONFIG register: If necessary, enable the interface clock autogating by setting the AUTOIDLE bit.
2. Program the INTCPS_IDLE register: If necessary, disable functional clock autogating or enable synchronizer autogating by setting the FUNCIDLE bit or TURBO bit accordingly.
3. Program the INTCPS_ILRm register for each interrupt line: Assign a priority level and set the FIQNIRQ bit for an FIQ interrupt (by default, interrupts are mapped to IRQ and priority is 0 [highest]).
4. Program the INTCPS_MIRn register: Enable interrupts (by default, all interrupt lines are masked).

   To program the INTCPS_MIRn register, the INTCPS_MIR_SETn and INTCPS_MIR_CLEARn registers are provided to facilitate the masking, even if it is possible for backward-compatibility to write directly to the INTCPS_MIRn register.

### 14.3.2 INTC Processing Sequence

After the INTCPS_MIRn and INTCPS_ILRm registers are configured to enable and assign priorities to incoming interrupts, the interrupt is processed as explained in the following subsections. IRQ and FIQ processing sequences are quite similar, the differences for the FIQ sequence are shown after a '/' character in the code below.

1. One or more unmasked incoming interrupts (M_IRQ_n signals) are received and IRQ or FIQ outputs (IRQ/FIQ) are not currently asserted.
2. If the INTCPS_ILRm[0] FIQNIRQ bit is cleared to 0, the MPU_INTC_IRQ output signal is generated. If the FIQNIRQ bit is set to 1, the MPU_INTC_FIQ output signal is generated.
3. The INTC performs the priority sorting and updates the INTCPS_SIR_IRQ[6:0] ACTIVEIRQ /INTCPS_SIR_FIQ[6:0] ACTIVEFIQ field with the current interrupt number.
4. During priority sorting, if the IRQ/FIQ is enabled at the host processor side, the host processor automatically saves the current context and executes the ISR as follows.

   The ARM host processor automatically performs the following actions in pseudo code:

```
LR = PC + 4                    /* return link */
SPSR = CPSR                    /* Save CPSR before execution */
CPSR[5] = 0                    /* Execute in ARM state */
CPSR[7] = 1                    /* Disable IRQ */
CPSR[8] = 1                    /* Disable Imprecise Data Aborts */
CPSR[9] = CP15_reg1_EEbit      /* Endianness on exception entry */
if interrupt == IRQ then
CPSR[4:0] = 0b10010            /* Enter IRQ mode */
if high vectors configured then
PC = 0xFFFF0018
else
PC = 0x00000018                /* execute interrupt vector */
else if interrupt == FIQ then
CPSR[4:0] = 0b10001            /* Enter FIQ mode */
CPSR[6] = 1                    /* Disable FIQ */
if high vectors configured then
PC = 0xFFFF001C
else
PC = 0x0000001C                /* execute interrupt vector */
endif
```


Copyright © 2011–2016, Texas Instruments Incorporated

5. The ISR saves the remaining context, identifies the interrupt source by reading the ACTIVEIRQ/ACTIVEFIQ field, and jumps to the relevant subroutine handler as follows:

> **CAUTION**
>
> The code in steps 5 and 7 is an assembly code compatible with ARM architecture V6 and V7. This code is developed for the Texas Instruments Code Composer Studio tool set. It is a draft version, only tested on an emulated environment.

```
 ;INTCPS_SIR_IRQ/INTCPS_SIR_FIQ register address
INTCPS_SIR_IRQ_ADDR/INTCPS_SIR_FIQ_ADDR .word 0x48200040/0x48200044
 ; ACTIVEIRQ bit field mask to get only the bit field
ACTIVEIRQ_MASK .equ 0x7F
_IRQ_ISR/_FIQ_ISR:
 ; Save the critical context
STMFD SP!, {R0-R12, LR} ; Save working registers and the Link register
MRS R11, SPSR ; Save the SPSR into R11
 ; Get the number of the highest priority active IRQ/FIQ
LDR R10, INTCPS_SIR_IRQ_ADDR/INTCPS_SIR_FIQ_ADDR
LDR R10, [R10] ; Get the INTCPS_SIR_IRQ/INTCPS_SIR_FIQ register
AND R10, R10, #ACTIVEIRQ_MASK ; Apply the mask to get the active IRQ number
 ; Jump to relevant subroutine handler
LDR PC, [PC, R10, lsl #2] ; PC base address points this instruction + 8
NOP ; To index the table by the PC
 ; Table of handler start addresses
.word IRQ0handler ;For IRQ0 of BANK0
.word IRQ1handler
.word IRQ2handler
```

6. The subroutine handler executes code specific to the peripheral generating the interrupt by handling the event and deasserting the interrupt condition at the peripheral side.

```
 ; IRQ0 subroutine
IRQ0handler:
 ; Save working registers
STMFD SP!, {R0-R1}
 ; Now read-modify-write the peripheral module status register
 ; to de-assert the M_IRQ_0 interrupt signal
 ; De-Assert the peripheral interrupt
MOV R0, #0x7 ; Mask for 3 flags
LDR R1, MODULE0_STATUS_REG_ADDR ; Get the address of the module Status Register
STR R0, [R1] ; Clear the 3 flags
 ; Restore working registers LDMFD SP!, {R0-R1}
 ; Jump to the end part of the ISR
B IRQ_ISR_end/FIQ_ISR_end
```


Copyright © 2011–2016, Texas Instruments Incorporated

7. After the return of the subroutine, the ISR sets the NEWIRQAGR/NEWFIQAGR bit to enable the processing of subsequent pending IRQs/FIQs and to restore ARM context in the following code. Because the writes are posted on an Interconnect bus, to be sure that the preceding writes are done before enabling IRQs/FIQs, a Data Synchronization Barrier is used. This operation ensure that the IRQ/FIQ line is de-asserted before IRQ/FIQ enabling. After that, the INTC processes any other pending interrupts or deasserts the IRQ/FIQ signal if there is no interrupt.

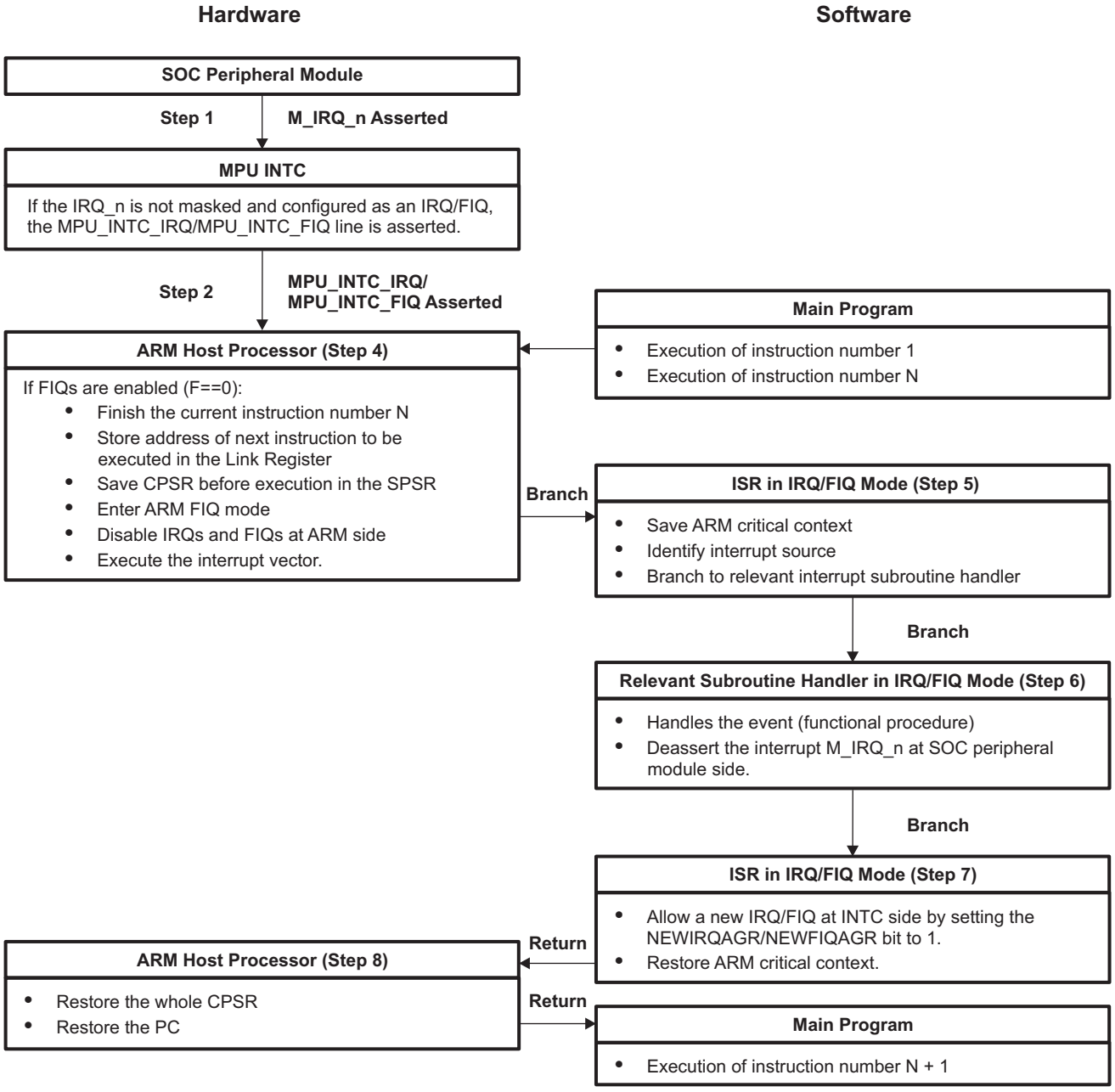
```
 ; INTCPS_CONTROL register address
INTCPS_CONTROL_ADDR .word 0x48200048;
NEWIRQAGR/NEWFIQAGR bit mask to set only the NEWIRQAGR/NEWFIQAGR bit
NEWIRQAGR_MASK/NEWFIQAGR_MASK .equ 0x01/0x02
IRQ_ISR_end/FIQ_ISR_end:
 ; Allow new IRQs/FIQs at INTC side
 ; The INTCPS_CONTROL register is a write only register so no need to write back others bits
MOV R0, #NEWIRQAGR_MASK/NEWFIQAGR_MASK ; Get the NEWIRQAGR/NEWFIQAGR bit position
LDR R1, INTCPS_CONTROL_ADDR
STR R0, [R1] ; Write the NEWIRQAGR/NEWFIQAGR bit to allow new IRQs/FIQ
 ; Data Synchronization Barrier MOV R0, #0
MCR P15, #0, R0, C7, C10, #4
 ; restore critical context
MSR SPSR, R11 ; Restore the SPSR from R11
LDMFD SP!, {R0-R12, LR} ; Restore working registers and  Link register
 ; Return after handling the interrupt
SUBS PC, LR, #4
8. After the ISR return, the ARM automatically restores its context as follows:
CPSR = SPSR
PC = LR
```

Figure 14-4 shows the IRQ/FIQ processing sequence from the originating device peripheral module to the main program interruption.

The priority sorting mechanism is frozen during an interrupt processing sequence. If an interrupt condition occurs during this time, the interrupt is not lost. It is sorted when the NEWIRQAGR/NEWFIQAGR bit is set (priority sorting is reactivated).


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 14-4. IRQ/FIQ Processing Sequence**





Copyright © 2011–2016, Texas Instruments Incorporated

### 14.3.3 INTC Preemptive Processing Sequence

Preemptive interrupts, also called nested interrupts, can reduce the latencies for higher priority interrupts. A preemptive ISR can be suspended by a higher priority interrupt. Thus, the higher priority interrupt can be served immediately. Nested interrupts must be used carefully to avoid using corrupted data. Programmers must save corruptible registers and enable IRQ or FIQ at ARM side. IRQ and FIQ processing sequences are quite similar, the differences for the FIQ sequence are shown after a '/' character in the code below.

To enable IRQ/FIQ preemption by higher priority IRQs/FIQs, programers can follow this procedure to write the ISR.

At the beginning of an IRQ/FIQ ISR:

1. Save the ARM critical context registers.
2. Save the INTCPS_THRESHOLD PRIORITYTHRESHOLD field before modifying it.
3. Read the active interrupt priority in the INTCPS_IRQ_PRIORITY IRQPRIORITY/INTCPS_FIQ_PRIORITY FIQPRIORITY field and write it to the PRIORITYTHRESHOLD(1) field.
4. Read the active interrupt number in the INTCPS_SIR_IRQ[6:0] ACTIVEIRQ/INTCPS_SIR_FIQ[6:0] ACTIVEFIQ field to identify the interrupt source.
5. Write 1 to the appropriate INTCPS_CONTROL NEWIRQAGR and (2) NEWFIQAGR bit while an interrupt is still processing to allow only higher priority interrupts to preempt.
6. Because the writes are posted on an Interconnect bus, to be sure that the preceding writes are done before enabling IRQs/FIQs, a Data Synchronization Barrier is used. This operation ensure that the IRQ line is de-asserted before IRQ/FIQ enabling.
7. Enable IRQ/FIQ at ARM side.
8. Jump to the relevant subroutine handler.


Copyright © 2011–2016, Texas Instruments Incorporated

The following sample code shows the previous steps:

**CAUTION**

The following code is an assembly code compatible with ARM architecture V6 and V7. This code is developed for the Texas Instruments Code Composer Studio tool set. It is a draft version, only tested on an emulated environment.

```
 ; bit field mask to get only the bit field
ACTIVEPRIO_MASK .equ 0x7F
_IRQ_ISR:
 ; Step 1 : Save the critical context
STMFD SP!, {R0-R12, LR} ; Save working registers
MRS R11, SPSR ; Save the SPSR into R11
 ; Step 2 : Save the INTCPS_THRESHOLD register into R12
LDR R0, INTCPS_THRESHOLD_ADDR
LDR R12, [R0]
(1) The priority-
threshold mechanism is enabled automatically when writing a priority in the range of 0x00 to
0x7F. Writing a value of 0xFF (reset default) disables the priority-
threshold mechanism. Values between 0x3F and 0xFF must not be used. When the hardware-
priority threshold is in use, the priorities of interrupts selected as FIQ or IRQ become linked
otherwise, they are independent. When they are linked, all FIQ priorities must be set higher than
all IRQ priorities to maintain the relative priority of FIQ over IRQ.
(2) When handling FIQs using the priority-
threshold mechanism, both NEWFIQAGR and NEWIRQAGR bits must be written at the same time to ensure
that the new priority threshold is applied while an IRQ sort is in progress. This IRQ will not
have been seen by the ARM, as it will have been masked on entry to the FIQ ISR. However, the
source of the IRQ remains active and it is finally processed when the priority threshold falls to
a priority sufficiently low to allow it to be processed. The precaution of writing to New FIQ
Agreement is not required during an IRQ ISR, as FIQ sorting is not affected (provided all FIQ
priorities are higher than all IRQ priorities).
 ; Step 3 : Get the priority of the highest priority active IRQ
LDR R1, INTCPS_IRQ_PRIORITY_ADDR/INTCPS_FIQ_PRIORITY_ADDR
LDR R1, [R1] ; Get the INTCPS_IRQ_PRIORITY/INTCPS_FIQ_PRIORITY register
AND R1, R1, #ACTIVEPRIO_MASK ; Apply the mask to get the priority of the IRQ
STR R1, [R0] ; Write it to the INTCPS_THRESHOLD register
 ; Step 4 : Get the number of the highest priority active IRQ
LDR R10, INTCPS_SIR_IRQ_ADDR/INTCPS_SIR_FIQ_ADDR
LDR R10, [R10] ; Get the INTCPS_SIR_IRQ/INTCPS_SIR_FIQ register
AND R10, R10, #ACTIVEIRQ_MASK ; Apply the mask to get the active IRQ number
 ; Step 5 : Allow new IRQs and FIQs at INTC side
MOV R0, #0x1/0x3 ; Get the NEWIRQAGR and NEWFIQAGR bit position
LDR R1, INTCPS_CONTROL_ADDR
STR R0, [R1] ; Write the NEWIRQAGR and NEWFIQAGR bit
 ; Step 6 : Data Synchronization Barrier
MOV R0, #0 MCR P15, #0, R0, C7, C10, #4
 ; Step 7 : Read-modify-write the CPSR to enable IRQs/FIQs at ARM side
MRS R0, CPSR ; Read the status register
BIC R0, R0, #0x80/0x40 ; Clear the I/F bit
MSR CPSR, R0 ; Write it back to enable IRQs
 ; Step 8 : Jump to relevant subroutine handler
LDR PC, [PC, R10, lsl #2] ; PC base address points this instruction + 8
NOP ; To index the table by the PC
 ; Table of handler start addresses
.word IRQ0handler ;IRQ0 BANK0
.word IRQ1handler
.word IRQ2handler
```


Copyright © 2011–2016, Texas Instruments Incorporated

After the return of the relevant IRQ/FIQ subroutine handle:

1. Disable IRQs/FIQs at ARM side.
2. Restore the INTCPS_THRESHOLD PRIORITYTHRESHOLD field.
3. Restore the ARM critical context registers.

The following sample code shows the three previous steps:

**CAUTION**

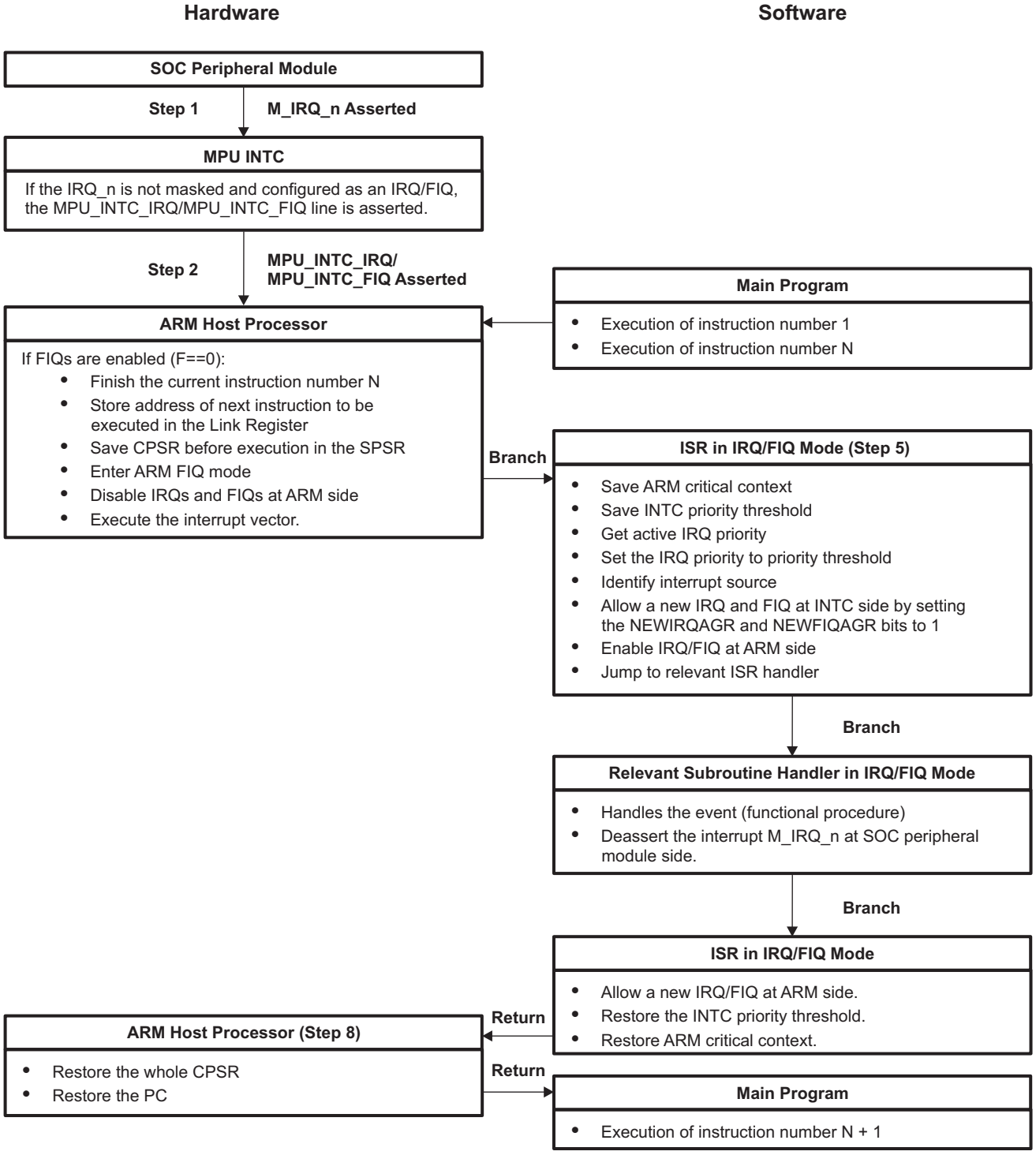
The following code is an assembly code compatible with ARM architecture V6 and V7. This code is developed for the Texas Instruments Code Composer Studio tool set. It is a draft version, only tested on an emulated environment.

```
IRQ_ISR_end:
 ; Step 1 : Read-modify-write the CPSR to disable IRQs/FIQs at ARM side
MRS R0, CPSR ; Read the CPSR
ORR R0, R0, #0x80/0x40 ; Set the I/F bit
MSR CPSR, R0 ; Write it back to disable IRQs
 ; Step 2 : Restore the INTCPS_THRESHOLD register from R12
LDR R0, INTCPS_THRESHOLD_ADDR
STR R12, [R0]
 ; Step 3 : Restore critical context
MSR SPSR, R11 ; Restore the SPSR from R11
LDMFD SP!, {R0-R12, LR} ; Restore working registers and Link register
 ; Return after handling the interrupt
SUBS PC, LR, #4
```

Figure 14-5 shows the nested IRQ/FIQ processing sequence from the originating device peripheral module to the main program interruption.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 14-5. Nested IRQ/FIQ Processing Sequence**





Copyright © 2011–2016, Texas Instruments Incorporated

### 14.3.4 Interrupt Preemption

If wanting to enable pre-emption by higher priority interrupts, the ISR should read the active interrupt priority and write it to the priority threshold register. Writing a '1' to the appropriate NEW_IRQ_AGR or NEW_FIQ_AGR bits of the CONTROL register while still processing the interrupt will now allow only higher priority interrupts to pre-empt.

For each level of pre-emption, the programmer must save the threshold value before modifying it and restore it at the end of that ISR level.

The priority threshold mechanism is enabled automatically when writing a priority in the range of 0 to 7Fh as will be read from the IRQ_PRIORITY and FIQ_PRIORITY registers. Writing a value of FFh (reset default) disables the priority threshold mechanism.

When the hardware priority threshold is in use the priorities of interrupts selected as FIQ or IRQ become linked, otherwise they are independent. When linked, it is required that all FIQ priorities be set higher than all IRQ priorities to maintain the relative priority of FIQ over IRQ.

When handling FIQs using the priority threshold mechanism, it is required to write both New FIQ Agreement and New IRQ Agreement bits at the same time to cover the case that the new priority threshold is applied while an IRQ sorting is in progress. This IRQ will not have been seen by the ARM as it will have been masked on entry to the FIQ ISR. However, the source of the IRQ will remain active and it will be finally processed when the priority threshold falls to a low enough priority. The precaution of writing to New FIQ Agreement (as well as New IRQ Agreement) is not required during an IRQ ISR as FIQ sorting will not be affected (provided all FIQ priorities are higher than all IRQ priorities).

### 14.3.5 ARM A8 INTC Spurious Interrupt Handling

The spurious flag indicates whether the result of the sorting (a window of 10 INTC functional clock cycles after the interrupt assertion) is invalid. The sorting is invalid if:

- The interrupt that triggered the sorting is no longer active during the sorting.
- A change in the mask has affected the result during the sorting time.

As a result, the values in the INTCPS_MIRn, INTCPS_ILRm, or INTCPS_MIR_SETn registers must not be changed while the corresponding interrupt is asserted. Only the active interrupt input that triggered the sort can be masked before it turn on the sort. If these registers are changed within the 10-cycle window after the interrupt assertion. The resulting values of the following registers become invalid:

- INTCPS_SIR_IRQ
- INTCPS_SIR_FIQ
- INTCPS_IRQ_PRIORITY
- INTCPS_FIQ_PRIORITY

This condition is detected for both IRQ and FIQ, and the invalid status is flagged across the SPURIOUSIRQFLAG (see NOTE 1) and SPURIOUSFIQFLAG (see NOTE 2) bit fields in the SIR and PRIORITY registers. A 0 indicates valid and a 1 indicates invalid interrupt number and priority. The invalid indication can be tested in software as a false register value.

**NOTE:**

1. The INTCPS_SIR_IRQ[31:7] SPURIOUSIRQFLAG bit field is a copy of the INTCPS_IRQ_PRIORITY[31:7] SPURIOUSIRQFLAG bit field.
2. The INTCPS_SIR_FIQ[31:7] SPURIOUSFIQFLAG bit field is a copy of the INTCPS_FIQ_PRIORITY[31:7] SPURIOUSFIQFLAG bit field.


Copyright © 2011–2016, Texas Instruments Incorporated

## 14.4 Registers

If the INTCPS_PROTECTION[0] PROTECTION bit is set, access to the INTC registers is restricted to the supervisor mode. Access to the INTCPS_PROTECTION register is always restricted to privileged mode.

Each register from INTCPS_ITRn to INTCPS_PENDING_FIQn contains 32 bits, 1 bit for each interrupt (in ascending order: bit 0 of the INTCPS_ITR0 register applies to interrupt line 0; bit 0 of the INTCPS_ITR1 register applies to interrupt line 32).

**CAUTION**

The INTC registers are limited to 32-bit and 16-bit data accesses; 8-bit data access is not allowed and can corrupt the register content.

Table 14-3 lists the memory-mapped registers for the INTC.

**Table 14-3. Interrupt Controller (INTC) Registers**

| Address Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 0h | INTCPS_REVISION | Revision Identification Register | Section 14.4.1 |
| 10h | INTCPS_SYSCONFIG | System Configuration Register | Section 14.4.2 |
| 14h | INTCPS_SYSSTATUS | System Status Register | Section 14.4.3 |
| 40h | INTCPS_SIR_IRQ | Spurious IRQ Flag Register | Section 14.4.4 |
| 44h | INTCPS_SIR_FIQ | Spurious FIQ Flag Register | Section 14.4.5 |
| 48h | INTCPS_CONTROL | Control Register | Section 14.4.6 |
| 4Ch | INTCPS_PROTECTION | Protection Mode Register | Section 14.4.7 |
| 50h | INTCPS_IDLE | Idle Mode Register | Section 14.4.8 |
| 60h | INTCPS_IRQ_PRIORITY | IRQ Priority Register | Section 14.4.9 |
| 64h | INTCPS_FIQ_PRIORITY | FIQ Priority Register | Section 14.4.10 |
| 68h | INTCPS_THRESHOLD | Priority Threshold Register | Section 14.4.11 |
| 80h + (20h × n)(1) | INTCPS_ITR0-3 | Interrupt Status Register | Section 14.4.12 |
| 84h + (20h × n)(1) | INTCPS_MIR0-3 | Interrupt Mask Register | Section 14.4.13 |
| 88h + (20h × n)(1) | INTCPS_MIR_CLEAR0-3 | Interrupt Mask Clear Register | Section 14.4.14 |
| 8Ch + (20h × n)(1) | INTCPS_MIR_SET0-3 | Interrupt Mask Set Register | Section 14.4.15 |
| 90h + (20h × n)(1) | INTCPS_ISR_SET0-3 | Software Interrupt Set Register | Section 14.4.16 |
| 94h + (20h × n)(1) | INTCPS_ISR_CLEAR0-3 | Software Interrupt Clear Register | Section 14.4.17 |
| 98h + (20h × n)(1) | INTCPS_PENDING_IRQ0-3 | IRQ Status Register | Section 14.4.18 |
| 9Ch + (20h × n)(1) | INTCPS_PENDING_FIQ0-3 | FIQ Status Register | Section 14.4.19 |
| 100h + (4h × m)(1) | INTCPS_ILR0-127 | Interrupt Priority Register | Section 14.4.20 |

(1) n = 0 to 3, m = 0 to 127


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.4.1 INTCPS_REVISION Register

This register contains the IP revision code.

**Figure 14-6. INTCPS_REVISION Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | MAJOR | | MINOR | |
| R-0 | | R-5h | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-4. INTCPS_REVISION Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reads return 0 |
| 7-4 | MAJOR | 5h | Major Revision. |
| 3-0 | MINOR | 0 | Minor Revision. |

### 14.4.2 INTCPS_SYSCONFIG Register

This register controls various parameters of the module interface.

**Figure 14-7. INTCPS_SYSCONFIG Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | SOFTRESET | AUTOIDLE |
| R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-5. INTCPS_SYSCONFIG Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Write 0s for future compatibility. Read returns reset value. |
| 1 | SOFTRESET | | Software reset. Set this bit to trigger a module reset. The bit is automatically reset by the hardware. Read returns 0. |
| | | W0 | No effect. |
| | | W1 | The module is reset. |
| 0 | AUTOIDLE | | Internal interface clock gating strategy. |
| | | 0 | Interface clock is free-running. |
| | | 1 | Automatic interface clock gating strategy is applied, based on the interface bus activity. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.4.3 INTCPS_SYSSTATUS Register

This register provides status information about the module.

**Figure 14-8. INTCPS_SYSSTATUS Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | RESETDONE |
| R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-6. INTCPS_SYSSTATUS Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Read returns reset value. |
| 0 | RESETDONE | | Internal reset monitoring. |
| | | 0 | Internal module reset is ongoing. |
| | | 1 | Reset complete. |

### 14.4.4 INTCPS_SIR_IRQ Register

This register supplies the currently active IRQ interrupt number.

**Figure 14-9. INTCPS_SIR_IRQ Register**

| 31 | 16 |
|---|---|
| SPURIOUSIRQFLAG | |
| R-1FF FFFFh | |

| 15 | 7 | 6 | 0 |
|---|---|---|---|
| SPURIOUSIRQFLAG | | ACTIVEIRQ | |
| R-1FF FFFFh | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-7. INTCPS_SIR_IRQ Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | SPURIOUSIRQFLAG | 0-1FF FFFFh | Spurious IRQ flag |
| 6-0 | ACTIVEIRQ | 0-7Fh | Active IRQ number |


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.4.5 INTCPS_SIR_FIQ Register

This register supplies the currently active FIQ interrupt number.

**Figure 14-10. INTCPS_SIR_FIQ Register**

| 31 | 16 |
|---|---|
| SPURIOUSFIQFLAG | |
| R-1FF FFFFh | |

| 15 | 7 | 6 | 0 |
|---|---|---|---|
| SPURIOUSFIQFLAG | | ACTIVEFIQ | |
| R-1FF FFFFh | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-8. INTCPS_SIR_FIQ Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | SPURIOUSFIQFLAG | 0-1FF FFFFh | Spurious FIQ flag |
| 6-0 | ACTIVEFIQ | 0-7Fh | Active FIQ number |

### 14.4.6 INTCPS_CONTROL Register

This register contains the new interrupt agreement bits.

**Figure 14-11. INTCPS_CONTROL Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | NEWFIQAGR | NEWIRQAGR |
| R-0 | | W-0 | W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-9. INTCPS_CONTROL Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Write 0s for future compatibility. Read returns reset value. |
| 1 | NEWFIQAGR | | Reset FIQ output and enable new FIQ generation. |
| | | W0 | No effect |
| | | W1 | Reset FIQ output and enable new FIQ generation. |
| 0 | NEWIRQAGR | | New IRQ generation |
| | | W0 | No effect |
| | | W1 | Reset IRQ output and enable new IRQ generation. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.4.7 INTCPS_PROTECTION Register

This register controls protection of the other registers. It can be accessed only in supervisor mode, regardless of the current value of the protection bit.

**Figure 14-12. INTCPS_PROTECTION Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | PROTECTION |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-10. INTCPS_PROTECTION Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Write 0s for future compatibility. Read returns reset value. |
| 0 | PROTECTION | | Protection mode |
| | | 0 | Protection mode is disabled. |
| | | 1 | Protection mode is enabled. When enabled, all the INTC registers are accessible only in privileged mode. |

### 14.4.8 INTCPS_IDLE Register

This register controls the functional clock auto-idle and the synchronizer clock auto-gating.

**Figure 14-13. INTCPS_IDLE Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | TURBO | FUNCIDLE |
| R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-11. INTCPS_IDLE Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Write 0s for future compatibility. Read returns reset value. |
| 1 | TURBO | | Input synchronizer clock auto-gating |
| | | 0 | Input synchronizer clock is free-running. |
| | | 1 | Input synchronizer clock is auto-gated based on interrupt input activity. |
| 0 | FUNCIDLE | | Functional clock idle mode |
| | | 0 | Functional clock gating strategy is applied. |
| | | 1 | Functional clock is free-running. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.4.9 INTCPS_IRQ_PRIORITY Register

This register supplies the currently active IRQ priority level.

**Figure 14-14. INTCPS_IRQ_PRIORITY Register**

| 31 | 16 |
|---|---|
| SPURIOUSIRQFLAG | |
| R-3FF FFFFh | |

| 15 | 6 | 5 | 0 |
|---|---|---|---|
| SPURIOUSIRQFLAG | | IRQPRIORITY | |
| R-3FF FFFFh | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-12. INTCPS_IRQ_PRIORITY Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | SPURIOUSIRQFLAG | 0-3FF FFFFh | Spurious IRQ flag |
| 5-0 | IRQPRIORITY | 0-3Fh | Current IRQ priority |

### 14.4.10 INTCPS_FIQ_PRIORITY Register

This register supplies the currently active FIQ priority level.

**Figure 14-15. INTCPS_FIQ_PRIORITY Register**

| 31 | 16 |
|---|---|
| SPURIOUSFIQFLAG | |
| R-3FF FFFFh | |

| 15 | 6 | 5 | 0 |
|---|---|---|---|
| SPURIOUSFIQFLAG | | FIQPRIORITY | |
| R-3FF FFFFh | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-13. INTCPS_FIQ_PRIORITY Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | SPURIOUSFIQFLAG | 0-3FF FFFFh | Spurious FIQ flag |
| 5-0 | FIQPRIORITY | 0-3Fh | Current FIQ priority |


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.4.11 INTCPS_THRESHOLD Register

This register sets the priority threshold.

**Figure 14-16. INTCPS_THRESHOLD Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | PRIORITYTHRESHOLD | |
| R-0 | | R/W-FFh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-14. INTCPS_THRESHOLD Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Write 0s for future compatibility. Read returns reset value. |
| 7-0 | PRIORITYTHRESHOLD | 0-FFh | Priority threshold |
| | | 0-7Fh | Priority threshold |
| | | FFh | Priority threshold is disabled. |

### 14.4.12 INTCPS_ITR0-3 Registers

This register shows the raw interrupt input status before masking.

**Figure 14-17. INTCPS_ITRn Register**

| 31 | 0 |
|---|---|
| ITR[*n*] | |
| R-x | |

LEGEND: R = Read only; -*n* = value after reset

**Table 14-15. INTCPS_ITRn Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ITR[*n*] | 0-FFFF FFFFh | Interrupt status before masking |


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.4.13 INTCPS_MIR0-3 Registers

This register contains the interrupt mask.

**Figure 14-18. INTCPS_MIRn Register**

| 31 | 0 |
|---|---|
| MR[*n*] | |
| R/W-FFFF FFFFh | |

LEGEND: R = Read only; -*n* = value after reset

**Table 14-16. INTCPS_MIRn Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | MR[*n*] | | Interrupt mask |
| | | 0 | Interrupt is unmasked. |
| | | 1 | Interrupt is masked. |

### 14.4.14 INTCPS_MIR_CLEAR0-3 Registers

This register is used to clear the interrupt mask bits.

**Figure 14-19. INTCPS_MIR_CLEARn Register**

| 31 | 0 |
|---|---|
| MIRCLEAR[*n*] | |
| W-0 | |

LEGEND: W = Write only; -*n* = value after reset

**Table 14-17. INTCPS_MIR_CLEARn Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | MIRCLEAR[*n*] | | Clear the interrupt mask [*n*] bits. Read returns 0 |
| | | W0 | No effect. |
| | | W1 | Clears the MIR mask bit [*n*] to 0. |

### 14.4.15 INTCPS_MIR_SET0-3 Registers

This register is used to set the interrupt mask bits.

**Figure 14-20. INTCPS_MIR_SETn Register**

| 31 | 0 |
|---|---|
| MIRSET[*n*] | |
| W-0 | |

LEGEND: W = Write only; -*n* = value after reset

**Table 14-18. INTCPS_MIR_SETn Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | MIRSET[*n*] | | Mask the interrupt [*n*] bits. Read returns 0. |
| | | W0 | No effect. |
| | | W1 | Set the MIR mask [*n*] bit to 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.4.16 INTCPS_ISR_SET0-3 Registers

This register is used to set the software interrupt bits. It is also used to read the currently active software interrupts.

**Figure 14-21. INTCPS_ISR_SETn Register**

| 31 | 0 |
|---|---|
| ISRSET[*n*] | |
| R/W-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 14-19. INTCPS_ISR_SETn Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ISRSET[*n*] | | Set the software interrupt [*n*] bits. Read returns the currently active software interrupts. |
| | | W0 | No effect. |
| | | W1 | Sets the software interrupt [*n*] bit to 1. |

### 14.4.17 INTCPS_ISR_CLEAR0-3 Registers

This register is used to clear the software interrupt bits.

**Figure 14-22. INTCPS_ISR_CLEARn Register**

| 31 | 0 |
|---|---|
| ISRCLEAR[*n*] | |
| R/W-FFFF FFFFh | |

LEGEND: R = Read only; -*n* = value after reset

**Table 14-20. INTCPS_ISR_CLEARn Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ISRCLEAR[*n*] | | Clear the software interrupt [*n*] bits. Read returns 0. |
| | | W0 | No effect. |
| | | W1 | Clears the software interrupt [*n*] bit to 0. |

### 14.4.18 INTCPS_PENDING_IRQ0-3 Registers

This register contains the IRQ status after masking.

**Figure 14-23. INTCPS_PENDING_IRQn Register**

| 31 | 0 |
|---|---|
| PENDINGIRQ | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 14-21. INTCPS_PENDING_IRQn Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | PENDINGIRQ | 0-FFFF FFFFh | IRQ status after masking. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 14.4.19 INTCPS_PENDING_FIQ0-3 Registers

This register contains the FIQ status after masking.

**Figure 14-24. INTCPS_PENDING_FIQn Register**

| 31 | 0 |
|---|---|
| PENDINGFIQ | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 14-22. INTCPS_PENDING_FIQn Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | PENDINGFIQ | 0-FFFF FFFFh | FIQ status after masking. |

### 14.4.20 INTCPS_ILR0-127 Registers

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

**Figure 14-25. INTCPS_ILRm Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | PRIORITY | | Reserved | FIQNIRQ |
| R-0 | | R/W-0 | | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 14-23. INTCPS_ILRm Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Write 0s for future compatibility |
| 7-2 | PRIORITY | 0-3Fh | Interrupt priority |
| 1 | Reserved | 0 | Write 0 for future compatibility. Read returns reset value. |
| 0 | FIQNIRQ | | Interrupt IRQ/FIQ mapping. Read returns reset value. |
| | | W0 | Interrupt is routed to IRQ. |
| | | W1 | Interrupt is routed to FIQ. |


Copyright © 2011–2016, Texas Instruments Incorporated

# Inter-Integrated Circuit (I2C) Controller Module

This chapter describes the multi-master inter-integrated circuit (I2C) controller module which provides an interface between a CPU and any I2C-bus-compatible device that connects via the I2C serial bus. External components attached to the I2C bus can serially transmit/receive up to 8-bit data to/from the CPU device through the two-wire I2C interface.

**Topic** **Page**

**15.1 Introduction** ..... **2890**
**15.2 Architecture** ..... **2891**
**15.3 I2C Registers** ..... **2903**


Copyright © 2011–2016, Texas Instruments Incorporated

## 15.1 Introduction

### 15.1.1 Overview

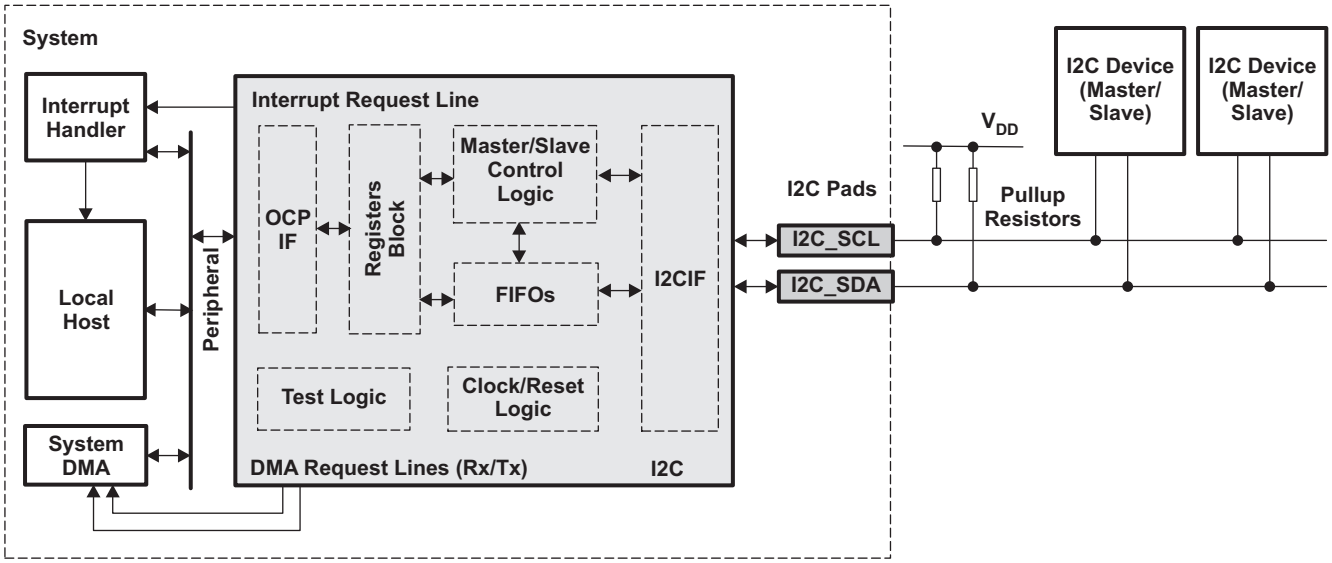
The multi-master I2C peripheral provides an interface between a CPU and any I2C-bus-compatible device that connects via the I2C serial bus. External components attached to the I2C bus can serially transmit/receive up to 8-bit data to/from the CPU device through the two-wire I2C interface.

The I2C bus is a multi-master bus. The I2C controller does support the multi-master mode that allows more than one device capable of controlling the bus to be connected to it. Each I2C device is recognized by a unique address and can operate as either transmitter or receiver, according to the function of the device. In addition to being a transmitter or receiver, a device connected to the I2C bus can also be considered as master or slave when performing data transfers. Note that a master device is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. During this transfer, any device addressed by this master is considered a slave.

### 15.1.2 Functional Block Diagram

Figure 15-1 shows an example of a system with multiple I2C compatible devices in which the I2C serial ports are all connected together for a two-way transfer from one device to other devices.

**Figure 15-1. I2C Functional Block Diagram**




### 15.1.3 Features

The multimaster I2C controller has the following features:

- Compliance with Philips I2C specification version 2.1
- Support for standard mode (up to 100K bits/s) and fast mode (up to 400K bits/s)
- 7-bit and 10-bit device addressing modes
- General call
- Start/restart/stop
- Multimaster transmitter/slave receiver mode
- Multimaster receiver/slave transmitter mode
- Combined master transmit/receive and receive/transmit mode
- Built-in FIFO size of 32 bytes for buffered read or write
- Module enable/disable capability


Copyright © 2011–2016, Texas Instruments Incorporated

- Programmable clock generation
- 8-bit-wide data access
- Designed for low-power consumption design
- Two DMA channels
- Wide interrupt capability

## 15.2 Architecture

The I2C peripheral consists of the following primary blocks:

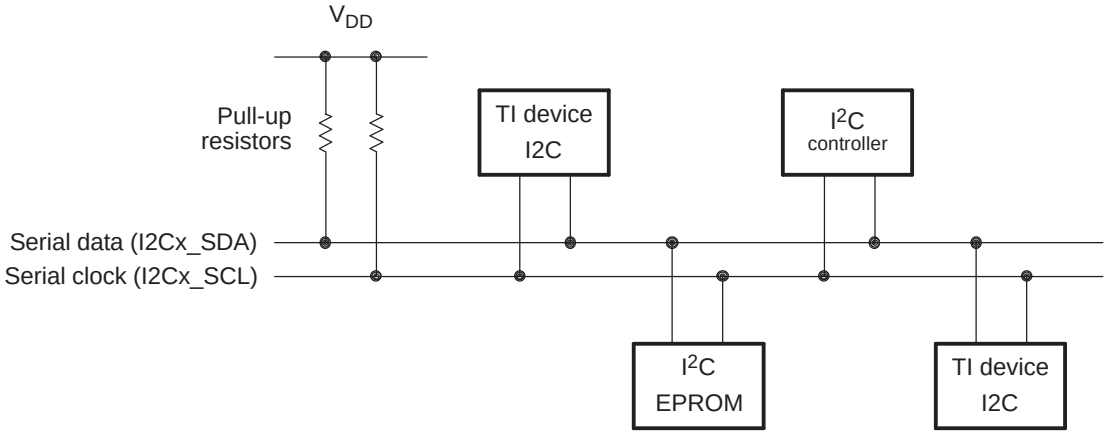
- A serial interface: one data pin (I2C_SDA) and one clock pin (I2C_SCL).
- Data registers to temporarily hold receive data and transmit data traveling between the I2C_SDA pin and the CPU or the DMA controller.
- Control and status registers
- A peripheral data bus interface to enable the CPU and the DMA controller to access the I2C peripheral registers.
- A clock synchronizer to synchronize the I2C input clock (from the processor clock generator) and the clock on the I2C_SCL pin, and to synchronize data transfers with masters of different clock speeds.
- A prescaler to divide down the input clock that is driven to the I2C peripheral
- A noise filter on each of the two pins, I2C_SDA and I2C_SCL
- An arbitrator to handle arbitration between the I2C peripheral (when it is a master) and another master
- Interrupt generation logic, so that an interrupt can be sent to the CPU
- DMA event generation logic to send an interrupt to the CPU upon reception or transmission of data.

### *15.2.1 I2C Master/Slave Contoller Signals*

Data is communicated to devices interfacing with the I2C via the serial data line (SDA) and the serial clock line (SCL). These two wires can carry information between a device and others connected to the I2C bus. Both SDA and SCL are bi-directional pins. They must be connected to a positive supply voltage via a pull-up resistor. When the bus is free, both pins are high. The driver of these two pins has an open drain to perform the required wired-AND function.

An example of multiple I2C modules that are connected for a two-way transfer from one device to other devices is shown in Figure 15-2.

**Figure 15-2. Multiple I2C Modules Connected**





Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-1. Signal Pads**

| Name | I2C Mode | |
|---|---|---|
| | Default Operating Mode | Description |
| I2C_SCL | In/ Out | I2C serial CLK line<br>Open-drain output buffer. Requires external pull-up resistor (Rp). |
| I2C_SDA | In/ Out | I2C serial data line<br>Open-drain output buffer. Requires external pull-up resistor (Rp). |

### 15.2.2 I2C Reset

The I2C module can be reset in the following three ways:

- A device reset causes all registers to be reset to their default values.
- A software reset by setting the SRST bit in the I2C_SYSC register. This bit has exactly the same action on the module logic as the device reset. All registers are reset to power up reset values.
- The I2C_EN bit in the I2C_CON register can be used to hold the I2C module in reset. When the device reset is removed, I2C_EN = 0 keeps the functional part of I2C module in reset state and all configuration registers can be accessed. I2C_EN = 0 does not reset the registers to power up reset values.

**Table 15-2. Reset State of I2C Signals**

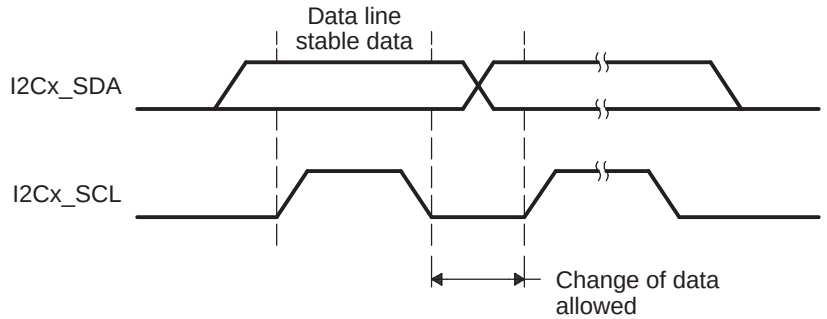
| Pin | I/O/Z [1] | System Reset | I2C Reset (I2C_EN = 0) |
|---|---|---|---|
| SDA | I/O/Z | High impedance | High impedance |
| SCL | I/O/Z | High impedance | High impedance |

(1) I = Input, O = Outpu, Z = High impedance

### 15.2.3 Data Validity

The data on the SDA line must be stable during the high period of the clock. The high and low states of the data line can only change when the clock signal on the SCL line is LOW.

**Figure 15-3. Bit Transfer on the I2C Bus**





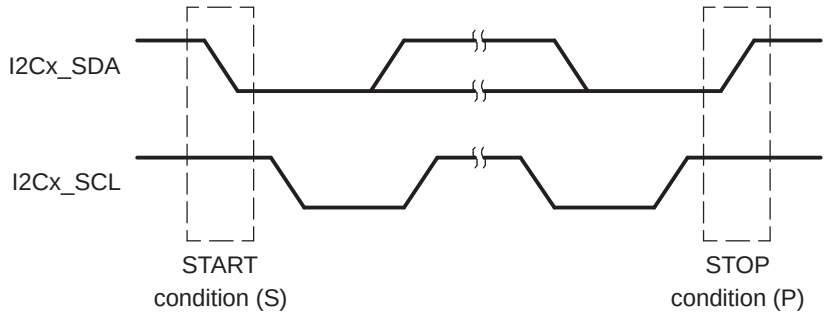
Copyright © 2011–2016, Texas Instruments Incorporated

### 15.2.4 START & STOP Conditions

The I2C module generates START and STOP conditions when it is configured as a master.

- START condition is a high-to-low transition on the SDA line while SCL is high.
- STOP condition is a low-to-high transition on the SDA line while SCL is high.
- The bus is considered to be busy after the START condition (BB = 1) and free after the STOP condition (BB = 0).

**Figure 15-4. Start and Stop Condition Events**

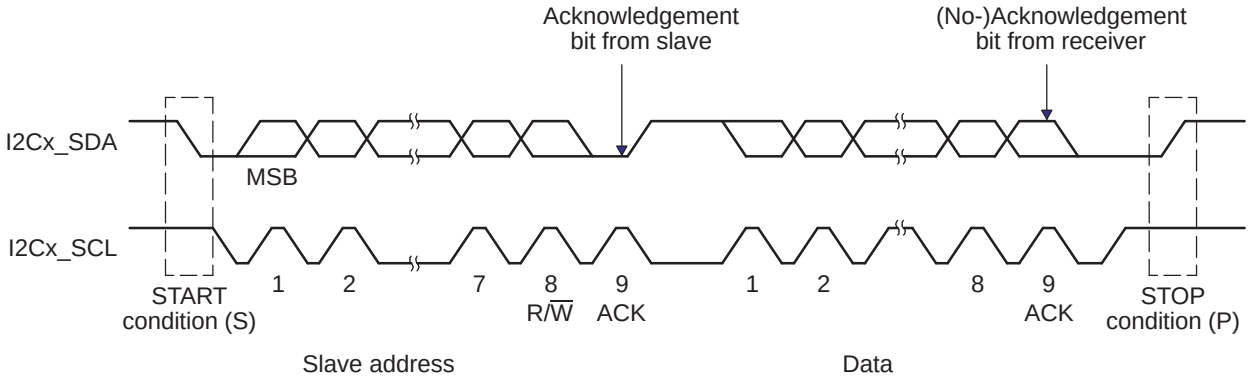



### 15.2.5 I2C Operation

#### 15.2.5.1 Serial Data Formats

The I2C controller operates in 8-bit word data format (byte write access supported for the last access). Each byte put on the SDA line is 8 bits long. The number of bytes that can be transmitted or received is restricted by the value programmed in the DCOUNT register. The data is transferred with the most significant bit (MSB) first. Each byte is followed by an acknowledge bit from the I2C module if it is in receiver mode.

**Figure 15-5. I2C Data Transfer**




The I2C module supports two data formats, as shown in Figure 15-6:

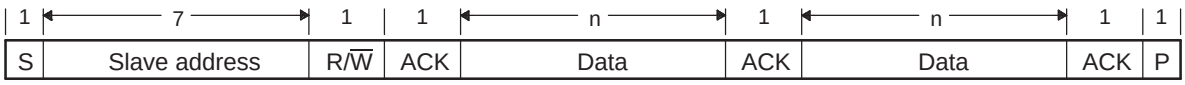
- 7-bit/10-bit addressing format
- 7-bit/10-bit addressing format with repeated start condition

The first byte after a start condition (S) always consists of 8 bits. In the acknowledge mode, an extra bit dedicated for acknowledgment is inserted after each byte. In the addressing formats with 7-bit addresses, the first byte is composed of 7 MSB slave address bits and 1 LSB R/$\overline{W}$ bit. In the addressing formats with 10-bit addresses, the first byte is composed of 7 MSB slave address bits, such as 11110XX, where XX is the two MSB of the 10-bit addresses, and 1 LSB R/$\overline{W}$ bit, which is 0 in this case.

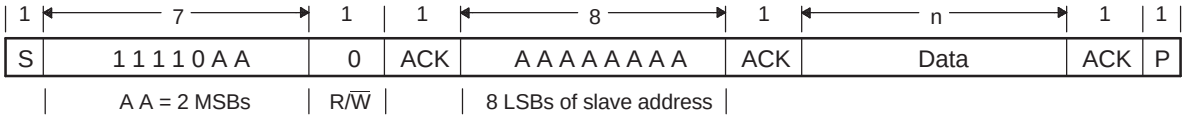

Copyright © 2011–2016, Texas Instruments Incorporated

The least significant R/$\overline{W}$ of the address byte indicates the direction of transmission of the following data bytes. If R/$\overline{W}$ is 0, the master writes data into the selected slave; if it is 1, the master reads data out of the slave.

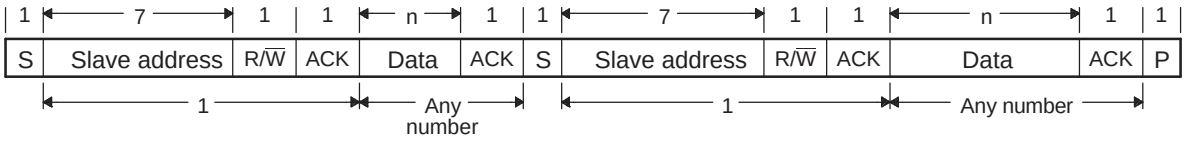
**Figure 15-6. I2C Data Transfer Formats**




7-Bit Addressing Format




10-Bit Addressing Format




7-Bit Addressing Format With Repeated START Condition

### 15.2.5.2 Master Transmitter

In this mode, data assembled in one of the previously described data formats is shifted out on the serial data line SDA in synch with the self-generated clock pulses on the serial clock line SCL. The clock pulses are inhibited and SCL held low when the intervention of the processor is required (XUDF) after a byte has been transmitted.

### 15.2.5.3 Master Receiver

This mode can only be entered from the master transmitter mode. With either of the address formats (Figure 6 (a), (b), and (c)), the master receiver is entered after the slave address byte and bit R/$\overline{W}$ has been transmitted, if R/$\overline{W}$ is high. Serial data bits received on bus line SDA are shifted in synch with the self-generated clock pulses on SCL. The clock pulses are inhibited and SCL held low when the intervention of the processor is required (ROVR) after a byte has been received. At the end of a transfer, it generates the stop condition.

### 15.2.5.4 Slave Transmitter

This mode can only be entered from the slave receiver mode. With either of the address formats (Figure 6 (a), (b), and (c)), the slave transmitter is entered if the slave address byte is the same as its own address and bit R/$\overline{W}$ has been transmitted, if R/$\overline{W}$ is high. The slave transmitter shifts the serial data out on the data line SDA in synch with the clock pulses that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the CPU is required (XUDF).

### 15.2.5.5 Slave Receiver

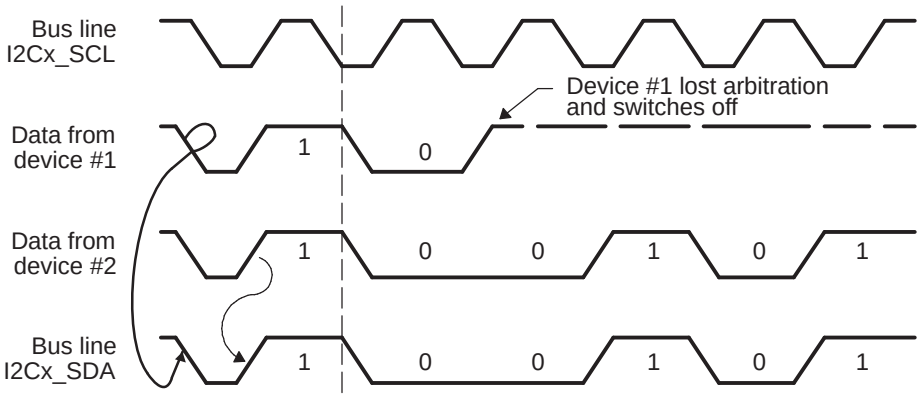
In this mode, serial data bits received on the bus line SDA are shifted-in in synch with the clock pulses on SCL that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the CPU is required (ROVR) following the reception of a byte.


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.2.6 Arbitration

If two or more master transmitters start a transmission on the same bus almost simultaneously, an arbitration procedure is invoked. The arbitration procedure uses the data presented on the serial bus by the competing transmitters. When a transmitter senses that a high signal it has presented on the bus has been overruled by a low signal, it switches to the slave receiver mode, sets the arbitration lost (AL) flag, and generates the arbitration lost interrupt. Figure 15-7 shows the arbitration procedure between two devices. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. Should two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

**Figure 15-7. Arbitration Procedure Between Two Master Transmitters**

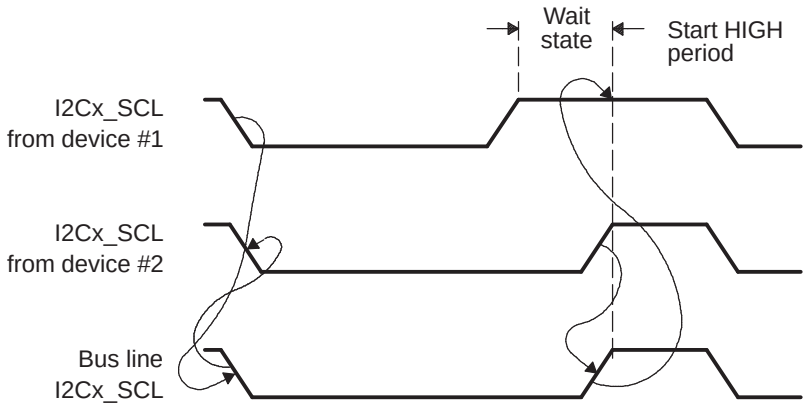



### 15.2.7 I2C Clock Generation and I2C Clock Synchronization

Under normal conditions, only one master device generates the clock signal, SCL. During the arbitration procedure, however, there are two or more master devices and the clock must be synchronized so that the data output can be compared. The wired-AND property of the clock line means that a device that first generates a low period of the clock line overrules the other devices. At this high/low transition, the clock generators of the other devices are forced to start generation of their own low period. The clock line is then held low by the device with the longest low period, while the other devices that finish their low periods must wait for the clock line to be released before starting their high periods. A synchronized signal on the clock line is thus obtained, where the slowest device determines the length of the low period and the fastest the length of the high period.

If a device pulls down the clock line for a longer time, the result is that all clock generators must enter the WAIT-state. In this way a slave can slow down a fast master and the slow device can create enough time to store a received byte or to prepare a byte to be transmitted. Figure 15-8 illustrates the clock synchronization.

**Figure 15-8. Synchronization of Two I2C Clock Generators**





Copyright © 2011–2016, Texas Instruments Incorporated

### 15.2.8 Prescaler (SCLK/ICLK)

The I2C module is operated with a functional clock (SCLK) frequency that can be in a range of 12-100 MHz, according to I2C mode that must be used (an internal ~24 MHz clock (ICLK) is recommended in case of F/S operation mode). Note that the frequency of the functional clock influences directly the I2C bus performance and timings.

The internal clock used for I2C logic - ICLK - is generated via the I2C prescaler block. The prescaler consists of a 4-bit register - I2C _PSC, and is used to divide the system clock (SCLK) to obtain the internal required clock for the I2C module.

### 15.2.9 Noise Filter

The noise filter is used to suppress any noise that is 50 ns or less, in the case of F/S mode of operation. It is designed to suppress noise with one ICLK. The noise filter is always one ICLK cycle, regardless of the bus speed. For FS mode (prescaler = 4, ICLK = 24 MHz), the maximum width of the suppressed spikes is 41.6 ns. To ensure a correct filtering, the prescaler must be programmed accordingly.

### 15.2.10 I2C Interrupts

The I2C module generates 12 types of interrupt: addressed as slave, bus free (stop condition detected), access error, start condition, arbitration-lost, no acknowledge, general call, registers-ready-for-access, receive and transmit data, receive and transmit draining. These 12 interrupts are accompanied with 12 interrupt masks and flags defined in the I2C_IRQENABLE_SET and respectively I2C_IRQSTATUS_RAW registers. Note that all these 12 interrupt events are sharing the same hardware interrupt line.

- Addressed As Slave interrupt (AAS) is generated to inform the Local Host that an external master addressed the module as a slave. When this interrupt occurs, the CPU can check the I2C_ACTOA status register to check which of the 4 own addresses was used by the external master to access the module.
- Bus Free interrupt (BF) is generated to inform the Local Host that the I2C bus became free (when a Stop Condition is detected on the bus) and the module can initiate his own I2C transaction.
- Start Condition interrupt (STC) is generated after the module being in idle mode have detected (synchronously or asynchronously) a possible Start Condition on the bus (signalized with WakeUp).
- Access Error interrupt (AERR) is generated if a Data read access is performed while RX FIFO is empty or a Data write access is performed while TX FIFO is full.
- Arbitration lost interrupt (AL) is generated when the I2C arbitration procedure is lost.
- No-acknowledge interrupt (NACK) is generated when the master I2C does not receive acknowledge from the receiver.
- General call interrupt (GC) is generated when the device detects the address of all zeros (8 bits).
- Registers-ready-for-access interrupt (ARDY) is generated by the I2C when the previously programmed address, data, and command have been performed and the status bits have been updated. This interrupt is used to let the CPU know that the I2C registers are ready for access.
- Receive interrupt/status (RRDY) is generated when there is received data ready to be read by the CPU from the I2C_DATA register (see the FIFO Management subsection (Section 15.2.13) for a complete description of required conditions for interrupt generation). The CPU can alternatively poll this bit to read the received data from the I2C_DATA register.
- Transmit interrupt/status (XRDY) is generated when the CPU needs to put more data in the I2C_DATA register after the transmitted data has been shifted out on the SDA pin (see the FIFO Management subsection (Section 15.2.13)for a complete description of required conditions for interrupt generation). The CPU can alternatively poll this bit to write the next transmitted data into the I2C_DATA register.
- Receive draining interrupt (RDR) is generated when the transfer length is not a multiple of threshold value, to inform the CPU that it can read the amount of data left to be transferred and to enable the draining mechanism. (see the Draining Feature subsection (Section 15.2.13.4) for additional details).
- Transmit draining interrupt (XDR) is generated when the transfer length is not a multiple of threshold value, to inform the CPU that it can write the amount of data left to be written and to enable the draining mechanism. (see the Draining Feature subsection (Section 15.2.13.4) for additional details).


Copyright © 2011–2016, Texas Instruments Incorporated

When the interrupt signal is activated, the Local Host must read the I2C_IRQSTATUS_RAW register to define the type of the interrupt, process the request, and then write into these registers the correct value to clear the interrupt flag.

### 15.2.11 DMA Events

The I2C module can generate two DMA requests events, read (I2C_DMA_RX) and write (I2C_DMA_TX) that can be used by the DMA controller to synchronously read received data from the I2C_DATA or write transmitted data to the I2C_DATA register. The DMA read and write requests are generated in a similar manner as RRDY and XRDY, respectively.

The I2C DMA request signals (I2C_DMA_TX and I2C_DMA_RX) are activated according to the FIFO Management subsection (Section 15.2.13).

### 15.2.12 Interrupt and DMA Events

I2C has two DMA channels (Tx and Rx). For event numbers, see the *Device Interrupts and EDMA Events* chapter.

I2C has one interrupt line for all the interrupt requests. For the interrupt number, see the *Device Interrupts and EDMA Events* chapter.

### 15.2.13 FIFO Management

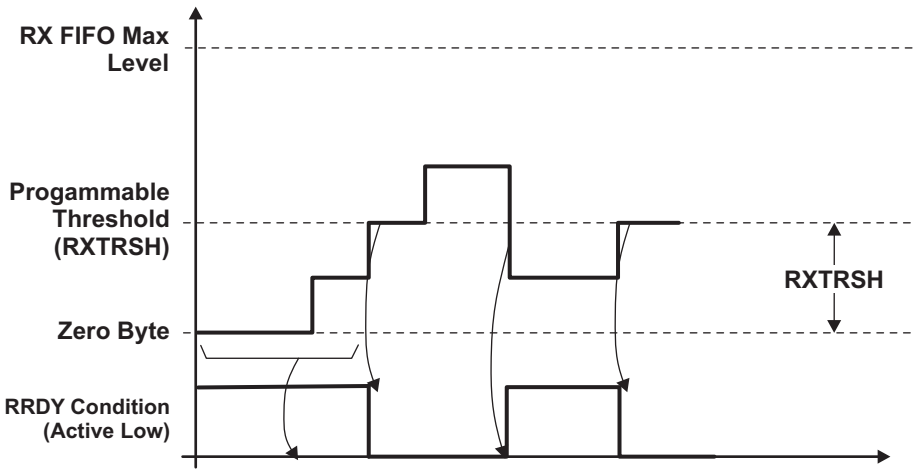
The I2C module implements two internal 32-bytes FIFOs with dual clock for RX and TX modes. The depth of the FIFOs is reflected in I2C_BUFSTAT.FIFODEPTH register.

#### 15.2.13.1 FIFO Interrupt Mode Operation

In FIFO interrupt mode (relevant interrupts enabled via I2C_IRQENABLE_SET register), the processor is informed of the status of the receiver and transmitter by an interrupt signal. These interrupts are raised when receive/transmit FIFO threshold (defined by I2C_BUF.TXTRSH or I2C_BUF.RXTRSH) is reached; the interrupt signals instruct the Local Host to transfer data to the destination (from the I2C module in receive mode and/or from any source to the I2C FIFO in transmit mode).

Figure 15-9 and Figure 15-10, respectively, illustrate receive and transmit operations from FIFO management point of view.

**Figure 15-9. Receive FIFO Interrupt Request Generation**

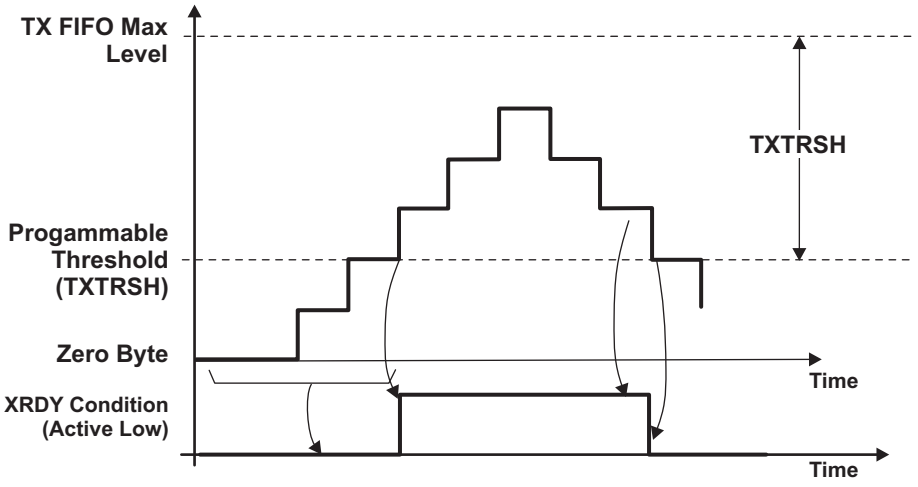



Note that in Figure 15-9, the RRDY Condition illustrates that the condition for generating a RRDY interrupt is achieved. The interrupt request is generated when this signal is active, and it can be cleared only by the CPU by writing a 1 in the corresponding interrupt flag. If the condition is still present after clearing the previous interrupt, another interrupt request will be generated.


Copyright © 2011–2016, Texas Instruments Incorporated

In receive mode, RRDY interrupt is not generated until the FIFO reaches its receive threshold. Once low, the interrupt can only be de-asserted when the Local Host has handled enough bytes to make the FIFO level below threshold. For each interrupt, the Local Host can be configured to read an amount of bytes equal with the value of the RX FIFO threshold + 1.

**Figure 15-10. Transmit FIFO Interrupt Request Generation**




Note that in the figure above, the XRDY Condition illustrates that the condition for generating a XRDY interrupt is achieved. The interrupt request is generated when this condition is achieved (when TX FIFO is empty, or the TX FIFO threshold is not reached and there are still data bytes to be transferred in the TX FIFO), and it can be cleared only by the CPU by writing a 1 in the corresponding interrupt flag after transmitting the configured number of bytes. If the condition is still present after clearing the previous interrupt, another interrupt request will be generated.

Note that in interrupt mode, the module offers two options for the CPU application to handle the interrupts:

- When detecting an interrupt request (XRDY or RRDY type), the CPU can write/read one data byte to/from the FIFO and then clear the interrupt. The module will not reassert the interrupt until the interrupt condition is not met.
- When detecting an interrupt request (XRDY or RRDY type), the CPU can be programmed to write/read the amount of data bytes specified by the corresponding FIFO threshold (I2C_BUF.TXTRSH + 1 or I2C_BUF.RXTRSH + 1). In this case, the interrupt condition will be cleared and the next interrupt will be asserted again when the XRDY or RRDY condition is met again.

If the second interrupt serving approach is used, an additional mechanism (draining feature) is implemented for the case when the transfer length is not a multiple of FIFO threshold (see the Draining Feature subsection (Section 15.2.13.4)).

In slave TX mode, the draining feature cannot be used, since the transfer length is not known at the configuration time, and the external master can end the transfer at any point by not acknowledging one data byte.

### 15.2.13.2 FIFO Polling Mode Operation

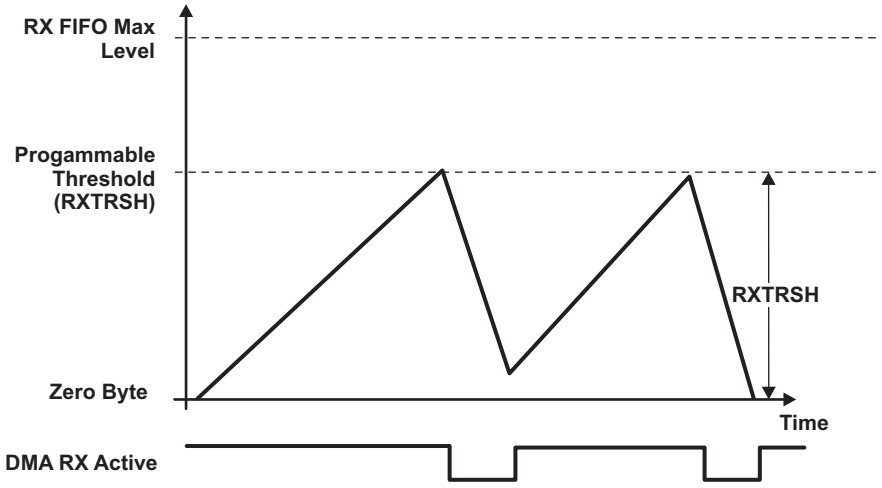
In FIFO polled mode (I2C_IRQENABLE_SET.XRDY_IE and I2C_IRQENABLE_SET.RRDY_IE disabled and DMA disabled), the status of the module (receiver or transmitter) can be checked by polling the XRDY and RRDY status registers (I2C_IRQSTATUS_RAW) (RDR and XDR can also be polled if draining feature is used). The XRDY and RRDY flags are accurately reflecting the interrupt conditions mentioned in Interrupt Mode. This mode is an alternative to the FIFO interrupt mode of operation, where the status of the receiver and transmitter is automatically known by means of interrupts sent to the CPU.


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.2.13.3 FIFO DMA Mode Operation

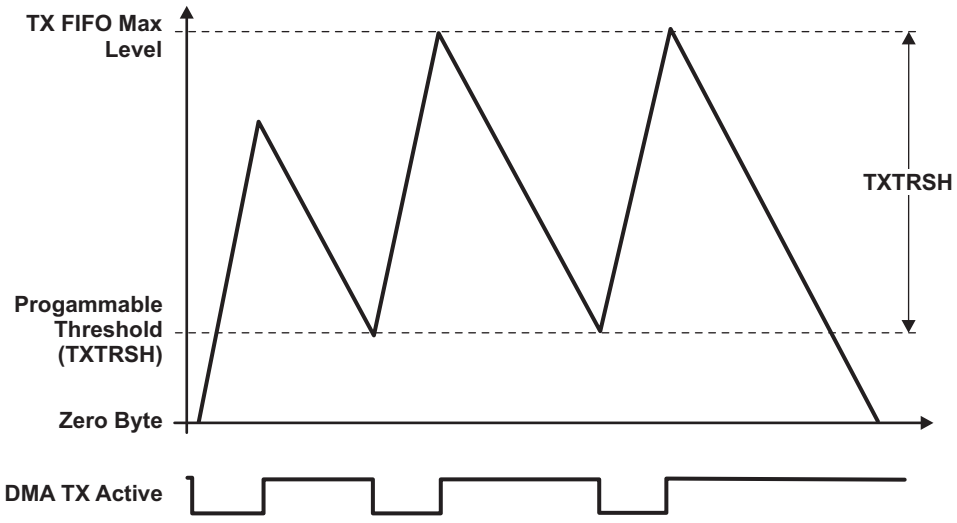
In receive mode, a DMA request is generated as soon as the receive FIFO exceeds its threshold level defined in the threshold level register (I2C_BUF.RXTRSH +1). This request should be de-asserted when the number of bytes defined by the threshold level has been read by the DMA, by setting the I2C_DMARXENABLE_CLR.DMARX_ENABLE_CLEAR field.

**Figure 15-11. Receive FIFO DMA Request Generation**




In transmit mode, a DMA request is automatically asserted when the transmit FIFO is empty. This request should be de-asserted when the number of bytes defined by the number in the threshold register (I2C_BUF.TXTHRS+1) has been written in the FIFO by the DMA, by setting the I2C_DMATXENABLE_CLR. DMATX_ENABLE_CLEAR field. If an insufficient number of characters are written, then the DMA request will remain active. Figure 15-12 and Figure 15-13 illustrate DMA TX transfers with different values for TXTRSH.

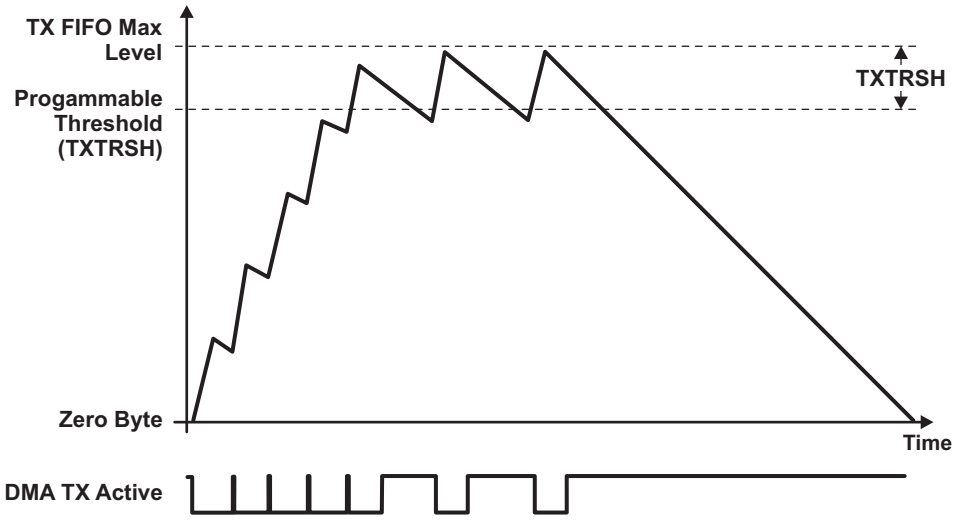
**Figure 15-12. Transmit FIFO DMA Request Generation (High Threshold)**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 15-13. Transmit FIFO DMA Request Generation (Low Threshold)**




Note that also in DMA mode it is possible to have a transfer whose length is not multiple of the configured FIFO threshold. In this case, the DMA draining feature is also used for transferring the additional bytes of the transfer (see the Draining Feature subsection (Section 15.2.13.4) for additional details).

In I2C Slave RX Mode, the Local Host can program the RX threshold with the desired value, and use the FIFO draining feature at the end of the I2C transfer to extract from the FIFO the remaining bytes if the threshold is not reached (see the Draining Feature subsection (Section 15.2.13.4) for additional details).

Note that in I2C Slave TX Mode, the TX FIFO threshold should be set to 1 (I2C_BUF.TXTRSH=0, default value), since the length of the transfer may not be known at configuration time. In this way, the interrupt (or accordingly, DMA) requests will be generated for each byte requested by the remote I2C master to be transferred over the I2C bus. This configuration will prevent the I2C core to request additional data from the CPU or from the DMA controller (using IRQ or DMA), data that will eventually not be extracted from the FIFO by the external master (which can use not acknowledge at any time to end the transfer). If the TX threshold is not set to 1, the module will generate an interrupt or assert a DMA only when the external master requests a byte and the FIFO is empty. However, in this case the TX FIFO will require to be cleared at the end of the transfer.

The I2C module offers the possibility to the user to clear the RX or TX FIFO. This is achieved through I2C_BUF.RXFIFO_CLR and I2C_BUF.TXFIFO_CLR registers, which act like software reset for the FIFOs. In DMA mode, these bits will also reset the DMA state machines.

The FIFO clearing feature can be used when the module is configured as a transmitter, the external receiver responds with a NACK in the middle of the transfer, and there is still data in TX FIFO waiting to be transferred.

On the Functional (I2C) domain, the thresholds can always be considered equal to 1. This means that the I2C Core can start transferring data on the I2C bus whenever it has data in the FIFOs (FIFO is not empty).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 15.2.13.4 Draining Feature

The Draining Feature is implemented by the I2C core for handling the end of the transfers whose length is not a multiple of FIFO threshold value, and offers the possibility to transfer the remaining amount of bytes (since the threshold is not reached).

Note that this feature prevents the CPU or the DMA controller to attempt more FIFO accesses than necessary (for example, to generate at the end of a transfer a DMA RX request having in the FIFO less bytes than the configured DMA transfer length). Otherwise, an Access Error interrupt will be generated (see I2C_IRQSTATUS_RAW.AERR interrupt).

The Draining mechanism will generate an interrupt (I2C_IRQSTATUS_RAW.RDR or I2C_IRQSTATUS_RAW.XDR) at the end of the transfer informing the CPU that it needs to check the amount of data left to be transferred (I2C_BUFSTAT.TXSTAT or RXSTAT) and to enable the Draining Feature of the DMA controller if DMA mode is enabled (by re-configuring the DMA transfer length according to this value), or perform only the required number of data accesses, if DMA mode is disabled.

In receiving mode (master or slave), if the RX FIFO threshold is not reached but the transfer was ended on the I2C bus and there is still data left in the FIFO (less than the threshold), the receive draining interrupt (I2C_IRQSTATUS_RAW.RDR) will be asserted to inform the local host that it can read the amount of data in the FIFO (I2C_BUFSTAT.RXSTAT). The CPU will perform a number of data read accesses equal with RXSTAT value (if interrupt or polling mode) or re-configure the DMA controller with the required value in order to drain the FIFO.

In master transmit mode, if the TX FIFO threshold is not reached but the amount of data remained to be written in the FIFO is less than TXTRSH, the transmit draining interrupt (I2C_IRQSTATUS_RAW.XDR) will be asserted to inform the local host that it can write the amount of data remained to be written in the TX FIFO (I2C_BUFSTAT.TXSTAT). The CPU will need to write the required number of data bytes (specified by TXSTAT value) or re-configure the DMA controller with the required value in order to transfer the last bytes to the FIFO.

Note that in master mode, the CPU can alternatively skip the checking of TXSTAT and RXSTAT values since it can obtain internally this information (by computing DCOUNT modulo TX/RXTRSH).

The draining feature is default disabled, and it can be enabled using I2C_IRQENABLE_SET.XDR_IE or I2C_IRQENABLE_SET.RDR_IE registers (default disabled) only for the transfers with length not equal with the threshold value.

### *15.2.14 How to Program I2C*

#### 15.2.14.1 Module Configuration Before Enabling the Module

1. Program the prescaler to obtain an approximately 12-MHz I2C module clock (I2C_PSC = x; this value is to be calculated and is dependent on the System clock frequency).
2. Program the I2C clock to obtain 100 Kbps or 400 Kbps (SCLL = x and SCLH = x; these values are to be calculated and are dependent on the System clock frequency).
3. Configure its own address (I2C_OA = x) - only in case of I2C operating mode (F/S mode).
4. Take the I2C module out of reset (I2C_CON:I2C_EN = 1).

#### 15.2.14.2 Initialization Procedure

1. Configure the I2C mode register (I2C_CON) bits.
2. Enable interrupt masks (I2C_IRQENABLE_SET), if using interrupt for transmit/receive data.
3. Enable the DMA (I2C_BUF and I2C_DMA/RX/TX/ENABLE_SET) and program the DMA controller - only in case of I2C operating mode (F/S mode), if using DMA for transmit/receive data.

#### 15.2.14.3 Configure Slave Address and DATA Counter Registers

In master mode, configure the slave address (I2C_SA = x) and the number of bytes associated with the transfer (I2C_CNT = x).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 15.2.14.4 Initiate a Transfer

Poll the bus busy (BB) bit in the I2C status register (I2C_IRQSTATUS_RAW). If it is cleared to 0 (bus not busy), configure START/STOP (I2C_CON: STT / I2C_CON: STP condition to initiate a transfer) - only in case of I2C operating mode (F/S mode).

#### 15.2.14.5 Receive Data

Poll the receive data ready interrupt flag bit (RRDY) in the I2C status register (I2C_IRQSTATUS_RAW), use the RRDY interrupt (I2C_IRQENABLE_SET.RRDY_IE set) or use the DMA RX (I2C_BUF.RDMA_EN set together with I2C_DMARXENABLE_SET) to read the receive data in the data receive register (I2C_DATA). Use draining feature (I2C_IRQSTATUS_RAW.RDR enabled by I2C_IRQENABLE_SET.RDR_IE)) if the transfer length is not equal with FIFO threshold.

#### 15.2.14.6 Transmit Data

Poll the transmit data ready interrupt flag bit (XRDY) in the I2C status register (I2C_IRQSTATUS_RAW), use the XRDY interrupt (I2C_IRQENABLE_SET.XRDY_IE set) or use the DMA TX (I2C_BUF.XDMA_EN set together with I2C_DMATXENABLE_SET) to write data into the data transmit register (I2C_DATA). Use draining feature (I2C_IRQSTATUS_RAW.XDR enabled by I2C_IRQENABLE_SET.XDR_IE)) if the transfer length is not equal with FIFO threshold.


Copyright © 2011–2016, Texas Instruments Incorporated

## 15.3 I2C Registers

Table 15-3 lists the registers for the I2C module.

**NOTE:** All bits defined as reserved must be written by software with 0s, for preserving future compatibility. When read, any reserved bit returns 0. Also, note that it is good software practice to use complete mask patterns for setting or testing individually bit fields within a register.

**Table 15-3. I2C Registers**

| Address Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 00h | I2C_REVNB_LO | Module Revision Register (low bytes) | Section 15.3.1 |
| 04h | I2C_REVNB_HI | Module Revision Register (high bytes) | Section 15.3.2 |
| 10h | I2C_SYSC | System Configuration Register | Section 15.3.3 |
| 20h | I2C_EOI | I2C End of Interrupt Register | Section 15.3.4 |
| 24h | I2C_IRQSTATUS_RAW | I2C Status Raw Register | Section 15.3.5 |
| 28h | I2C_IRQSTATUS | I2C Status Register | Section 15.3.6 |
| 2Ch | I2C_IRQENABLE_SET | I2C Interrupt Enable Set Register | Section 15.3.7 |
| 30h | I2C_IRQENABLE_CLR | I2C Interrupt Enable Clear Register | Section 15.3.8 |
| 34h | I2C_WE | I2C Wakeup Enable Register | Section 15.3.9 |
| 38h | I2C_DMARXENABLE_SET | Receive DMA Enable Set Register | Section 15.3.10 |
| 3Ch | I2C_DMATXENABLE_SET | Transmit DMA Enable Set Register | Section 15.3.11 |
| 40h | I2C_DMARXENABLE_CLR | Receive DMA Enable Clear Register | Section 15.3.12 |
| 44h | I2C_DMATXENABLE_CLR | Transmit DMA Enable Clear Register | Section 15.3.13 |
| 48h | I2C_DMARXWAKE_EN | Receive DMA Wakeup Register | Section 15.3.14 |
| 4Ch | I2C_DMATXWAKE_EN | Transmit DMA Wakeup Register | Section 15.3.15 |
| 90h | 2C_SYSS | System Status Register | Section 15.3.16 |
| 94h | I2C_BUF | Buffer Configuration Register | Section 15.3.17 |
| 98h | I2C_CNT | Data Counter Register | Section 15.3.18 |
| 9Ch | I2C_DATA | Data Access Register | Section 15.3.19 |
| A4h | I2C_CON | I2C Configuration Register | Section 15.3.20 |
| A8h | I2C_OA | I2C Own Address Register | Section 15.3.21 |
| ACh | I2C_SA | I2C Slave Address Register | Section 15.3.22 |
| B0h | I2C_PSC | I2C Clock Prescaler Register | Section 15.3.23 |
| B4h | I2C_SCLL | I2C SCL Low Time Register | Section 15.3.24 |
| B8h | I2C_SCLH | I2C SCL High Time Register | Section 15.3.25 |
| BCh | I2C_SYSTEST | System Test Register | Section 15.3.26 |
| C0h | I2C_BUFSTAT | I2C Buffer Status Register | Section 15.3.27 |
| C4h | I2C_OA1 | I2C Own Address 1 Register | Section 15.3.28 |
| C8h | I2C_OA2 | I2C Own Address 2 Register | Section 15.3.29 |
| CCh | I2C_OA3 | I2C Own Address 3 Register | Section 15.3.30 |
| D0h | I2C_ACTOA | Active Own Address Register | Section 15.3.31 |
| D4h | I2C_SBLOCK | I2C Clock Blocking Enable Register | Section 15.3.32 |


Copyright © 2011–2016, Texas Instruments Incorporated

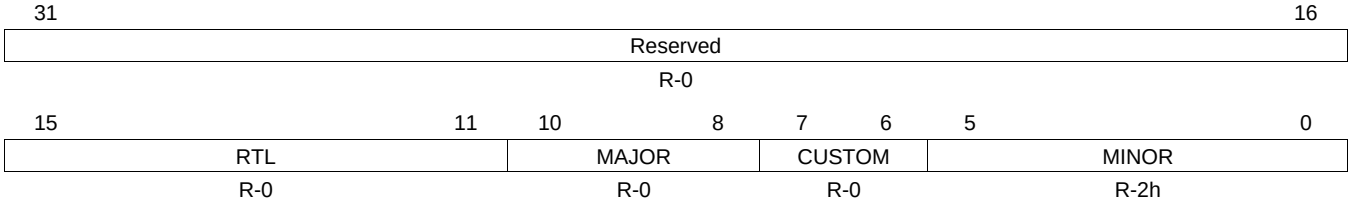
### 15.3.1 I2C_REVNB_LO Register (Module Revision LOW BYTES)

This read-only register contains the hard-coded revision number of the module. A write to this register has no effect.

**NOTE:**

- I2C controller with interrupt using interrupt vector register (I2C_IV) is revision 1.x.
- I2C controller with interrupt using status register bits (I2C_IRQSTATUS_RAW) is revision 2.x.

**Figure 15-14. I2C_REVNB_LO Register (Module Revision) (LOW BYTES)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-4. I2C_REVNB_LO Register (Module Revision) (LOW BYTES) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15-11 | RTL | 0-1Fh | RTL version. |
| 10-8 | MAJOR | 0-7h | Major Revision.<br>This field changes when there is a major feature change.<br>This field does not change due to bug fix, or minor feature change. |
| 7-6 | CUSTOM | 0-3h | Indicates a special version for a particular device. Consequence of use may avoid use of standard Chip Support Library (CSL) / Drivers.<br>0 if non-custom. |
| 5-0 | MINOR | 0-3Fh | Minor Revision<br>This field changes when features are scaled up or down.<br>This field does not change due to bug fix, or major feature change. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.2 I2C_REVNB_HI Register (HIGH BYTES) (Module Revision)

A reset has no effect on the value returned.

**Figure 15-15. I2C_REVNB_HI Register (HIGH BYTES) (Module Revision)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 12 | 11 | 0 |
|---|---|---|---|---|---|
| SCHEME | | Reserved | | FUNC | |
| R-1 | | R-1 | | R-40h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-5. I2C_REVNB_HI Register (HIGH BYTES) (Module Revision) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15-14 | SCHEME | 0-3h | Used to distinguish between old Scheme and current.<br>Spare bit to encode future schemes. |
| 13-12 | Reserved | 0-3h | Reads return 1h |
| 11-0 | FUNC | 0-FFFh | Function: Indicates a software compatible module family |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.3 I2C_SYSC Register (System Configuration)

This register allows controlling various parameters of the peripheral interface.

**Figure 15-16. I2C_SYSC Register (System Configuration)**

| 31 | | | | | | | | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |
| 15 | | | | 10 | 9 | 8 | 7 | | 5 | 4 | 3 | 2 | 1 | 0 | |
| Reserved | | | | | CLKACTIVITY | | Reserved | | | IDLEMODE | | ENAWAKEUP | SRST | AUTOIDLE | |
| R-0 | | | | | R/W-0 | | R-0 | | | R/W-0 | | R/W-0 | R/W-0 | R/W-1 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-6. I2C_SYSC Register (System Configuration) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9-8 | CLKACTIVITY | | Clock Activity selection bits. Those bits (one bit for each clock signal present on the boundary of the module) are set to 1 to disable external clock gating mechanism in Idle Mode. |
| | | 0h | Both clocks can be cut off |
| | | 1h | Only Interface/OCP clock must be kept active; system clock can be cut off |
| | | 2h | Only system clock must be kept active; Interface/OCP clock can be cut off |
| | | 3h | Both clocks must be kept active |
| | | | Values after reset are low (for both 2 bits). |
| | | | **Note:** If the System (functional) Clock is cut-off, the module will assert a WakeUp event when it will asynchronously detect a Start Condition on the I2C Bus. Note that in this case the first transfer will not be taken into account by the module (NACK will be detected by the external master). |
| 7-5 | Reserved | 0 | Reads return 0h. |
| 4-3 | IDLEMODE | | Idle Mode selection bits. These two bits are used to select one of the idle mode operation mechanisms. |
| | | 0h | Force Idle mode |
| | | 1h | No Idle mode |
| | | 2h | Smart Idle mode |
| | | 3h | smartidle_wakeup |
| | | | Value after reset is 00 (Force Idle). |
| 2 | ENAWAKEUP | | Enable Wakeup control bit. When this bit is set to 1, the module enables its own wakeup mechanism. |
| | | 0 | Wakeup mechanism is disabled |
| | | 1 | Wakeup mechanism is enabled |
| | | | Value after reset is low. |
| 1 | SRST | | SoftReset bit. When this bit is set to 1, entire module is reset as for the hardware reset. This bit is automatically cleared to 0 by the core and it is only reset by the hardware reset. During reads, it always returns 0. |
| | | 0 | Normal mode |
| | | 1 | The module is reset |
| | | | Value after reset is low. |
| 0 | AUTOIDLE | | Autoidle bit. When this bit is set to 1, the module activates its own idle mode mechanism. By evaluating its internal state, the module can decide to gate part of his internal clock tree in order to improve the overall power consumption. |
| | | 0 | Auto Idle mechanism is disabled |
| | | 1 | Auto Idle mechanism is enabled |
| | | | Value after reset is high. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.4 I2C_EOI Register (I2C End of Interrupt)

Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if an new interrupt event is pending, when using the pulsed output. Unused when using the level interrupt line (depending on module integration).

**Figure 15-17. I2C_EOI Register (I2C End of Interrupt)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | LINE_NUMBER |
| R-0 | | W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-7. I2C_EOI Register (I2C End of Interrupt) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-1 | Reserved | Reserved |
| 0 | LINE_NUMBER | Software End Of Interrupt (EOI) control. Write number of interrupt output. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.5 I2C_IRQSTATUS_RAW Register (I2C Status Raw)

This register provides core status information for interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit will set it to 1, that is, trigger the IRQ (mostly for debug). Writing a 0 will have no effect, that is, the register value will not be modified. Only enabled, active events will trigger an actual interrupt request on the IRQ output line.

**Figure 15-18. I2C_IRQSTATUS_RAW Register (I2C Status Raw)**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XDR | RDR | BB | ROVR | XUDF | AAS | BF |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AERR | STC | GC | XRDY | RRDY | ARDY | NACK | AL |
| R/W-0 | R/W-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-8. I2C_IRQSTATUS_RAW Register (I2C Status Raw) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Write 0s for future compatibility. Read returns 0. |
| 14 | XDR | | Transmit draining IRQ status. I2C Master Transmit mode only.<br>This read/clear only bit is set to 1 when the module is configured as a master transmitter, the TX FIFO level is below the configured threshold (TXTRSH) and the amount of data still to be transferred is less than TXTRSH. When this bit is set to 1 by the core, CPU must read the I2C_BUFSTAT.TXSTAT register in order to check the amount of data that need to be written in the TX FIFO. Then, according to the mode set (DMA or interrupt), the CPU can enable the DMA draining feature of the DMA controller with the number of data bytes to be transferred (I2C_BUFSTAT.TXSTAT), or generate write data accesses according to this value (IRQ mode). The interrupt needs to be cleared after the DMA controller was reconfigured (if DMA mode enabled), or before generating data accesses to the FIFO (if IRQ mode enabled).<br>If the corresponding interrupt was enabled, an interrupt is signaled to the local host. The CPU can also poll this bit. For more details about TDR generation, refer to the FIFO Management subsection.<br>The CPU can only clear this bit by writing a 1 into this register. A write 0 has no effect. |
| | | 0 | Transmit draining inactive |
| | | 1 | Transmit draining enabled |
| | | | Value after reset is low. |
| 13 | RDR | | Receive draining IRQ status. I2C Receive mode only.<br>This read/clear only bit is set to 1 when the module is configured as a receiver, a stop condition was received on the bus and the RX FIFO level is below the configured threshold (RXTRSH). When this bit is set to 1 by the core, CPU must read the I2C_BUFSTAT.RXSTAT register in order to check the amount of data left to be transferred from the FIFO. Then, according to the mode set (DMA or interrupt), the CPU needs to enable the draining feature of the DMA controller with the number of data bytes to be transferred (I2C_BUFSTAT.RXSTAT), or generate read data accesses according to this value (IRQ mode). The interrupt needs to be cleared after the DMA controller was reconfigured (if DMA mode enabled), or before generating data accesses to the FIFO (if IRQ mode enabled).<br>If the corresponding interrupt was enabled, an interrupt is signaled to the local host. The CPU can also poll this bit. For more details about RDR generation, refer to the FIFO Management subsection.<br>The CPU can only clear this bit by writing a 1 into this register. A write 0 has no effect. |
| | | 0 | Receive draining inactive |
| | | 1 | Receive draining enabled |
| | | | Value after reset is low. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-8. I2C_IRQSTATUS_RAW Register (I2C Status Raw) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 12 | BB | | This read-only bit indicates the state of the serial bus.<br>In slave mode, on reception of a start condition, the device sets BB to 1. BB is cleared to 0 after reception of a stop condition.<br>In master mode, the software controls BB. To start a transmission with a start condition, MST, TRX, and STT must be set to 1 in the I2C_CON register. To end a transmission with a stop condition, STP must be set to 1 in the I2C_CON register. When BB = 1 and STT = 1, a restart condition is generated. |
| | | 0 | Bus is free |
| | | 1 | Bus is occupied |
| | | | Value after reset is low. |
| 11 | ROVR | | Receive overrun status. Writing into this bit has no effect. I2C receive mode only.<br>This read-only bit indicates whether the receiver has experienced overrun. Overrun occurs when the shift register is full and the receive FIFO is full. An overrun condition does not result in a data loss; the peripheral is just holding the bus (low on SCL) and prevents other bytes from being received.<br>ROVR is set to 1 when the I2C has recognized an overrun. ROVR is clear when reading I2C_DATA register, or when resetting the I2C (I2C_CON:I2C_EN = 0). |
| | | 0 | Normal operation |
| | | 1 | Receiver overrun |
| | | | Value after reset is low. |
| 10 | XUDF | | Transmit underflow status. Writing into this bit has no effect. I2C transmit mode only.<br>This read-only bit indicates whether the transmitter has experienced underflow. In master transmit mode, underflow occurs when the shift register is empty, the transmit FIFO is empty, and there are still some bytes to transmit (DCOUNT = 0).<br>In slave transmit mode, underflow occurs when the shift register is empty, the transmit FIFO is empty, and there are still some bytes to transmit (read request from external I2C master).<br>XUDF is set to 1 when the I2C has recognized an underflow. The core holds the line till the underflow cause has disappeared.<br>XUDF is clear when writing I2C_DATA register or resetting the I2C (I2C_CON:I2C_EN = 0). |
| | | 0 | Normal operation |
| | | 1 | Transmit underflow |
| | | | Value after reset is low. |
| 9 | AAS | | Address recognized as slave IRQ status. I2C mode only.<br>This read only bit is set to 1 by the device when it has recognized its own slave address (or one of the alternative own addresses), or an address of all zeros (8 bits). When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled.<br>This bit can be cleared in 2 ways:<br>• If the interrupt was enabled, it will be cleared by writing 1 into this register (writing 0 has no effect)<br>• If the interrupt was not enabled, the AAS bit is reset to 0 by restart or stop |
| | | 0 | No action |
| | | 1 | Address recognized |
| | | | Value after reset is low. |
| 8 | BF | | I2C mode only.<br>This read only bit is set to 1 by the device when the I2C bus became free (after a transfer is ended on the bus – stop condition detected). This interrupt informs the Local Host that it can initiate its own I2C transfer on the bus.<br>When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect. |
| | | 0 | No action |
| | | 1 | Bus Free |
| | | | Value after reset is low. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-8. I2C_IRQSTATUS_RAW Register (I2C Status Raw) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7 | AERR | | Access Error IRQ status. I2C mode only.<br>This read/clear only bit is set to 1 by the device if an Interface/OCP write access is performed to I2C_DATA while the TX FIFO is full or if an Interface/OCP read access is performed to the I2C_DATA while the RX FIFO is empty.<br>Note that, when the RX FIFO is empty, a read access will return to the previous read data value. When the TX FIFO is full, a write access is ignored. In both events, the FIFO pointers will not be updated.<br>When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect. |
| | | 0 | No action |
| | | 1 | Access Error |
| | | | Value after reset is low. |
| 6 | STC | | Start Condition IRQ status. I2C mode only.<br>This read/clear only bit is set to 1 by the device if previously the module was in idle mode and a start condition was asynchronously detected on the I2C Bus and signalized with an Wakeup (if the I2C_SYSC.ClockActivity allows the system clock to be cut-off). When the Active Mode will be restored and the interrupt generated, this bit will indicate the reason of the wakeup.<br>**Note 1:** The corresponding interrupt for this bit should be enabled only if the module was configured to allow the possibility of cutting-off the system clock while in Idle State (I2C_SYSC.ClockActivity = 00 or 01).<br>**Note 2:** The first transfer (corresponding to the detected start condition) will be lost (not taken into account by the module) and it will be used only for generating the WakeUp enable for restoring the Active Mode of the module. On the I2C line, the external master which generated the transfer will detect this behavior as a not acknowledge to the address phase and will possibly restart the transfer.<br>The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect. |
| | | 0 | No action |
| | | 1 | Start Condition detected |
| | | | Value after reset is low. |
| 5 | GC | | General call IRQ status. Set to '1' by core when General call address detected and interrupt signaled to MPUSS. Write '1' to clear. I2C mode only.<br>This read/clear only bit is set to 1 by the device if it detects the address of all zeros (8 bits) (general call). When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect.<br>**Note:** When this bit is set to 1, AAS also reads as 1. |
| | | 0 | No general call detected |
| | | 1 | General call address detected |
| | | | Value after reset is low. |
| 4 | XRDY | | Transmit data ready IRQ status. Set to '1' by core when transmitter and when new data is requested. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. Transmit mode only (I2C mode).<br>This read/clear only bit (XRDY) is set to 1 when the I2C peripheral is a master or slave transmitter, the CPU needs to send data through the I2C bus, and the module (transmitter) requires new data to be served. Note that a master transmitter requests new data if the FIFO TX level is below the threshold (TXTRSH) and the required amount of data remained to be transmitted (I2C_BUFSTAT.TXSTAT) is greater than the threshold. A slave transmitter requests new data when the FIFO TX level is below the threshold (if TXTRSH > 1), or anytime there is a read request from external master (for each acknowledge received from the master), if TXTRSH = 1.<br>When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can also poll this bit (refer to the FIFO Management subsection for details about XRDY generation). The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect.<br>**Note:** If the DMA transmit mode is enabled (I2C_BUF.XDMA_EN is set, together with I2C_DMATXENABLE_SET), this bit is forced to 0 and no interrupt will be generated; instead, a DMA TX request to the main DMA controller of the system is generated. |
| | | 0 | Transmission ongoing |
| | | 1 | Transmit data ready |
| | | | Value after reset is low. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-8. I2C_IRQSTATUS_RAW Register (I2C Status Raw) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 3 | RRDY | | Receive mode only (I2C mode).<br>This read/clear only RRDY is set to 1 when the RX FIFO level is above the configured threshold (RXTRSH). When this bit is set to 1 by the core, CPU is able to read new data from the I2C_DATA register. If the corresponding interrupt was enabled, an interrupt is signaled to the local host. The CPU to read the received data in I2C_DATA register can also poll this bit (refer to the FIFO Management subsection for details about RRDY generation).<br>The CPU can only clear this bit by writing a 1 into this register. A write 0 has no effect.<br>If the DMA receive mode is enabled (I2C_BUF.RDMA_EN is set, together with I2C_DMARXENABLE_SET), this bit is forced to 0 and no interrupt will be generated; instead a DMA RX request to the main DMA controller of the system is generated. |
| | | 0 | Receive FIFO threshold not reached |
| | | 1 | Receive data ready for read (RX FIFO threshold reached) |
| | | | Value after reset is low. |
| 2 | ARDY | | I2C mode only. This read/clear only bit, when set to 1, indicates that the previously programmed data and command (receive or transmit, master or slave) has been performed and status bit has been updated. The CPU uses this flag to let it know that the I2C registers are ready to be accessed again. The CPU can only clear this bit by writing a 1 into this register. A write 0 has no effect. |
| | | | **Mode** / **Others** / **ARDY Set Condition**:<br>I2C Master transmit / STP = 1 / DCOUNT = 0<br>I2C Master receive / STP = 1 / DCOUNT = 0 and receiver FIFO empty<br>I2C Master transmit / STP = 0 / DCOUNT passed 0<br>I2C Master receive / STP = 0 / DCOUNT passed 0 and receiver FIFO empty<br>I2C Master transmit / - / Stop or restart condition received from master<br>I2C Slave receive / - / Stop or restart condition and receiver FIFO empty |
| | | 0 | No action |
| | | 1 | Access ready |
| | | | Value after reset is low. |
| 1 | NACK | | No acknowledgement IRQ status. Bit is set when No Acknowledge has been received, an interrupt is signaled to MPUSS. Write '1' to clear this bit. I2C mode only.<br>The read/clear only No Acknowledge flag bit is set when the hardware detects No Acknowledge has been received. When a NACK event occurs on the bus, this bit is set to 1, the core automatically ends the transfer and clears the MST/STP bits in the I2C_CON register and the I2C becomes a slave. Clearing the FIFOs from remaining data might be required.<br>The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect. |
| | | 0 | Normal operation |
| | | 1 | Not Acknowledge detected |
| | | | Value after reset is low. |
| 0 | AL | | Arbitration lost IRQ status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to MPUSS. During reads, it always returns 0. I2C mode only.<br>The read/clear only Arbitration Lost flag bit is set to 1 when the device (configured in master mode) detects it has lost an arbitration (in Address Phase). This happens when two or more masters initiate a transfer on the I2C bus almost simultaneously or when the I2C attempts to start a transfer while BB (bus busy) is 1.<br>When this is set to 1 due to arbitration lost, the core automatically clears the MST/STP bits in the I2C_CON register and the I2C becomes a slave receiver.<br>The CPU can only clear this bit by writing a 1 to this register. Writing 0 has no effect. |
| | | 0 | Normal operation |
| | | 1 | Arbitration lost detected |
| | | | Value after reset is low. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.6 I2C_IRQSTATUS Register (I2C Status)

This register provides core status information for interrupt handling, showing all active and enabled events and masking the others. The fields are read-write. Writing a 1 to a bit will clear it to 0, that is, clear the IRQ. Writing a 0 will have no effect, that is, the register value will not be modified. Only enabled, active events will trigger an actual interrupt request on the IRQ output line.

For all the internal fields of the I2C_IRQSTATUS register, the descriptions given in the I2C_IRQSTATUS_RAW subsection are valid.

**Figure 15-19. I2C_IRQSTATUS Register (I2C Status)**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XDR | RDR | BB | ROVR | XUDF | AAS | BF |
| R-0 | R/W1C-0 | R/W1C-0 | R/W-0 | R/W-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AERR | STC | GC | XRDY | RRDY | ARDY | NACK | AL |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-9. I2C_IRQSTATUS Register (I2C Status) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Write 0s for future compatibility. Read returns 0. |
| 14 | XDR | | Transmit draining IRQ enabled status. |
| | | 0 | Transmit draining inactive |
| | | 1 | Transmit draining enabled |
| 13 | RDR | | Receive draining IRQ enabled status. |
| | | 0 | Receive draining inactive |
| | | 1 | Receive draining enabled |
| 12 | BB | | Bus busy enabled status.Writing into this bit has no effect. |
| | | 0 | Bus is free |
| | | 1 | Bus is occupied |
| 11 | ROVR | | Receive overrun enabled status.Writing into this bit has no effect. |
| | | 0 | Normal operation |
| | | 1 | Receiver overrun |
| 10 | XUDF | | Transmit underflow enabled status.Writing into this bit has no effect. |
| | | 0 | Normal operation |
| | | 1 | Transmit underflow |
| 9 | AAS | | Address recognized as slave IRQ enabled status. |
| | | 0 | No action |
| | | 1 | Address recognized |
| 8 | BF | | Bus Free IRQ enabled status. |
| | | 0 | No action |
| | | 1 | Bus free |
| 7 | AERR | | Access Error IRQ enabled status. |
| | | 0 | No action |
| | | 1 | Access error |
| 6 | STC | | Start Condition IRQ enabled status. |
| | | 0 | No action |
| | | 1 | Start condition detected |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-9. I2C_IRQSTATUS Register (I2C Status) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 5 | GC | | General call IRQ enabled status. Set to '1' by core when General call address detected and interrupt signaled to MPUSS. Write '1' to clear. |
| | | 0 | No general call detected |
| | | 1 | General call address detected |
| 4 | XRDY | | Transmit data ready IRQ enabled status. Set to '1' by core when transmitter and when new data is requested. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. |
| | | 0 | Transmision ongoing |
| | | 1 | Transmit data ready |
| 3 | RRDY | | Receive data ready IRQ enabled status. Set to '1' by core when receiver mode, a new data is able to be read. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. |
| | | 0 | No data available |
| | | 1 | Receive data available |
| 2 | ARDY | | Register access ready IRQ enabled status. When set to '1' it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to MPUSS. Write '1' to clear. |
| | | 0 | Module busy |
| | | 1 | Access ready |
| 1 | NACK | | No acknowledgement IRQ enabled status. Bit is set when No Acknowledge has been received, an interrupt is signaled to MPUSS. Write '1' to clear this bit. |
| | | 0 | Normal operation |
| | | 1 | Not Acknowledge detected |
| 0 | AL | | Arbitration lost IRQ enabled status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to MPUSS. During reads, it always returns 0. |
| | | 0 | Normal operation |
| | | 1 | Arbitration lost detected |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.7 I2C_IRQENABLE_SET Register (I2C Interrupt Enable Set)

All 1-bit fields enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit will enable the field. Writing a 0 will have no effect, that is, the register value will not be modified.

For all the internal fields of the I2C_IRQENABLE_SET register, the descriptions given in the I2C_IRQSTATUS_RAW subsection are valid.

**Figure 15-20. I2C_IRQENABLE_SET Register (I2C Interrupt Enable Set)**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XDR_IE | RDR_IE | Reserved | ROVR | XUDF | AAS_IE | BF_IE |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AERR_IE | STC_IE | GC_IE | XRDY_IE | RRDY_IE | ARDY_IE | NACK_IE | AL_IE |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-10. I2C_IRQENABLE_SET Register (I2C Interrupt Enable Set) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Write 0s for future compatibility. Read returns 0. |
| 14 | XDR_IE | | Transmit draining interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[XDR]. |
| | | 0 | Transmit draining interrupt disabled |
| | | 1 | Transmit draining interrupt enabled |
| 13 | RDR_IE | | Receive draining interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[RDR]. |
| | | 0 | Receive draining interrupt disabled |
| | | 1 | Receive draining interrupt enabled |
| 12 | Reserved | 0 | Reserved |
| 11 | ROVR | | Receive overrun enable set. |
| | | 0 | Receive overrun interrupt disabled |
| | | 1 | Receive draining interrupt enabled |
| 10 | XUDF | | Transmit underflow enable set. |
| | | 0 | Transmit underflow interrupt disabled |
| | | 1 | Transmit underflow interrupt enabled |
| 9 | AAS_IE | | Addressed as slave interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[AAS]. |
| | | 0 | Addressed as slave interrupt disabled |
| | | 1 | Addressed as slave interrupt enabled |
| 8 | BF_IE | | Bus free interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[BF]. |
| | | 0 | Bus free interrupt disabled |
| | | 1 | Bus free interrupt enabled |
| 7 | AERR_IE | | Access error interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[AERR]. |
| | | 0 | Access error interrupt disabled |
| | | 1 | Access error interrupt enabled |
| 6 | STC_IE | | Start condition interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[STC]. |
| | | 0 | Start condition interrupt disabled |
| | | 1 | Start condition interrupt enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-10. I2C_IRQENABLE_SET Register (I2C Interrupt Enable Set) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 5 | GC_IE | | General call interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[GC]. |
| | | 0 | General call interrupt disabled |
| | | 1 | General call interrupt enabled |
| 4 | XRDY_IE | | Transmit data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[XRDY]. |
| | | 0 | Transmit data ready interrupt disabled |
| | | 1 | Transmit data ready interrupt enabled |
| 3 | RRDY_IE | | Receive data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[RRDY] |
| | | 0 | Receive data ready interrupt disabled |
| | | 1 | Receive data ready interrupt enabled |
| 2 | ARDY_IE | | Register access ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[ARDY]. |
| | | 0 | Register access ready interrupt disabled |
| | | 1 | Register access ready interrupt enabled |
| 1 | NACK_IE | | No acknowledgement interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[NACK]. |
| | | 0 | Not Acknowledge interrupt disabled |
| | | 1 | Not Acknowledge interrupt enabled |
| 0 | AL_IE | | Arbitration lost interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[AL]. |
| | | 0 | Arbitration lost interrupt disabled |
| | | 1 | Arbitration lost interrupt enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.8 I2C_IRQENABLE_CLR Register (I2C Interrupt Enable Clear)

All 1-bit fields clear a specific interrupt event. Writing a 1 to a bit will disable the interrupt field. Writing a 0 will have no effect, that is, the register value will not be modified.

For all the internal fields of the I2C_IRQENABLE_CLR register, the descriptions given in the I2C_IRQSTATUS_RAW subsection are valid.

**Figure 15-21. I2C_IRQENABLE_CLR Register (I2C Interrupt Enable Clear)**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | XDR_IE | RDR_IE | Reserved | ROVR | XUDF | AAS_IE | BF_IE |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| AERR_IE | STC_IE | GC_IE | XRDY_IE | RRDY_IE | ARDY_IE | NACK_IE | AL_IE |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-11. I2C_IRQENABLE_CLR Register (I2C Interrupt Enable Clear) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Write 0s for future compatibility. Read returns 0. |
| 14 | XDR_IE | | Transmit draining interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[XDR]. |
| | | 0 | Transmit draining interrupt disabled |
| | | 1 | Transmit draining interrupt enabled |
| 13 | RDR_IE | | Receive draining interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[RDR]. |
| | | 0 | Receive draining interrupt disabled |
| | | 1 | Receive draining interrupt enabled |
| 12 | Reserved | 0 | Reserved |
| 11 | ROVR | | Receive overrun enable clear. |
| | | 0 | Receive overrun interrupt disabled |
| | | 1 | Receive draining interrupt enabled |
| 10 | XUDF | | Transmit underflow enable clear. |
| | | 0 | Transmit underflow interrupt disabled |
| | | 1 | Transmit underflow interrupt enabled |
| 9 | AAS_IE | | Addressed as slave interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[AAS]. |
| | | 0 | Addressed as slave interrupt disabled |
| | | 1 | Addressed as slave interrupt enabled |
| 8 | BF_IE | | Bus Free interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[BF]. |
| | | 0 | Bus free interrupt disabled |
| | | 1 | Bus free interrupt enabled |
| 7 | AERR_IE | | Access error interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[AERR]. |
| | | 0 | Access error interrupt disabled |
| | | 1 | Access error interrupt enabled |
| 6 | STC_IE | | Start condition interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[STC]. |
| | | 0 | Start condition interrupt disabled |
| | | 1 | Start condition interrupt enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-11. I2C_IRQENABLE_CLR Register (I2C Interrupt Enable Clear) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 5 | GC_IE | | General call interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[GC]. |
| | | 0 | General call interrupt disabled |
| | | 1 | General call interrupt enabled |
| 4 | XRDY_IE | | Transmit data ready interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[XRDY]. |
| | | 0 | Transmit data ready interrupt disabled |
| | | 1 | Transmit data ready interrupt enabled |
| 3 | RRDY_IE | | Receive data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[RRDY] |
| | | 0 | Receive data ready interrupt disabled |
| | | 1 | Receive data ready interrupt enabled |
| 2 | ARDY_IE | | Register access ready interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[ARDY]. |
| | | 0 | Register access ready interrupt disabled |
| | | 1 | Register access ready interrupt enabled |
| 1 | NACK_IE | | No acknowledgement interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[NACK]. |
| | | 0 | Not Acknowledge interrupt disabled |
| | | 1 | Not Acknowledge interrupt enabled |
| 0 | AL_IE | | Arbitration lost interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[AL]. |
| | | 0 | Arbitration lost interrupt disabled |
| | | 1 | Arbitration lost interrupt enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.9 I2C_WE Register (I2C Wakeup Enable)

Every 1-bit field in the I2C_WE register enables a specific (synchronous) IRQ request source to generate an asynchronous wakeup (on the appropriate swakeup line). When a bit location is set to 1 by the local host, a wakeup is signaled to the local host if the corresponding event is captured by the core of the I2C controller. Value after reset is low (all bits).

**NOTE:**

- There is no need for an Access Error WakeUp event, since this event occurs only when the module is in Active Mode (for Interface/OCP accesses to FIFO) and is signaled by an interrupt.
- With the exception of Start Condition WakeUp, which is asynchronously detected when the Functional clock is turned-off, all the other WakeUp events require the Functional (System) clock to be enabled.

**Figure 15-22. I2C_WE Register (I2C Wakeup Enable)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XDR_WE | RDR_WE | Reserved | ROVR_WE | XUDF_WE | AAS_WE | BF_WE |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | STC_WE | GC_WE | Reserved | DRDY_WE | ARDY_WE | NACK_WE | AL_WE |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-12. I2C_WE Register (I2C Wakeup Enable) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved |
| 14 | XDR_WE | | Transmit draining wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode, the TX FIFO level is below the threshold and the amount of data left to be transferred is less than TXTRSH value. This allows for the module to inform the CPU that it can check the amount of data to be written to the FIFO. |
| | | 0 | Transmit draining wakeup disabled |
| | | 1 | Transmit draining wakeup enabled |
| 13 | RDR_WE | | Receive draining wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C is in idle mode, configured as a receiver, and it has detected a stop condition on the bus but the RX FIFO threshold is not reached (but the FIFO is not empty). This allows for the module to inform the CPU that it can check the amount of data to be transferred from the FIFO. |
| | | 0 | Receive draining wakeup disabled |
| | | 1 | Receive draining wakeup enabled |
| 12 | Reserved | 0 | Reserved |
| 11 | ROVR_WE | | Receive overrun wakeup enable |
| | | 0 | Receive overrun wakeup disabled |
| | | 1 | Receive overrun wakeup enabled |
| 10 | XUDF_WE | | Transmit underflow wakeup enable |
| | | 0 | Transmit underflow wakeup disabled |
| | | 1 | Transmit underflow wakeup enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-12. I2C_WE Register (I2C Wakeup Enable) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 9 | AAS_WE | | Address as slave IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode, and external master addresses the I2C module as a slave. This allows for the module to inform the CPU that it can check which of the own addresses was used by the external master to access the I2C core. |
| | | 0 | Addressed as slave wakeup disabled |
| | | 1 | Addressed as slave wakeup enabled |
| 8 | BF_WE | | Bus free IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode and the I2C bus became free. This allows for the module to inform the CPU that it can initiate its own transfer on the I2C line. |
| | | 0 | Bus Free wakeup disabled |
| | | 1 | Bus Free wakeup enabled |
| 7 | Reserved | 0 | Reserved |
| 6 | STC_WE | | Start condition IRQ wakeup set. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode (with the functional clock inactive) and a possible start condition is detected on the I2C line. The STC WakeUp is generated only if the I2C_SYSC.ClockActivity field indicates that the functional clock can be disabled. Note that if the functional clock is not active, the start condition is asynchronously detected (no filtering and synchronization is used). For this reason, it is possible that the signalized start condition to be a glitch.<br>If the functional clock cannot be disabled (I2C_SYSC.ClockActivity = 10 or 11), the programmer should not enable this wakeup, since the module has other synchronously detected WakeUp event that might be used to exit from idle mode, only if the detected transfer is accessing the I2C module. |
| | | 0 | Start condition wakeup disabled |
| | | 1 | Start condition wakeup enabled |
| 5 | GC_WE | | General call IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode and a general call is received on I2C line. |
| | | 0 | General call wakeup disabled |
| | | 1 | General call wakeup enabled |
| 4 | Reserved | 0 | Reserved |
| 3 | DRDY_WE | | Receive/Transmit data ready IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is involved into a long transfer and no more registers accesses are performed on the interface (for example module are set in F/S I2C master transmitter mode and FIFO is full). If in the middle of such a transaction, the FIFO buffer needs more data to be transferred, CPU must be informed to write (in case of transmitter mode) or read (if receiver mode) in/from the FIFO. |
| | | 0 | Transmit/receive data ready wakeup disabled |
| | | 1 | Transmit/receive data ready wakeup enabled |
| 2 | ARDY_WE | | Register access ready IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is involved into a long transfer and no more registers accesses are performed on the interface (for example the module is set in F/S I2C master transmitter mode and FIFO is full). If the current transaction is finished, the module needs to inform CPU about transmission completion. |
| | | 0 | Register access ready wakeup disabled |
| | | 1 | Register access ready wakeup enabled |
| 1 | NACK_WE | | No acknowledgment IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is involved into a long transfer and no more registers accesses are performed on the interface (for example the module is set in F/S I2C master transmitter mode and FIFO is full). If in the middle of such of a transaction a Not Acknowledgment event is raised, the module needs to inform CPU about transmission error. |
| | | 0 | Not Acknowledge wakeup disabled |
| | | 1 | Not Acknowledge wakeup enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-12. I2C_WE Register (I2C Wakeup Enable) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | AL_WE | | Arbitration lost IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is configured as a master and it loses the arbitration. This wake up is very useful when the module is configured as a master transmitter, all the necessary data is provided in the FIFO Tx, STT is enabled and the module enters in Idle Mode. If the module loses the arbitration, an Arbitration Lost event is raised and the module needs to inform CPU about transmission error.<br>**Note:** The AL wakeup must be enabled only for multimaster communication. If the AL_WE is not enabled and the scenario described above occurs, the module will not be able to inform the CPU about the state of the transfer and it will be blocked in an undetermined state. |
| | | 0 | Arbitration lost wakeup disabled |
| | | 1 | Arbitration lost wakeup enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.10 I2C_DMARXENABLE_SET Register (Receive DMA Enable Set)

The 1-bit field enables a receive DMA request. Writing a 1 to this field will set it to 1. Writing a 0 will have no effect, that is, the register value is not modified. Note that the I2C_BUF.RDMA_EN field is the global (slave) DMA enabler, and that it is disabled by default. The I2C_BUF.RDMA_EN field should also be set to 1 to enable a receive DMA request.

**Figure 15-23. I2C_DMARXENABLE_SET Register (Receive DMA Enable Set)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | DMARX_ENABLE_SET |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-13. I2C_DMARXENABLE_SET Register (Receive DMA Enable Set) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | DMARX_ENABLE_SET | 0-1 | Receive DMA channel enable set. |

### 15.3.11 I2C_DMATXENABLE_SET Register (Transmit DMA Enable Set)

The 1-bit field enables a transmit DMA request. Writing a 1 to this field will set it to 1. Writing a 0 will have no effect, that is, the register value is not modified. Note that the I2C_BUF.XDMA_EN field is the global (slave) DMA enabler, and that it is disabled by default. The I2C_BUF.XDMA_EN field should also be set to 1 to enable a transmit DMA request.

**Figure 15-24. I2C_DMATXENABLE_SET Register (Transmit DMA Enable Set)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | DMATX_ENABLE_SET |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-14. I2C_DMATXENABLE_SET Register (Transmit DMA Enable Set) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | DMATX_ENABLE_SET | 0-1 | Transmit DMA channel enable set. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.12 I2C_DMARXENABLE_CLR Register (Receive DMA Enable Clear)

The 1-bit field disables a receive DMA request. Writing a 1 to a bit will clear it to 0. Another result of setting to '1' the DMARX_ENABLE_CLEAR field, is the reset of the DMA RX request and wakeup lines. Writing a 0 will have no effect, that is, the register value is not modified.

**Figure 15-25. I2C_DMARXENABLE_CLR Register (Receive DMA Enable Clear)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | DMARX_ENABLE_CLEAR |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-15. I2C_DMARXENABLE_CLR Register (Receive DMA Enable Clear) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | DMARX_ENABLE_CLEAR | 0-1 | Receive DMA channel enable clear. |

### 15.3.13 I2C_DMATXENABLE_CLR Register (Transmit DMA Enable Clear)

The 1-bit field disables a transmit DMA request. Writing a 1 to a bit will clear it to 0. Another result of setting to '1' the DMATX_ENABLE_CLEAR field, is the reset of the DMA TX request and wakeup lines. Writing a 0 will have no effect, that is, the register value is not modified.

**Figure 15-26. I2C_DMATXENABLE_CLR Register (Transmit DMA Enable Clear)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | DMATX_ENABLE_CLEAR |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-16. I2C_DMATXENABLE_CLR Register (Transmit DMA Enable Clear) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | DMATX_ENABLE_CLEAR | 0-1 | Transmit DMA channel enable clear. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.14 I2C_DMARXWAKE_EN Register (Receive DMA Wakeup)

All 1-bit fields enable a specific (synchronous) DMA request source to generate an asynchronous wakeup (on the appropriate swakeup line). Note that the I2C_SYSC.ENAWAKEUP field is the global (slave) wakeup enabler, and that it is disabled by default.

**Figure 15-27. I2C_DMARXWAKE_EN Register (Receive DMA Wakeup)**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XDR | RDR | Reserved | ROVR | XUDF | AAS | BF |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | STC | GC | Reserved | DRDY | ARDY | NACK | AL |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 15-17. I2C_DMARXWAKE_EN Register (Receive DMA Wakeup) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved |
| 14 | XDR | | Transmit draining wakeup set. |
| | | 0 | Transmit draining interrupt disabled |
| | | 1 | Transmit draining interrupt enabled |
| 13 | RDR | | Receive draining wakeup set. |
| | | 0 | Receive draining interrupt disabled |
| | | 1 | Receive draining interrupt enabled |
| 12 | Reserved | 0 | Reserved |
| 11 | ROVR | | Receive overrun wakeup set. |
| | | 0 | Receive overrun interrupt disabled |
| | | 1 | Receive draining interrupt enabled |
| 10 | XUDF | | Transmit underflow wakeup set. |
| | | 0 | Transmit underflow interrupt disabled |
| | | 1 | Transmit underflow interrupt enabled |
| 9 | AAS | | Address as slave IRQ wakeup set. |
| | | 0 | Addressed as slave interrupt disabled |
| | | 1 | Addressed as slave interrupt enabled |
| 8 | BF | | Bus free IRQ wakeup set. |
| | | 0 | Bus free wakeup disabled |
| | | 1 | Bus free wakeup enabled |
| 7 | Reserved | 0 | Reserved |
| 6 | STC | | Start condition IRQ wakeup set. |
| | | 0 | Start condition wakeup disabled |
| | | 1 | Start condition wakeup enabled |
| 5 | GC | | General call IRQ wakeup set. |
| | | 0 | General call wakeup disabled |
| | | 1 | General call wakeup enabled |
| 4 | Reserved | 0 | Reserved |
| 3 | DRDY | | Receive/transmit data ready IRQ wakeup set. |
| | | 0 | Transmit/receive data ready wakeup disabled |
| | | 1 | Transmit/receive data ready wakeup enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-17. I2C_DMARXWAKE_EN Register (Receive DMA Wakeup) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 2 | ARDY | | Register access ready IRQ wakeup set. |
| | | 0 | Register access ready wakeup disabled |
| | | 1 | Register access ready wakeup enabled |
| 1 | NACK | | No acknowledgment IRQ wakeup set. |
| | | 0 | Not Acknowledge wakeup disabled |
| | | 1 | Not Acknowledge wakeup enabled |
| 0 | AL | | Arbitration lost IRQ wakeup set. |
| | | 0 | Arbitration lost wakeup disabled |
| | | 1 | Arbitration lost wakeup enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.15 I2C_DMATXWAKE_EN Register (Transmit DMA Wakeup)

All 1-bit fields enable a specific (synchronous) DMA request source to generate an asynchronous wakeup (on the appropriate swakeup line). Note that the I2C_SYSC.ENAWAKEUP field is the global (slave) wakeup enabler, and that it is disabled by default.

**Figure 15-28. I2C_DMATXWAKE_EN Register (Transmit DMA Wakeup)**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XDR | RDR | Reserved | ROVR | XUDF | AAS | BF |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | STC | GC | Reserved | DRDY | ARDY | NACK | AL |
| R-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 15-18. I2C_DMATXWAKE_EN Register (Transmit DMA Wakeup) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved |
| 14 | XDR | | Transmit draining wakeup set. |
| | | 0 | Transmit draining interrupt disabled |
| | | 1 | Transmit draining interrupt enabled |
| 13 | RDR | | Receive draining wakeup set. |
| | | 0 | Receive draining interrupt disabled |
| | | 1 | Receive draining interrupt enabled |
| 12 | Reserved | 0 | Reserved |
| 11 | ROVR | | Receive overrun wakeup set. |
| | | 0 | Receive overrun interrupt disabled |
| | | 1 | Receive draining interrupt enabled |
| 10 | XUDF | | Transmit underflow wakeup set. |
| | | 0 | Transmit underflow interrupt disabled |
| | | 1 | Transmit underflow interrupt enabled |
| 9 | AAS | | Address as slave IRQ wakeup set. |
| | | 0 | Addressed as slave interrupt disabled |
| | | 1 | Addressed as slave interrupt enabled |
| 8 | BF | | Bus free IRQ wakeup set. |
| | | 0 | Bus free wakeup disabled |
| | | 1 | Bus free wakeup enabled |
| 7 | Reserved | 0 | Reserved |
| 6 | STC | | Start condition IRQ wakeup set. |
| | | 0 | Start condition wakeup disabled |
| | | 1 | Start condition wakeup enabled |
| 5 | GC | | General call IRQ wakeup set. |
| | | 0 | General call wakeup disabled |
| | | 1 | General call wakeup enabled |
| 4 | Reserved | 0 | Reserved |
| 3 | DRDY | | Receive/transmit data ready IRQ wakeup set. |
| | | 0 | Transmit/receive data ready wakeup disabled |
| | | 1 | Transmit/receive data ready wakeup enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-18. I2C_DMATXWAKE_EN Register (Transmit DMA Wakeup) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 2 | ARDY | | Register access ready IRQ wakeup set. |
| | | 0 | Register access ready wakeup disabled |
| | | 1 | Register access ready wakeup enabled |
| 1 | NACK | | No acknowledgment IRQ wakeup set. |
| | | 0 | Not Acknowledge wakeup disabled |
| | | 1 | Not Acknowledge wakeup enabled |
| 0 | AL | | Arbitration lost IRQ wakeup set. |
| | | 0 | Arbitration lost wakeup disabled |
| | | 1 | Arbitration lost wakeup enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.16 I2C_SYSS Register (System Status)

**Figure 15-29. I2C_SYSS Register (System Status)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | RDONE |
| R-0 | | R/W-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-19. I2C_SYSS Register (System Status) Field Descriptions**

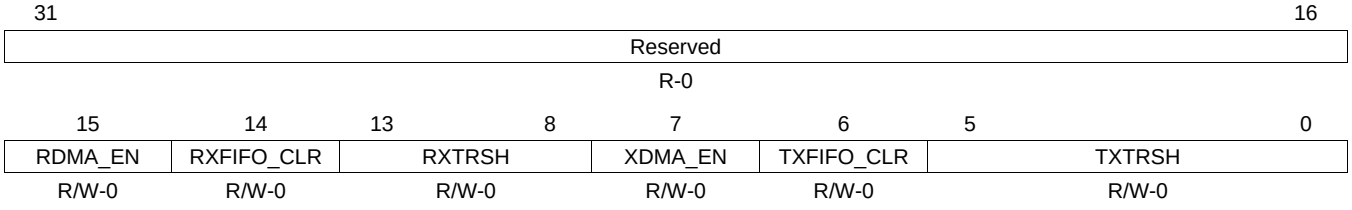
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | RDONE | | Reset done bit. This read-only bit indicates the state of the reset in case of hardware reset, global software reset (I2C_SYSC.SRST) or partial software reset (I2C_CON.I2C_EN).<br>The module must receive all its clocks before it can grant a reset-completed status. |
| | | 0 | Internal module reset in ongoing |
| | | 1 | Reset completed |
| | | | Value after reset is low. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.17 I2C_BUF Register (Buffer Configuration)

This read/write register enables DMA transfers and allows the configuration of FIFO thresholds for the FIFO management (see the FIFO Management subsection).

**Figure 15-30. I2C_BUF Register (Buffer Configuration)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-20. I2C_BUF Register (Buffer Configuration) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15 | RDMA_EN | | Receive DMA channel enable. When this bit is set to 1, the receive DMA channel is enabled and the receive data ready status bit (I2C_IRQSTATUS_RAW: RRDY) is forced to 0 by the core. |
| | | 0 | Receive DMA channel disabled |
| | | 1 | Receive DMA channel enabled |
| | | | Value after reset is low. |
| 14 | RXFIFO_CLR | | Receive FIFO clear. When set, receive FIFO is cleared (hardware reset for RX FIFO generated). This bit is automatically reset by the hardware. During reads, it always returns 0. |
| | | 0 | Normal mode |
| | | 1 | Rx FIFO is reset |
| | | | Value after reset is low. |
| 13-8 | RXTRSH | 0-3Fh | Threshold value for FIFO buffer in RX mode. The receive threshold value is used to specify the trigger level for data receive transfers. The value is specified from the Interface/OCP point of view. |
| | | 0 | Receive Threshold value = 1 |
| | | 1 | Receive Threshold value = 2 |
| | | - | |
| | | - | |
| | | 3Fh | Receive Threshold value = 64 |
| | | | Value after reset is 00h. |
| | | | For the FIFO management description, see the FIFO Management subsection. |
| | | | **Note1:** programmed threshold cannot exceed the actual depth of the FIFO. |
| | | | **Note2:** the threshold must not be changed while a transfer is in progress (after STT was configured or after the module was addressed as a slave). |
| 7 | XDMA_EN | | Transmit DMA channel enable. When this bit is set to 1, the transmit DMA channel is enabled and the transmit data ready status (I2C_IRQSTATUS_RAW: XRDY) bit is forced to 0 by the core. |
| | | 0 | Transmit DMA channel disabled |
| | | 1 | Transmit DMA channel enabled |
| | | | Value after reset is low. |
| 6 | TXFIFO_CLR | | Transmit FIFO clear. When set, transmit FIFO is cleared (hardware reset for TX FIFO). This bit is automatically reset by the hardware. During reads, it always returns 0. |
| | | 0 | Normal mode |
| | | 1 | Tx FIFO is reset |
| | | | Value after reset is low. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-20. I2C_BUF Register (Buffer Configuration) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 5-0 | TXTRSH | 0-3Fh | Threshold value for FIFO buffer in TX mode. The Transmit Threshold value is used to specify the trigger level for data transfers. The value is specified from the OCP point of view. |
| | | 0 | Transmit Threshold value = 1 |
| | | 1 | Transmit Threshold value = 2 |
| | | - | |
| | | - | |
| | | 3Fh | Transmit Threshold value = 64 |
| | | | Value after reset is 00h |
| | | | **Note1**: programmed threshold cannot exceed the actual depth of the FIFO. |
| | | | **Note2:** the threshold must not be changed while a transfer is in progress (after STT was configured or after the module was addressed as a slave). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.18 I2C_CNT Register (Data Counter)

**CAUTION**

During an active transfer phase (between STT having been set to 1 and reception of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This read/write register is used to control the numbers of bytes in the I2C data payload.

**Figure 15-31. I2C_CNT Register (Data Counter)**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | DCOUNT | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-21. I2C_CNT Register (Data Counter) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15-0 | DCOUNT | 0-FFFFh | Data count. I2C Master Mode only (receive or transmit; F/S).<br>This 16-bit countdown counter decrements by 1 for every byte received or sent through the I2C interface. A write initializes DCOUNT to a saved initial value. A read returns the number of bytes that are yet to be received or sent. A read into DCOUNT returns the initial value only before a start condition and after a stop condition.<br>When DCOUNT reaches 0, the core generates a stop condition if a stop condition was specified (I2C_CON.STP = 1) and the ARDY status flag is set to 1 in the I2C_IRQSTATUS_RAW register.<br>Note that DCOUNT must not be reconfigured after I2C_CON.STT was enabled and before ARDY is received.<br>**Note1:** In case of I2C mode of operation, if I2C_CON.STP = 0, then the I2C asserts SCL = 0 when DCOUNT reaches 0. The CPU can then reprogram DCOUNT to a new value and resume sending or receiving data with a new start condition (restart). This process repeats until the CPU sets to 1 the I2C_CON.STP bit.<br>The ARDY flag is set each time DCOUNT reaches 0 and DCOUNT is reloaded to its initial value. |
| | | 0 | Data counter = 65536 bytes (2^16) |
| | | 1 | Data counter = 1 bytes |
| | | - | |
| | | - | |
| | | FFFFh | Data counter = 65535 bytes (2^16 - 1) |
| | | | Values after reset are low (all 16 bits).<br>**Note2:** Since for DCOUNT = 0, the transfer length is 65536, the module does not allow the possibility to initiate zero data bytes transfers. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.19 I2C_DATA Register (Data Access)

This register is the entry point for the local host to read data from or write data to the FIFO buffer.

**Figure 15-32. I2C_DATA Register (Data Access)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | DATA | |
| R-0 | | R/W-x | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 15-22. I2C_DATA Register (Data Access) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | DATA | 0-FFh | Transmit/Receive data FIFO endpoint. When read, this register contains the received I2C data. When written, this register contains the byte value to transmit over the I2C data. In SYSTEST loop back mode (I2C_SYSTEST: TMODE = 11), this register is also the entry/receive point for the data.<br>Values after reset are unknown (all 8-bits).<br>**Note:** A read access, when the buffer is empty, returns the previous read data value. A write access, when the buffer is full, is ignored. In both events, the FIFO pointers are not updated and an Access Error (AERR) Interrupt is generated. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.20 I2C_CON Register (I2C Configuration)

**CAUTION**

During an active transfer phase (between STT having been set to 1 and reception of ARDY), no modification must be done in this register (except STP enable). Changing it may result in an unpredictable behavior.

**Figure 15-33. I2C_CON Register (I2C Configuration)**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| I2C_EN | Reserved | OPMODE | | STB | MST | TRX | XSA |
| R/W-0 | R-0 | R/W-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| XOA0 | XOA01 | XOA02 | XOA3 | Reserved | | STP | STT |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | | R/W-0 | R/W-0 |

LEGEND: RR/W-0/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-23. I2C_CON Register (I2C Configuration) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15 | I2C_EN | | I2C module enable. When this bit is cleared to 0, the I2C controller is not enabled and reset. When 0, receive and transmit FIFOs are cleared and all status bits are set to their default values. All configuration registers (I2C_IRQENABLE_SET, I2C_IRQWAKE_SET, I2C_BUF, I2C_CNT, I2C_CON, I2C_OA, I2C_SA, I2C_PSC, I2C_SCLL and I2C_SCLH) are not reset, they keep their initial values and can be accessed. The CPU must set this bit to 1 for normal operation. |
| | | 0 | Controller in reset. FIFO are cleared and status bits are set to their default value. |
| | | 1 | Module enabled |
| | | | Value after reset is low. |
| 14 | Reserved | 0 | Reserved |
| 13-12 | OPMODE | | Operation mode selection. These two bits select module operation mode. |
| | | 0h | I2C Fast/Standard mode |
| | | 1h | Reserved |
| | | 2h | Reserved |
| | | 3h | Reserved |
| | | | Value after reset is 00. |
| 11 | STB | | Start byte mode (I2C master mode only). The start byte mode bit is set to 1 by the CPU to configure the I2C in start byte mode (I2C_SA = 0000 0001). See the Philips I2C spec for more details [1]. |
| | | 0 | Normal mode |
| | | 1 | Start byte mode |
| | | | Value after reset is low. |
| 10 | MST | | Master/slave mode (I2C mode only). When this bit is cleared, the I2C controller is in the slave mode and the serial clock (SCL) is received from the master device. When this bit is set, the I2C controller is in the master mode and generates the serial clock. |
| | | | **Note:** This bit is automatically cleared at the end of the transfer on a detected stop condition, in case of arbitration lost or when the module is configured as a master but addressed as a slave by an external master. |
| | | 0 | Slave mode |
| | | 1 | Master mode |
| | | | Value after reset is low. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-23. I2C_CON Register (I2C Configuration) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 9 | TRX | | Transmitter/receiver mode (i2C master mode only). When this bit is cleared, the I2C controller is in the receiver mode and data on data line SDA is shifted into the receiver FIFO and can be read from I2C_DATA register. When this bit is set, the I2C controller is in the transmitter mode and the data written in the transmitter FIFO via I2C_DATA is shifted out on data line SDA. |
| | | 0 | Receiver mode |
| | | 1 | Transmitter mode |
| | | | Value after reset is low. |
| | | | The operating modes are defined as follows: |
| | | | **MST** / **TRX** / **Operating Modes**: 0 / x / Slave receiver; 0 / x / Slave transmitter; 1 / 0 / Master receiver; 1 / 1 / Master transmitter |
| 8 | XSA | | Expand slave address. (I2C mode only). When set, this bit expands the slave address to 10-bit. |
| | | 0 | 7-bit address mode |
| | | 1 | 10-bit address mode |
| | | | Value after reset is low. |
| 7 | XOA0 | | Expand own address 0. (I2C mode only). When set, this bit expands the base own address (OA0) to 10-bit. |
| | | 0 | 7-bit address mode |
| | | 1 | 10-bit address mode |
| | | | Value after reset is low. |
| 6 | XOA1 | | Expand own address 1. (I2C mode only). When set, this bit expands the first alternative own address (OA1) to 10-bit. |
| | | 0 | 7-bit address mode |
| | | 1 | 10-bit address mode |
| | | | Value after reset is low. |
| 5 | XOA2 | | Expand own address 2. (I2C mode only). When set, this bit expands the second alternative own address (OA2) to 10-bit. |
| | | 0 | 7-bit address mode. |
| | | 1 | 10-bit address mode |
| | | | Value after reset is low. |
| 4 | XOA3 | | Expand own address 3. When set, this bit expands the third alternative own address (OA3) to 10-bit. |
| | | 0 | 7-bit address mode |
| | | 1 | 10-bit address mode |
| | | | Value after reset is low. |
| 3-2 | Reserved | 0 | Reserved |
| 1 | STP | | Stop condition (I2C master mode only). This bit can be set to a 1 by the CPU to generate a stop condition. It is reset to 0 by the hardware after the stop condition has been generated. The stop condition is generated when DCOUNT passes 0. |
| | | | When this bit is not set to 1 before the end of the transfer (DCOUNT = 0), the stop condition is not generated and the SCL line is hold to 0 by the master, which can re-start a new transfer by setting the STT bit to 1. |
| | | 0 | No action or stop condition detected |
| | | 1 | Stop condition queried |
| | | | Value after reset is low |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-23. I2C_CON Register (I2C Configuration) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | STT | | Start condition (I2C master mode only). This bit can be set to a 1 by the CPU to generate a start condition. It is reset to 0 by the hardware after the start condition has been generated. The start/stop bits can be configured to generate different transfer formats. |
| | | 0 | No action or start condition detected |
| | | 1 | Start condition queried |
| | | | Value after reset is low. |

| STT | STP | Conditions | Bus Activities |
|---|---|---|---|
| 1 | 0 | Start | S-A-D |
| 0 | 1 | Stop | P |
| 1 | 1 | Start-Stop (DCOUNT= n) | S-A-D..(n)..D-P |
| 1 | 0 | Start (DCOUNT= n) | S-A-D..(n)..D |

**Note:** DCOUNT is data count value in I2C_CNT register.

### 15.3.21 I2C_OA Register (I2C Own Address)

**CAUTION**

During an active transfer phase (between STT having been set to 1 and reception of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the module's base I2C 7-bit or 10-bit address (base own address).

**Figure 15-34. I2C_OA Register (I2C Own Address)**

| 31 | 10 | 9 | 0 |
|---|---|---|---|
| Reserved | | OA | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 15-24. I2C_OA Register (I2C Own Address) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9-0 | OA | 0-3FFh | Own address. This field specifies either:<br>• A 10-bit address coded on OA [9:0] when XOA (Expand Own Address, I2C_CON[7]) is set to 1.<br>• A 7-bit address coded on OA [6:0] when XOA (Expand Own Address, I2C_CON[7]) is cleared to 0. In this case, OA [9:7] bits must be cleared to 000 by application software.<br>Value after reset is low (all 10 bits). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.22 I2C_SA Register (I2C Slave Address)

**CAUTION**

During an active transfer phase (between STT having been set to 1 and reception of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the addressed I2C module 7-bit or 10-bit address (slave address).

**Figure 15-35. I2C_SA Register (I2C Own Address)**

| 31 | 10 | 9 | 0 |
|---|---|---|---|
| Reserved | | SA | |
| R-0 | | R/W-3FFh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-25. I2C_SA Register (I2C Slave Address) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9-0 | SA | 0-3FFh | Slave address. This field specifies either:<br>• A 10-bit address coded on SA [9:0] when XSA (Expand Slave Address, I2C_CON[8]) is set to 1.<br>• A 7-bit address coded on SA [6:0] when XSA (Expand Slave Address, I2C_CON[8]) is cleared to 0. In this case, SA [9:7] bits must be cleared to 000 by application software.<br>Value after reset is low (all 10 bits). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.23 I2C_PSC Register (I2C Clock Prescaler)

**CAUTION**

During an active mode (I2C_EN bit in I2C_CON register is set to 1), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the internal clocking of the I2C peripheral core.

**Figure 15-36. I2C_PSC Register (I2C Own Address)**

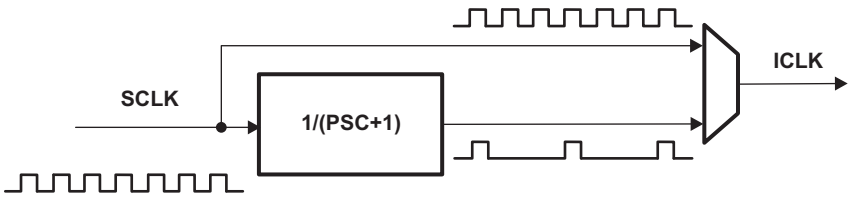
| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | PSC | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-26. I2C_PSC Register (I2C Clock Prescaler) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | PSC | 0-FFh | Fast/Standard mode prescale sampling clock divider value.<br>The core uses this 8-bit value to divide the system clock (SCLK) and generates its own internal sampling clock (ICLK) for Fast and Standard operation modes. The core logic is sampled at the clock rate of the system clock for the module divided by (PSC + 1).<br>0h Divide by 1<br>1h Divide by 2<br>-<br>-<br>FFh Divide by 256<br>Value after reset is low (all 8 bits). |

**Figure 15-37. Clock Divider**





Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.24 I2C_SCLL Register (I2C SCL Low Time)

**CAUTION**

During an active mode (I2C_EN bit in I2C_CON register is set to 1), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to determine the SCL low time value when master.

**Figure 15-38. I2C_SCLL Register (I2C SCL Low Time)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | SCLL | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-27. I2C_SCLL Register (I2C SCL Low Time) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | SCLL | 0-FFh | Fast/Standard mode SCL low time. I2C master mode only (FS).<br>This 8-bit value is used to generate the SCL low time value (tLOW) when the peripheral is operated in master mode.<br>$t_{LOW}$ = (SCLL + 7) * $I_{CLK}$ time period,<br>Value after reset is low (all 8 bits). |

### 15.3.25 I2C_SCLH Register (I2C SCL High Time)

**CAUTION**

During an active mode (I2C_EN bit in I2C_CON register is set to 1), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to determine the SCL high time value when master.

**Figure 15-39. I2C_SCLH Register (I2C SCL High Time)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | SCLH | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-28. I2C_SCLH Register (I2C SCL High Time) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | SCLH | 0-FFh | Fast/Standard mode SCL low time. I2C master mode only (FS).<br>This 8-bit value is used to generate the SCL high time value (tHIGH) when the peripheral is operated in master mode.<br>$t_{HIGH}$ = (SCLH + 5 + floor[t_rise/$I_{CLK}$ Period]) * $I_{CLK}$ time period.<br>Value after reset is low (all 8 bits). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.26 I2C_SYSTEST Register (System Test)

**CAUTION**

Never enable this register for normal I2C operation

This register is used to facilitate system-level tests by overriding some of the standard functional features of the peripheral. It allows testing of SCL counters, controlling the signals that connect to I/O pins, or creating digital loop-back for self-test when the module is configured in system test (SYSTEST) mode. It also provides stop/non-stop functionality in the debug mode.

**Figure 15-40. I2C_SYSTEST Register (System Test)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| ST_EN | FREE | TMODE | | SSB | Reserved | | SCL_I_FUNC |
| R/W-0 | R/W-0 | R/W-0 | | R/W-0 | R-0 | | R-1 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SCL_O_FUNC | SDA_I_FUNC | SDA_O_FUNC | Reserved | SCL_I | SCL_O | SDA_I | SDA_O |
| R-1 | R-1 | R-1 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-29. I2C_SYSTEST Register (System Test) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15 | ST_EN | | System test enable. This bit must be set to 1 to permit other system test registers bits to be set. |
| | | 0 | Normal mode. All others bits in register are read only. |
| | | 1 | System test enabled. Permit other system test registers bits to be set. |
| | | | Value after reset is low. |
| 14 | FREE | | Free running mode (on breakpoint). This bit is used to determine the state of the I2C controller when a breakpoint is encountered in the HLL debugger. |
| | | | **Note:** This bit can be set independently of ST_EN value. FREE = 0: the I2C controller stops immediately after completion of the on-going bit transfer. Stopping the transfer is achieved by forcing the SCL line low. Note that in this case there will be no status register updates. |
| | | | FREE = 1: the I2C interface runs free. When Suspend indication will be asserted, there will be no accesses on the OCP Interface (the CPU is in debug mode) and consequently the FIFOs will reach full/empty state (according to RX or TX modes) and the I2C SDA line will be kept low. Note that the status registers will be updated, but no DMA, IRQ or WakeUp will be generated. |
| | | | The status registers likely to be updated in this mode are: I2C_IRQSTATUS_RAW.XRDY, I2C_IRQSTATUS_RAW.RRDY, I2C_IRQSTATUS_RAW.XUDF, I2C_IRQSTATUS_RAW.ROVR, I2C_IRQSTATUS_RAW.ARDY and I2C_IRQSTATUS_RAW.NACK. |
| | | 0 | Stop mode (on breakpoint condition). If Master mode, it stops after completion of the on-going bit transfer. In slave mode, it stops during the phase transfer when 1 byte is completely transmitted/received. |
| | | 1 | Free running mode |
| | | | Value after reset is low. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-29. I2C_SYSTEST Register (System Test) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 13-12 | TMODE | | Test mode select. In normal functional mode (ST_EN = 0), these bits are don't care. They are always read as 00 and a write is ignored.<br>In system test mode (ST_EN = 1), these bits can be set according to the following table to permit various system tests. |
| | | 0 | Functional mode (default) |
| | | 1h | Reserved |
| | | 2h | Test of SCL counters (SCLL, SCLH, PSC). SCL provides a permanent clock with master mode. |
| | | 3h | Loop back mode select + SDA/SCL IO mode select |
| | | | Values after reset are low (2 bits).<br>SCL counter test mode: in this mode, the SCL pin is driven with a permanent clock as if mastered with the parameters set in the I2C_PSC, I2C_SCLL, and I2C_SCLH registers.<br>Loop back mode: in the master transmit mode only, data transmitted out of the I2C_DATA register (write action) is received in the same I2C_DATA register via an internal path through the FIFO buffer. The DMA and interrupt requests are normally generated if enabled.<br>SDA/SCL IO mode: in this mode, the SCL IO and SDA IO are controlled via the I2C_SYSTEST [5:0] register bits. |
| 11 | SSB | | Set status bits. Writing 1 into this bit also sets the 6 read/clear-only status bits contained in I2C_IRQSTATUS_RAW register (bits 5:0) to 1. Writing 0 into this bit doesn't clear status bits that are already set; only writing 1 into a set status bit can clear it (see I2C_IRQSTATUS_RAW operation). This bit must be cleared prior attempting to clear a status bit. |
| | | 0 | No action |
| | | 1 | Set all interrupt status bits to 1 |
| | | | Value after reset is low. |
| 10-9 | Reserved | 0 | Reserved |
| 8 | SCL_I_FUNC | | SCL line input value (functional mode). This read-only bit returns the logical state taken by the SCL line (either 1 or 0). It is active both in functional and test mode |
| | | Read:0 | Read 0 from SCL line |
| | | Read:1 | Read 1 from SCL line |
| | | | Value after reset is low. |
| 7 | SCL_O_FUNC | | SCL line output value (functional mode). This read-only bit returns the value driven by the module on the SCL line (either 1 or 0). It is active both in functional and test mode. |
| | | Read:0 | Driven 0 on SCL line |
| | | Read:1 | Driven 1 on SCL line |
| | | | Value after reset is low. |
| 6 | SDA_I_FUNC | | SDA line input value (functional mode). This read-only bit returns the logical state taken by the SDA line (either 1 or 0). It is active both in functional and test mode. |
| | | Read:0 | Read 0 from SDA line |
| | | Read:1 | Read 1 from SDA line |
| | | | Value after reset is low. |
| 5 | SDA_O_FUNC | | SDA line output value (functional mode). This read-only bit returns the value driven by the module on the SDA line (either 1 or 0). It is active both in functional and test mode. |
| | | Read:0 | Driven 0 to SDA line |
| | | Read:1 | Driven 1 to SDA line |
| | | | Value after reset is low. |
| 4 | Reserved | 0 | Reserved |
| 3 | SCL_I | | SCL line sense input value. In normal functional mode (ST_EN = 0), this read-only bit always reads 0. In system test mode (ST_EN = 1 & TMODE = 11), this read-only bit returns the logical state taken by the SCL line (either 1 or 0). |
| | | Read:0 | Read 0 from SCL line |
| | | Read:1 | Read 1 from SCL line |
| | | | Value after reset is low. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 15-29. I2C_SYSTEST Register (System Test) Field Descriptions (continued)**

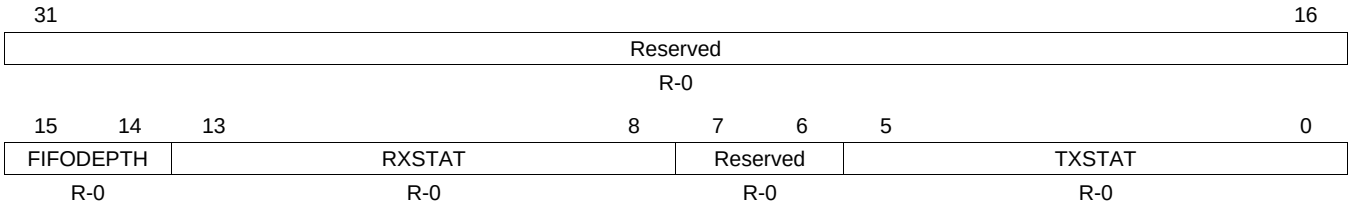
| Bit | Field | Value | Description |
|---|---|---|---|
| 2 | SCL_O | | SCL line drive output value. In normal functional mode (ST_EN = 0), this bit is don't care. It always reads 0 and a write is ignored. In system test mode (ST_EN = 1 & TMODE = 11), a 0 forces a low level on the SCL line and a 1 puts the I2C output driver to a high-impedance state. |
| | | Read:0 | Forces 0 on the SCL data line |
| | | Read:1 | SCL output driver in high-impedance state |
| | | | Value after reset is low. |
| 1 | SDA_I | | SDA line sense input value. In normal functional mode (ST_EN = 0), this read-only bit always reads 0. In system test mode (ST_EN = 1 & TMODE = 11), this read-only bit returns the logical state taken by the SDA line (either 1 or 0). |
| | | Read:0 | Read 0 from SDA line |
| | | Read:1 | Read 1 from SDA line |
| | | | Value after reset is low. |
| 0 | SDA_O | | SDA line drive output value. In normal functional mode (ST_EN = 0), this bit is don't care. It reads as 0 and a write is ignored. In system test mode (ST_EN = 1 & TMODE = 11), a 0 forces a low level on the SDA line and a 1 puts the I2C output driver to a high-impedance state. |
| | | 0 | Write 0 to SDA line |
| | | 1 | Write 1 to SDA line |
| | | | Value after reset is low. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.27 I2C_BUFSTAT Register (I2C Buffer Status)

This read-only register reflects the status of the internal buffers for the FIFO management (see the FIFO Management subsection).

**Figure 15-41. I2C_BUFSTAT Register (I2C Buffer Status)**




LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 15-30. I2C_BUFSTAT Register (I2C Buffer Status) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15-14 | FIFODEPTH | | Internal FIFO buffers depth. This read-only bit indicates the internal FIFO buffer depth. |
| | | 0 | 8-bytes FIFO |
| | | 1h | 16-bytes FIFO |
| | | 2h | 32-bytes FIFO |
| | | 3h | 64-bytes FIFO |
| | | | Value after reset is given by the boundary module generic parameter. |
| 13-8 | RXSTAT | 0-3Fh | RX buffer status. This read-only field indicates the number of bytes to be transferred from the FIFO at the end of the I2C transfer (when RDR is asserted). It corresponds to the level indication of the RX FIFO (number of written locations). |
| | | | Value after reset is 0. |
| 7-6 | Reserved | 0 | Reserved |
| 5-0 | TXSTAT | 0-3Fh | TX buffer status. This read-only field indicates the number of data bytes still left to be written in the TX FIFO (it's equal with the initial value of I2C_CNT.DCOUNT minus the number of data bytes already written in the TX FIFO through the OCP Interface). |
| | | | Value after reset is equal with 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.28 I2C_OA1 Register (OA1) (Own Address 1)

**CAUTION**

During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the first alternative I2C 7-bit or 10-bit address (own address 1 - OA1).

**Figure 15-42. I2C_OA1 Register (OA1) (Own Address 1)**

| 31 | 10 | 9 | 0 |
|---|---|---|---|
| Reserved | | 0A1 | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-31. I2C_OA1 Register (OA1) (Own Address 1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9-0 | OA1 | 0-3FFh | Own address 1. This field specifies either:<br>• A 10-bit address coded on OA1 [9:0] when XOA1 (Expand Own Address 1 - XOA1, I2C_CON[6]) is set to 1.<br>• A 7-bit address coded on OA1 [6:0] when XOA1 (Expand Own Address 1 – XOA1, I2C_CON[6]) is cleared to 0. In this case, OA1 [9:7] bits must be cleared to 000 by application software.<br>Value after reset is low (all 10 bits). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.29 I2C_OA2 Register (I2C Own Address 2)

**CAUTION**

During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the first alternative I2C 7-bit or 10-bit address (own address 2 - OA2).

**Figure 15-43. I2C_OA2 Register (I2C Own Address 2)**

| 31 | 10 | 9 | 0 |
|---|---|---|---|
| Reserved | | 0A2 | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-32. I2C_OA2 Register (I2C Own Address 2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9-0 | OA2 | 0-3FFh | Own address 2. This field specifies either:<br>• A 10-bit address coded on OA2 [9:0] when XOA1 (Expand Own Address 2 - XOA2, I2C_CON[5]) is set to 1.<br>• A 7-bit address coded on OA2 [6:0] when XOA2 (Expand Own Address 2 – XOA2, I2C_CON[5]) is cleared to 0. In this case, OA2 [9:7] bits must be cleared to 000 by application software.<br>Value after reset is low (all 10 bits). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.30 I2C_OA3 Register (I2C Own Address 3)

**CAUTION**

During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the first alternative I2C 7-bit or 10-bit address (own address 3 - OA3).

**Figure 15-44. I2C_OA3 Register (I2C Own Address 3)**

| 31 | 10 | 9 | 0 |
|---|---|---|---|
| Reserved | | 0A3 | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-33. I2C_OA3 Register (I2C Own Address 3) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9-0 | OA3 | 0-3FFh | Own address 2. This field specifies either:<br>• A 10-bit address coded on OA3 [9:0] when XOA3 (Expand Own Address 3 - XOA3, I2C_CON[4]) is set to 1.<br>• A 7-bit address coded on OA3 [6:0] when XOA1 (Expand Own Address 3 – XOA3, I2C_CON[4]) is cleared to 0. In this case, OA3 [9:7] bits must be cleared to 000 by application software.<br>Value after reset is low (all 10 bits). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.31 I2C_ACTOA Register (Active Own Address)

This read-only register is used to indicate which one of the module's four own addresses the external master used when addressing the module. The CPU can read this register when the AAS indication was activated. The indication is cleared at the end of the transfer.

**Figure 15-45. I2C_ACTOA Register (Active Own Address)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | OA3_ACT | OA2_ACT | OA1_ACT | OA0_ACT |
| R-0 | | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-34. I2C_ACTOA Register (Active Own Address) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | OA3_ACT | | Own address 3 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module. |
| | | 0 | Own address inactive |
| | | 1 | Own address active |
| | | | Value after reset is low. |
| 2 | OA2_ACT | | Own address 2 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module. |
| | | 0 | Own address inactive |
| | | 1 | Own address active |
| | | | Value after reset is low. |
| 1 | OA1_ACT | | Own address 1 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module. |
| | | 0 | Own address inactive |
| | | 1 | Own address active |
| | | | Value after reset is low. |
| 0 | OA0_ACT | | Own address 0 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module. |
| | | 0 | Own address inactive |
| | | 1 | Own address active |
| | | | Value after reset is low. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 15.3.32 I2C_SBLOCK Register (I2C Clock Blocking Enable)

This read/write register controls the automatic blocking of I2C clock feature in slave mode. It is used for the Local Host to configure for which of the 4 own addresses, the core must block the I2C clock (keep SCL line low) right after the Address Phase, when it is addressed as a slave.

**Figure 15-46. I2C_SBLOCK Register (I2C Clock Blocking Enable)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | OA3_EN | OA2_EN | OA1_EN | OA0_EN |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 15-35. I2C_SBLOCK Register (I2C Clock Blocking Enable) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | OA3_EN | | Enable I2C clock blocking for own address 3. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write '0' in the corresponding field. |
| | | 0 | I2C clock released |
| | | 1 | I2C clock blocked |
| | | | Value after reset is low. |
| 2 | OA2_EN | | Enable I2C clock blocking for own address 2. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write '0' in the corresponding field. |
| | | 0 | I2C clock released |
| | | 1 | I2C clock blocked |
| | | | Value after reset is low. |
| 1 | OA1_EN | | Enable I2C clock blocking for own address 1. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write '0' in the corresponding field. |
| | | 0 | I2C clock released |
| | | 1 | I2C clock blocked |
| | | | Value after reset is low. |
| 0 | OA0_EN | | Enable I2C clock blocking for own address 0. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write '0' in the corresponding field. |
| | | 0 | I2C clock released |
| | | 1 | I2C clock blocked |
| | | | Value after reset is low. |


Copyright © 2011–2016, Texas Instruments Incorporated

# Multichannel Audio Serial Port (McASP)

This chapter is a description of the IP of the multichannel audio serial port (McASP).

**Topic** **Page**

16.1 Introduction ..... 2948
16.2 Architecture ..... 2960
16.3 McASP Registers ..... 3003


Copyright © 2011–2016, Texas Instruments Incorporated

## 16.1 Introduction

### 16.1.1 Purpose of the Peripheral

The multichannel audio serial port (McASP) functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. The McASP is useful for time-division multiplexed (TDM) stream, Inter-Integrated Sound (I2S) protocols, and intercomponent digital audio interface transmission (DIT). The MCASP has the flexibility to gluelessly connect to a Sony / Philips digital interface (S/PDIF) transmit physical layer component. The McASP consists of transmit and receive sections that may operate synchronized, or completely independently with separate master clocks, bit clocks, and frame syncs, and using different transmit modes with different bit-stream formats. The McASP module also includes serializers that can be individually enabled to either transmit or receive.

Although intercomponent digital audio interface reception (DIR) mode (that is, S/PDIF stream receiving) is NOT natively supported by the MCASP module, a specific TDM mode implementation for the MCASP receivers allows an easy connection to external DIR components (for example, S/PDIF to I2S format converters).

### 16.1.2 Features

Features of the McASP include:

- Two independent clock generator modules for transmit and receive.
  - Clocking flexibility allows the McASP to receive and transmit at different rates. For example, the McASP can receive data at 48 kHz but output up-sampled data at 96 kHz or 192 kHz.
- Independent transmit and receive modules, each includes:
  - Programmable clock and frame sync generator.
  - TDM streams from 2 to 32, and 384 time slots.
  - Support for time slot sizes of 8, 12, 16, 20, 24, 28, and 32 bits.
  - Data formatter for bit manipulation.
- Individually assignable serial data pins.
- Glueless connection to audio analog-to-digital converters (ADC), digital-to-analog converters (DAC), codec, digital audio interface receiver (DIR), and S/PDIF transmit physical layer components.
- Wide variety of I2S and similar bit-stream format.
- Integrated digital audio interface transmitter (DIT) supports (up to 10 transmit pins):
  - S/PDIF, IEC60958-1, AES-3 formats.
  - Enhanced channel status/user data RAM.
- 384-slot TDM with external digital audio interface receiver (DIR) device.
  - For DIR reception, an external DIR receiver integrated circuit should be used with I2S output format and connected to the McASP receive section.
- Extensive error checking and recovery.
  - Transmit underruns and receiver overruns due to the system not meeting real-time requirements.
  - Early or late frame sync in TDM mode.
  - Out-of-range high-frequency master clock for both transmit and receive.
  - External error signal coming into the AMUTEIN input.
  - DMA error due to incorrect programming.
- up to 16 serial data pins depending of the McASP instance in use (see your device-specific data manual) - each serializer having associated one transmit (Tx) and one receive (Rx) channel, which allows support of:
  - 16 data channels, each configurable to either transmit or receive at a time
  - A single 32-bit buffer per serializer for transmit and receive operations
  - One transmit interrupt request (AXINT) and one receive interrupt request (ARINT) linked with each of the 16 serializers


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** Because a serializer receive and transmit channels data is shared on the same MCASP data pin, you can choose to have either Tx or Rx function from a serializer, NOT both at the same time.

### 16.1.3 Protocols Supported

The McASP supports a wide variety of protocols.

- Transmit section supports:
  - Wide variety of I2S and similar bit-stream formats.
  - TDM streams from 2 to 32 time slots.
  - S/PDIF, IEC60958-1, AES-3 formats.
- Receive section supports:
  - Wide variety of I2S and similar bit-stream formats.
  - TDM streams from 2 to 32 time slots.
  - TDM stream of 384 time slots specifically designed for easy interface to external digital interface receiver (DIR) device transmitting DIR frames to McASP using the I2S protocol (one time slot for each DIR subframe).

The transmit and receive sections may each be individually programmed to support the following options on the basic serial protocol:

- Programmable clock and frame sync polarity (rising or falling edge): ACLKR/X, AHCLKR/X, and AFSR/X.
- Slot length (number of bits per time slot): 8, 12, 16, 20, 24, 28, 32 bits supported.
- Word length (bits per word): 8, 12, 16, 20, 24, 28, 32 bits; always less than or equal to the time slot length.
- First-bit data delay: 0, 1, 2 bit clocks.
- Left/right alignment of word inside slot.
- Bit order: MSB first or LSB first.
- Bit mask/pad/rotate function.
  - Automatically aligns data internally in either Q31 or integer formats.
  - Automatically masks non-significant bits (sets to 0, 1, or extends value of another bit).

In DIT mode for McASP, additional features of the transmitter are:

- Transmit-only mode 384 time slots (subframe) per frame.
- Bi-phase encoded 3.3 V output.
- Support for consumer and professional applications.
- Channel status RAM (384 bits).
- User data RAM (384 bits).
- Separate valid bit (V) for subframe A, B.

In I2S mode, the transmit and receive sections can support simultaneous transfers on up to all serial data pins operating as 192 kHz stereo channels.

In DIT mode, the transmitter can support a 192 kHz frame rate (stereo) on up to all serial data pins simultaneously (note that the internal bit clock for DIT runs two times faster than the equivalent bit clock for I2S mode, due to the need to generate Biphase Mark Encoded Data).

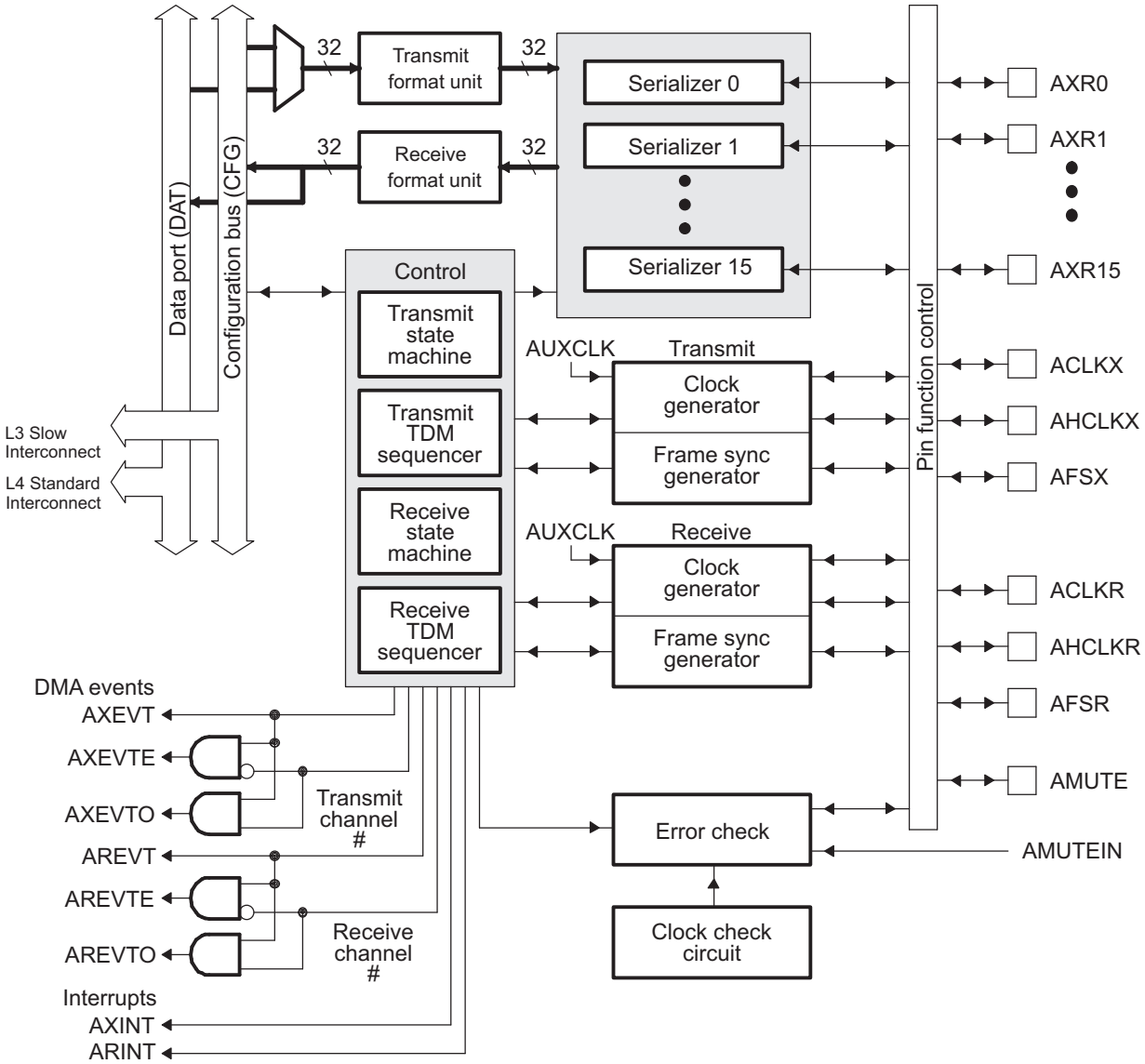
The MCASP does NOT natively support DIR-mode reception (that is, receiving in the S/PDIF format). To allow this, the MCASP can use a DIR-input to I2S-output converter implemented by an external device (that is, external DIR component). To facilitate reception in this case, the TDM mode of MCASP receivers logic is extended to support a non-standard 384-slot TDM stream.


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.1.4 Functional Block Diagram

Figure 16-1 shows the major blocks of the McASP. The McASP has independent receive/transmit clock generators and frame sync generators.

**Figure 16-1. McASP Block Diagram**




A McASP has up to 16 serial data pins depending of the McASP instance in use (see your device-specific data manual). When referring to AXRn data pins in this document, n will be up to 15.

B AMUTEIN is not a dedicated McASP pin, but typically comes from one of the external interrupt pins.



Copyright © 2011–2016, Texas Instruments Incorporated

### 16.1.4.1 System Level Connections

Figure 16-2 through Figure 16-6 show examples of McASP usage in digital audio encoder/decoder systems.

**Figure 16-2. McASP to Parallel 2-Channel DACs**

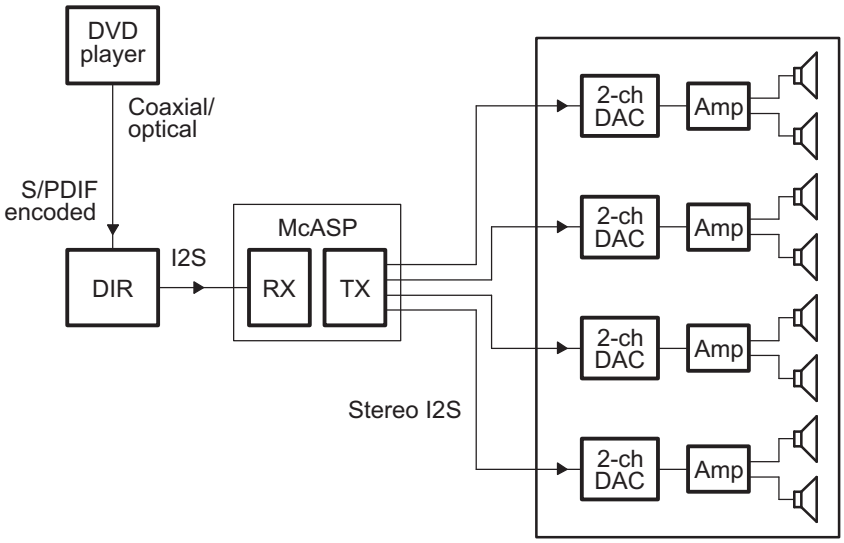
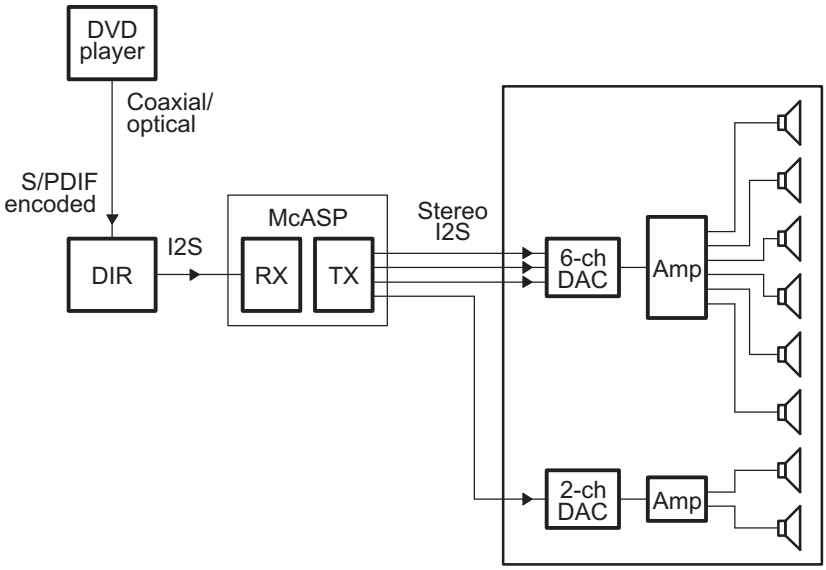



**Figure 16-3. McASP to 6-Channel DAC and 2-Channel DAC**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 16-4. McASP to Digital Amplifier**

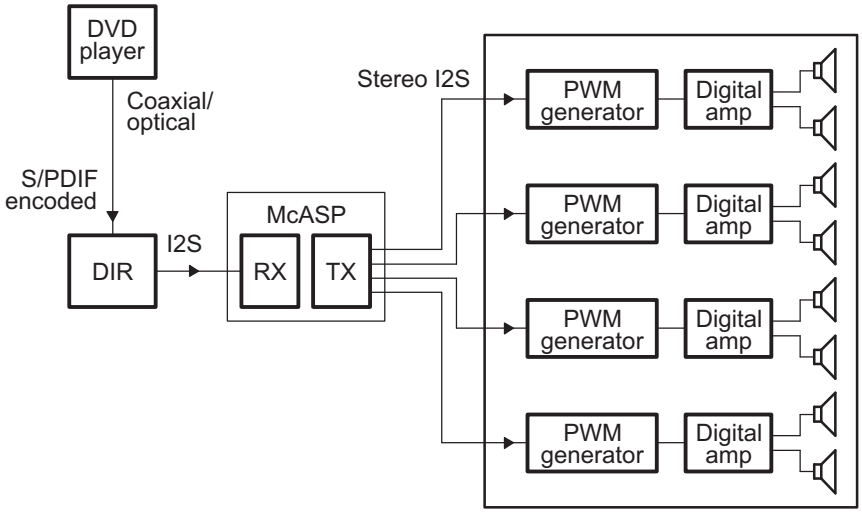



**Figure 16-5. McASP as Digital Audio Encoder**

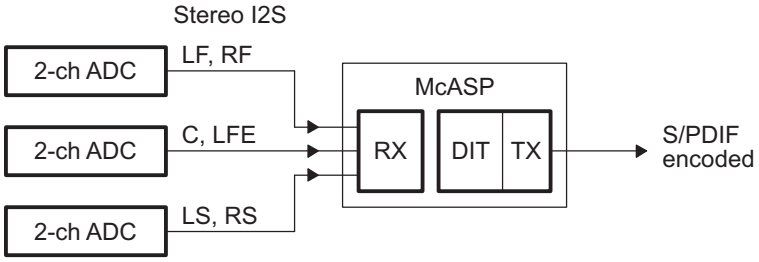
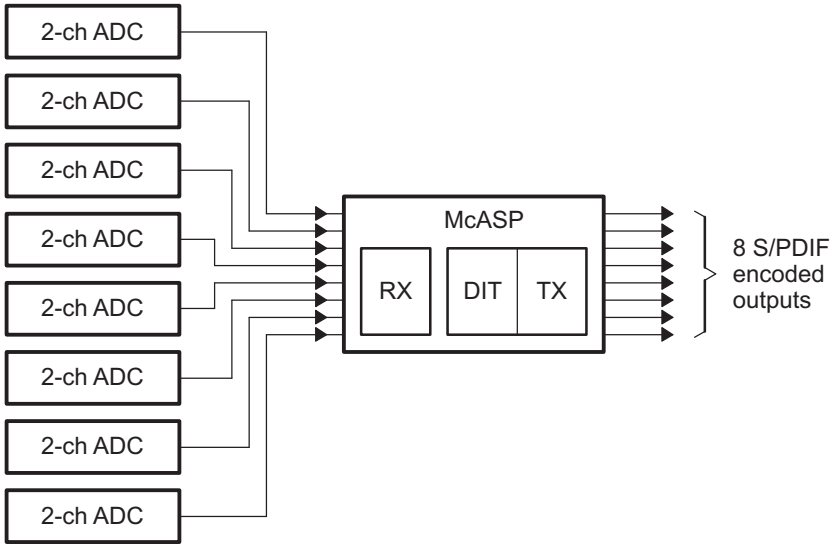



**Figure 16-6. McASP as 16 Channel Digital Processor**





Copyright © 2011–2016, Texas Instruments Incorporated

### 16.1.5 Supported Use Case Statement

### 16.1.6 Industry Standard Compliance Statement

The McASP supports the following industry standard interfaces.

#### 16.1.6.1 TDM Format

The McASP transmitter and receiver support the multichannel, synchronous time-division-multiplexed (TDM) format via the TDM transfer mode. Within this transfer mode, a wide variety of serial data formats are supported, including formats compatible with devices using the Inter-Integrated Sound (I2S) protocol. This section briefly discusses the TDM format and the I2S protocol.

##### 16.1.6.1.1 TDM Format

The TDM format is typically used when communicating between integrated circuit devices on the same printed circuit board or on another printed circuit board within the same piece of equipment. For example, the TDM format is used to transfer data between the processor and one or more analog-to-digital converter (ADC), digital-to-analog converter (DAC), or S/PDIF receiver (DIR) devices.

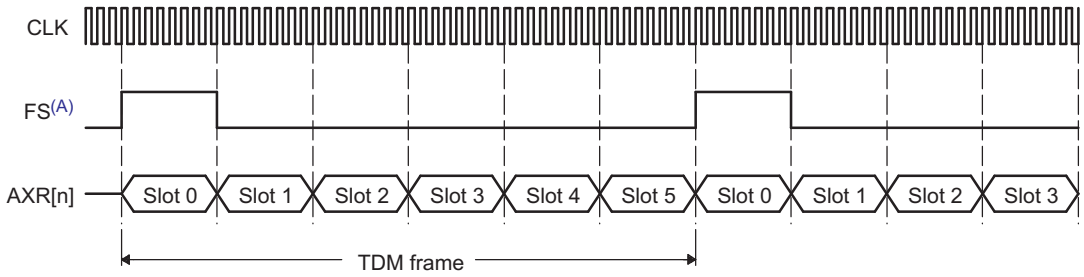
The TDM format consists of three components in a basic synchronous serial transfer: clock (CLK), data (AXRn), and the frame sync (FS). In a TDM transfer, all data bits (AXRn) are synchronous to the serial clock (ACLKX or ACLKR). The data bits are grouped into words and slots (as defined in Section 16.1.7). The "slots" are also commonly referred to as "time slots" or "channels" in TDM terminology. A frame consists of multiple slots (or channels). Each TDM frame is defined by the frame sync signal (AFSX or AFSR). Data transfer is continuous and periodic, since the TDM format is most commonly used to communicate with data converters that operate at a fixed sample rate.

There are no delays between slots. The last bit of slot N is followed immediately on the next serial clock cycle with the first bit of slot N + 1, and the last bit of the last slot is followed immediately on the next serial clock with the first bit of the first slot. However, the frame sync may be offset from the first bit of the first slot with a 0, 1, or 2-cycle delay.

It is required that the transmitter and receiver in the system agree on the number of bits per slot, since the determination of a slot boundary is not made by the frame sync signal (although the frame sync marks the beginning of slot 0 and the beginning of a new frame).

Figure 16-7 shows the TDM format. Figure 16-8 shows the different bit delays from the frame sync.

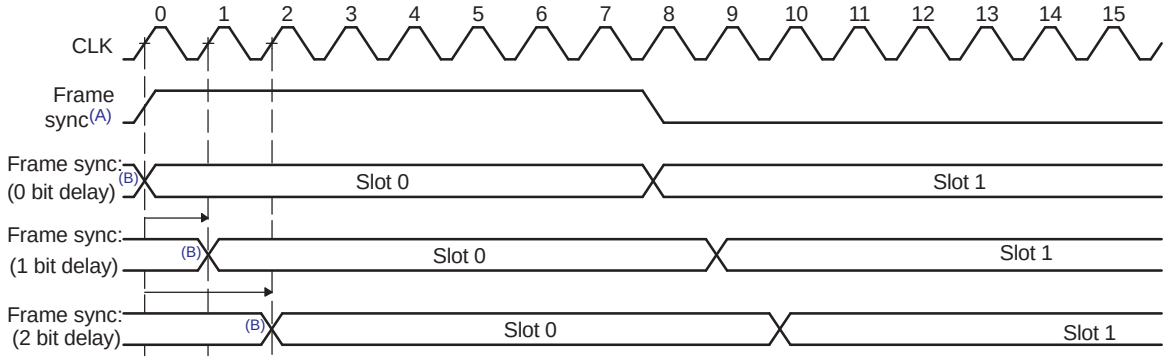
**Figure 16-7. TDM Format–6 Channel TDM Example**




A FS duration of slot is shown. FS duration of single bit is also supported.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 16-8. TDM Format Bit Delays from Frame Sync**




A FS duration of slot is shown. FS duration of single bit is also supported.

B Last bit of last slot of previous frame. No gap between this bit and the first bit of slot 0 is allowed.

In a typical audio system, one frame of data is transferred during each data converter sample period $f_s$. To support multiple channels, the choices are to either include more time slots per frame (thus operating with a higher bit clock rate), or to use additional data pins to transfer the same number of channels (thus operating with a slower bit clock rate).

For example, a particular six channel DAC may be designed to transfer over a single serial data pin AXRn as shown in Figure 16-7. In this case the serial clock must run fast enough to transfer a total of 6 channels within each frame period. Alternatively, a similar six channel DAC may be designed to use three serial data pins AXR[0,1,2], transferring two channels of data on each pin during each sample period. In the latter case, if the sample period remains the same, the serial clock can run three times slower than the former case. The McASP is flexible enough to support either type of DAC.

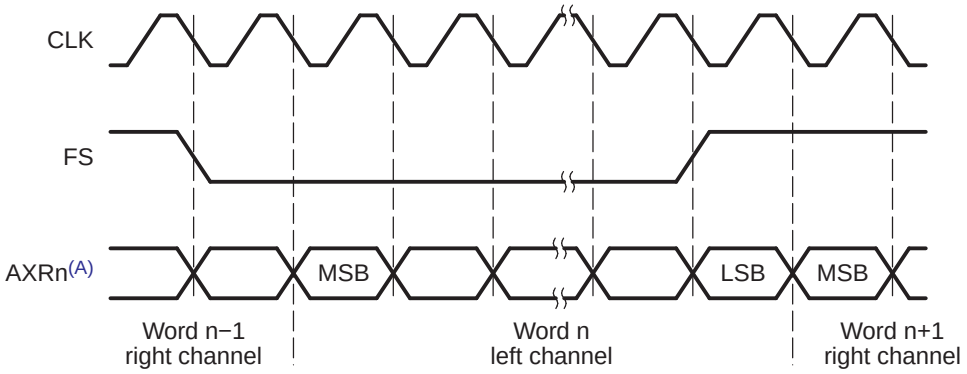
#### *16.1.6.1.2 Inter-Integrated Sound (I2S) Format*

The inter-integrated sound (I2S) format is used extensively in audio interfaces. The TDM transfer mode of the McASP supports the I2S format when configured to 2 slots per frame.

I2S format is specifically designed to transfer a stereo channel (left and right) over a single data pin AXRn. "Slots" are also commonly referred to as "channels". The frame width duration in the I2S format is the same as the slot size. The frame signal is also referred to as "word select" in the I2S format. Figure 16-9 shows the I2S protocol.

The McASP supports transfer of multiple stereo channels over multiple AXRn pins.

**Figure 16-9. Inter-Integrated Sound (I2S) Format**




A 1 to 16 data pins may be supported.


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.1.6.2 S/PDIF Coding Format

The McASP transmitter supports the S/PDIF format with 3.3V biphase-mark encoded output. The S/PDIF format is supported by the digital audio interface transmit (DIT) transfer mode of the McASP. This section briefly discusses the S/PDIF coding format.

#### *16.1.6.2.1 Biphase-Mark Code (BMC)*

In S/PDIF format, the digital signal is coded using the biphase-mark code (BMC). The clock, frame, and data are embedded in only one signal—the data pin AXRn. In the BMC system, each data bit is encoded into two logical states (00, 01, 10, or 11) at the pin. These two logical states form a cell. The duration of the cell, which equals to the duration of the data bit, is called a time interval. A logical 1 is represented by two transitions of the signal within a time interval, which corresponds to a cell with logical states 01 or 10. A logical 0 is represented by one transition within a time interval, which corresponds to a cell with logical states 00 or 11. In addition, the logical level at the start of a cell is inverted from the level at the end of the previous cell. Figure 16-10 and Table 16-1 show how data is encoded to the BMC format.

As shown in Figure 16-10, the frequency of the clock is twice the unencoded data bit rate. In addition, the clock is always programmed to 128 × $f_s$, where $f_s$ is the sample rate (see Section 16.1.6.2.3 for details on how this clock rate is derived based on the S/PDIF format). The device receiving in S/PDIF format can recover the clock and frame information from the BMC signal.

**Figure 16-10. Biphase-Mark Code (BMC)**

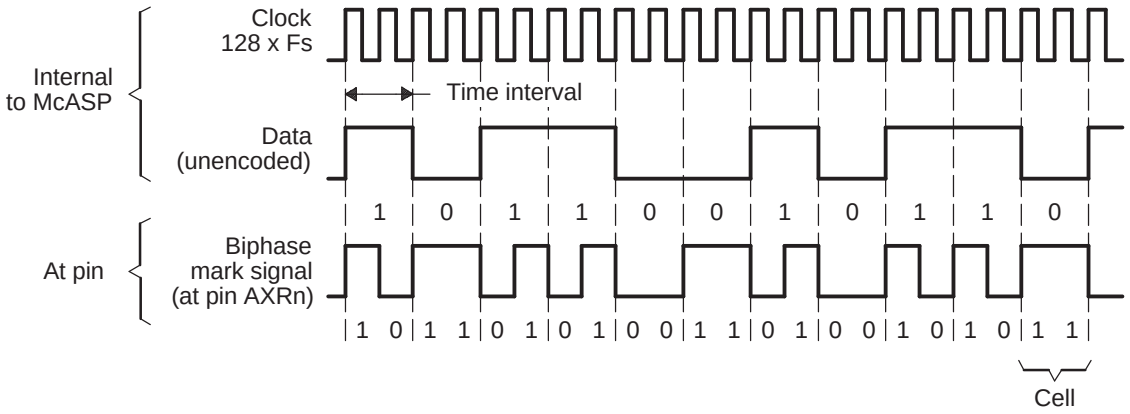



**Table 16-1. Biphase-Mark Encoder**

| Data (Unencoded) | Previous State at Pin AXRn | BMC-Encoded Cell Output at AXRn |
|---|---|---|
| 0 | 0 | 11 |
| 0 | 1 | 00 |
| 1 | 0 | 10 |
| 1 | 1 | 01 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.1.6.2.2 Subframe Format

Every audio sample transmitted in a subframe consists of 32 S/PDIF time intervals (or cells), numbered from 0 to 31. Figure 16-11 shows a subframe.

- **Time intervals 0-3** carry one of the three permitted preambles to signify the type of audio sample in the current subframe. The preamble is *not* encoded in BMC format, and therefore the preamble code can contain more than two consecutive 0 or 1 logical states in a row. See Table 16-2.
- **Time intervals 4-27** carry the audio sample word in linear 2s-complement representation. The most-significant bit (MSB) is carried by time interval 27. When a 24-bit coding range is used, the least-significant bit (LSB) is in time interval 4. When a 20-bit coding range is used, time intervals 8-27 carry the audio sample word with the LSB in time interval 8. Time intervals 4-7 may be used for other applications and are designated auxiliary sample bits.
- If the source provides fewer bits than the interface allows (either 20 or 24), the unused LSBs are set to logical 0. For a nonlinear PCM audio application or a data application, the main data field may carry any other information.
- **Time interval 28** carries the validity bit (V) associated with the main data field in the subframe.
- **Time interval 29** carries the user data channel (U) associated with the main data field in the subframe.
- **Time interval 30** carries the channel status information (C) associated with the main data field in the subframe. The channel status indicates if the data in the subframe is digital audio or some other type of data.
- **Time interval 31** carries a parity bit (P) such that time intervals 4-31 carry an even number of 1s and an even number of 0s (even parity). As shown in Table 16-2, the preambles (time intervals 0-3) are also defined with even parity.

**Figure 16-11. S/PDIF Subframe Format**

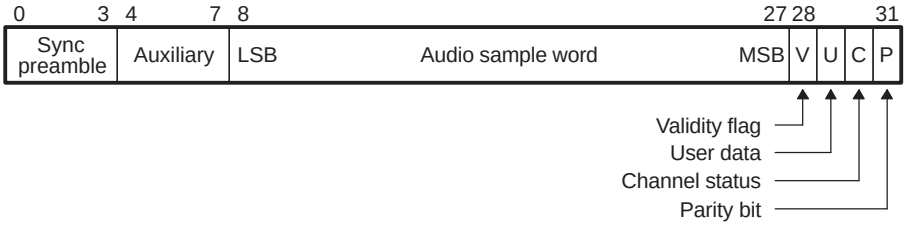



**Table 16-2. Preamble Codes**

| Preamble Code[1] | Previous Logical State | Logical States on pin AXRn[2] | Description |
|---|---|---|---|
| B (or Z) | 0 | 1110 1000 | Start of a block and subframe 1 |
| M (or X) | 0 | 1110 0010 | Subframe 1 |
| W (or Y) | 0 | 1110 0100 | Subframe 2 |

(1) Historically, preamble codes are referred to as B, M, W. For use in professional applications, preambles are referred to as Z, X, Y, respectively.

(2) The preamble is not BMC encoded. Each logical state is synchronized to the serial clock. These 8 logical states make up time slots (cells) 0 to 3 in the S/PDIF stream.

As shown in Table 16-2, the McASP DIT only generates one polarity of preambles and it assumes the previous logical state to be 0. This is because the McASP assures an even-polarity encoding scheme when transmitting in DIT mode. If an underrun condition occurs, the DIT resynchronizes to the correct logic level on the AXRn pin before continuing with the next transmission.


Copyright © 2011–2016, Texas Instruments Incorporated

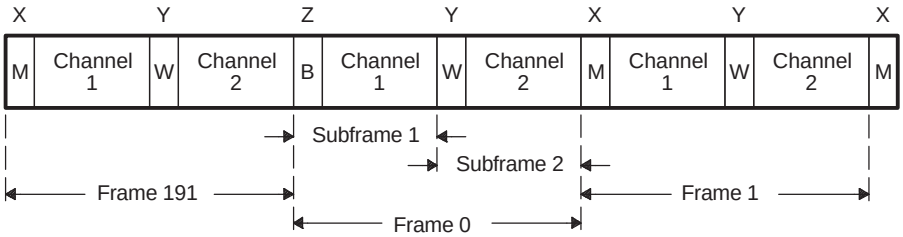
#### 16.1.6.2.3 Frame Format

An S/PDIF frame is composed of two subframes (Figure 16-12). For linear coded audio applications, the rate of frame transmission normally corresponds exactly to the source sampling frequency $f_s$. The S/PDIF format clock rate is therefore 128 × $f_s$ (128 = 32 cells/subframe × 2 clocks/cell × 2 subframes/sample). For example, for an S/PDIF stream at a 192 kHz sampling frequency, the serial clock is 128 × 192 kHz = 24.58 MHz.

In 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive subframes. Both subframes contain valid data. The first subframe (**left** or **A** channel in stereophonic operation and **primary** channel in monophonic operation) normally starts with preamble M. However, the preamble of the first subframe changes to preamble B once every 192 frames to identify the start of the block structure used to organize the channel status information. The second subframe (**right** or **B** channel in stereophonic operation and **secondary** channel in monophonic operation) always starts with preamble W.

In single-channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried in the first subframe and may be duplicated in the second subframe. If the second subframe is not carrying duplicate data, cell 28 (validity bit) is set to logical 1.

**Figure 16-12. S/PDIF Frame Format**




### 16.1.7 Definition of Terms

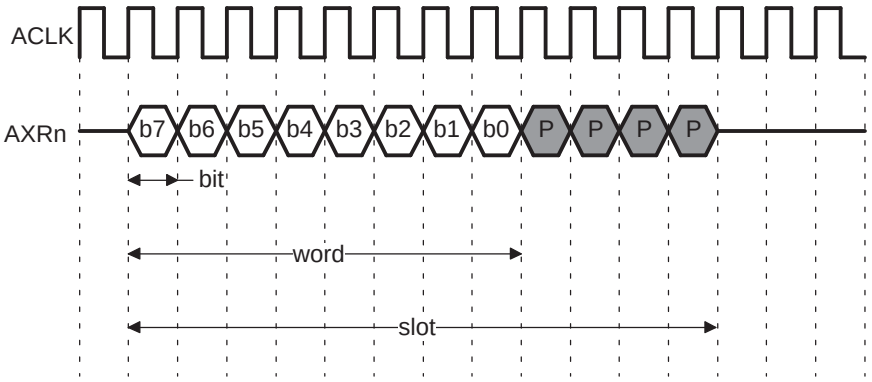
The serial bit stream transmitted or received by the McASP is a long sequence of 1s and 0s, either output or input on one of the audio transmit/receive pins (AXRn). However, the sequence has a hierarchical organization that can be described in terms of frames of data, slots, words, and bits.

A basic synchronous serial interface consists of three important components: clock, frame sync, and data. Figure 16-13 shows two of the three basic components—the clock (ACLK) and the data (AXRn). Figure 16-13 does not specify whether the clock is for transmit (ACLKX) or receive (ACLKR) because the definitions of terms apply to both receive and transmit interfaces. In operation, the transmitter uses ACLKX as the serial clock, and the receiver uses ACLKR as the serial clock. Optionally, the receiver can use ACLKX as the serial clock when the transmitter and receiver of the McASP are configured to operate synchronously.

| | |
|---|---|
| **Bit** | A bit is the smallest entity in the serial data stream. The beginning and end of each bit is marked by an edge of the serial clock. The duration of a bit is a serial clock period. A "1" is represented by a logic high on the AXRn pin for the entire duration of the bit. A "0" is represented by a logic low on the AXRn pin for the entire duration of the bit. |
| **Word** | A word is a group of bits that make up the data being transferred between the processor and the external device. Figure 16-13 shows an 8-bit word. |
| **Slot** | A slot consists of the bits that make up the word, and may consist of additional bits used to pad the word to a convenient number of bits for the interface between the processor and the external device. In Figure 16-13, the audio data consists of only 8 bits of useful data (8-bit word), but it is padded with 4 zeros (12-bit slot) to satisfy the desired protocol in interfacing to an external device. Within a slot, the bits may be shifted in/out of the McASP on the AXRn pin either MSB or LSB first. When the word size is smaller than the slot size, the word may be aligned to the left (beginning) of the slot or to the right (end) of the slot. The additional bits in the slot not belonging to the word may be padded with 0, 1, or with one of the bits (the MSB or the LSB typically) from the data word. These options are shown in Figure 16-14. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 16-13. Definition of Bit, Word, and Slot**

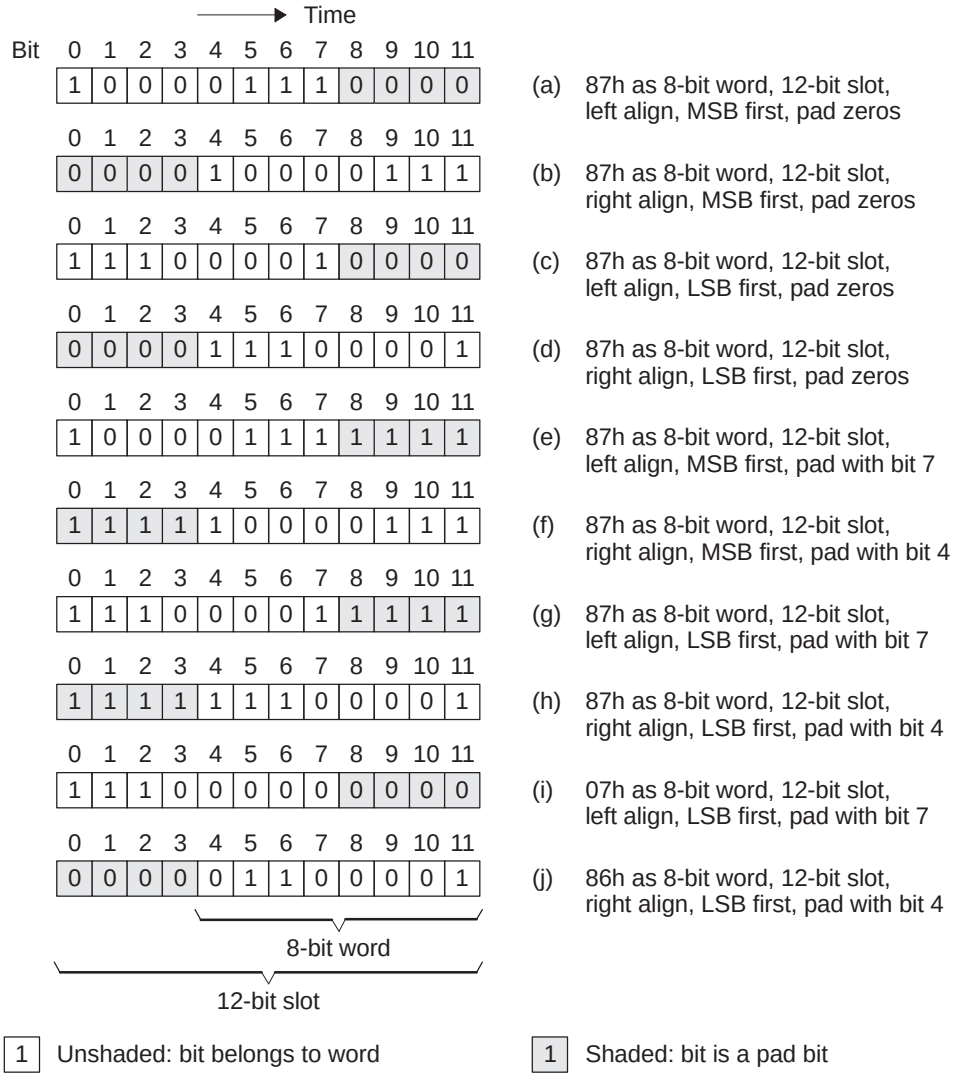



(1) b7:b0 - bits. Bits b7 to b0 form a word.

(2) P - pad bits. Bits b7 to b0, together with the four pad bits, form a slot.

(3) In this example, the data is transmitted MSB first, left aligned.

**Figure 16-14. Bit Order and Word Alignment Within a Slot Examples**





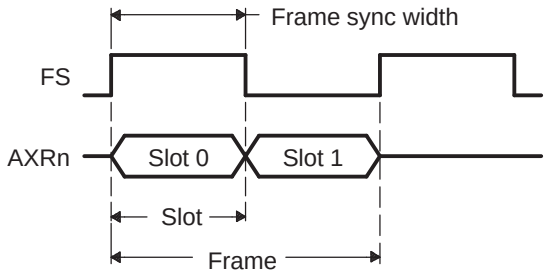

Copyright © 2011–2016, Texas Instruments Incorporated

The third basic element of a synchronous serial interface is the frame synchronization signal, also referred to as frame sync (FS) in this document.

| | |
|---|---|
| **Frame** | A frame contains one or multiple slots, as determined by the desired protocol. Figure 16-15 shows an example frame of data and the frame definitions. Figure 16-15 does not specify whether the frame sync (FS) is for transmit (AFSX) or receive (AFSR) because the definitions of terms apply to both receive and transmit interfaces. In operation, the transmitter uses AFSX and the receiver uses AFSR. Optionally, the receiver can use AFSX as the frame sync when the transmitter and receiver of the McASP are configured to operate synchronously. |

This section only shows the generic definition of the frame sync. See Section 16.1.6 and Section 16.2.6.1 for details on the frame sync formats required for the different transfer modes and protocols (burst mode, TDM mode and I2S format, DIT mode and S/PDIF format).

**Figure 16-15. Definition of Frame and Frame Sync Width**




(1) In this example, there are two slots in a frame, and FS duration of slot length is shown.

Other terms used throughout the document:

| | |
|---|---|
| **TDM** | Time-division multiplexed. See Section 16.1.6.1 for details on the TDM protocol. |
| **DIR** | Digital audio interface receive. The McASP does not natively support receiving in the S/PDIF format. The McASP supports I2S format output by an external DIR device. |
| **DIT** | Digital audio interface transmit. The McASP supports transmitting in S/PDIF format on up to all data pins configured as outputs. |
| **I2S** | Inter-Integrated Sound protocol, commonly used on audio interfaces. The McASP supports the I2S protocol as part of the TDM mode (when configured as a 2-slot frame). |
| **Slot or Time Slot** | For TDM format, the term time slot is interchangeable with the term slot defined in this section. For DIT format, a McASP time slot corresponds to a DIT subframe. |


Copyright © 2011–2016, Texas Instruments Incorporated

## 16.2 Architecture

### 16.2.1 Overview

Figure 16-1 shows the major blocks of the McASP. The McASP has independent receive/transmit clock generators and frame sync generators, error-checking logic, and up to 16 serial data pins. Refer to the device-specific data manual for the features available on your device.

All the McASP pins on the device may be individually programmed as general-purpose I/O (GPIO) if they are not used for serial port functions.

The McASP includes the following pins:

- Serializers;
  - Data pins AXRn: Up to 16 pins per McASP.
- Transmit clock generator:
  - AHCLKX: McASP transmit high-frequency master clock.
  - ACLKX: McASP transmit bit clock.
- Transmit Frame Sync Generator;
  - AFSX: McASP transmit frame sync or left/right clock (LRCLK).
- Receive clock generator;
  - AHCLKR: McASP receive high-frequency master clock.
  - ACLKR: McASP receive bit clock.
- Receive Frame Sync Generator;
  - AFSR: McASP receive frame sync or left/right clock (LRCLK).
- Mute in/out;
  - AMUTEIN: McASP mute input (from external device).
  - AMUTE: McASP mute output.

### 16.2.2 Clock and Frame Sync Generators

The McASP clock generators are able to produce two independent clock zones: transmit and receive clock zones. The serial clock generators may be programmed independently for the transmit section and the receive section, and may be completely asynchronous to each other. The serial clock (clock at the bit rate) may be sourced:

- Internally - by passing through two clock dividers off the internal clock source (AUXCLK).
- Externally - directly from ACLKR/X pin, which are configured as inputs. In this case, the Rx/Tx high-speed clock logic is bypassed for the XCLK/RCLK generation.
- Mixed - an external high-frequency clock is input to the McASP on either the AHCLKX or AHCLKR pins, and divided down to produce the bit rate clock.

In the internal/mixed cases, the bit rate clock is generated internally and should be driven out on the ACLKX (for transmit) or ACLKR (for receive) pins. In the internal case, an internally-generated high-frequency clock may be driven out onto the AHCLKX or AHCLKR pins to serve as a reference clock for other components in the system.

The McASP requires a minimum of a bit clock and a frame sync to operate, and provides the capability to reference these clocks from an external high-frequency master clock. In DIT mode, it is possible to use only internally-generated clocks and frame syncs.

A typical role of the MCASP frame sync signal is to carry the left/right clock (LRCLK) signal when transmitting and receiving stereo data.


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.2.2.1 Transmit Clock

The transmit bit clock, ACLKX, (Figure 16-16) may be either externally sourced from the ACLKX pin or internally generated, as selected by the CLKXM bit. If internally generated (CLKXM = 1), the clock is divided down by a programmable bit clock divider (CLKXDIV) from the transmit high-frequency master clock (AHCLKX).

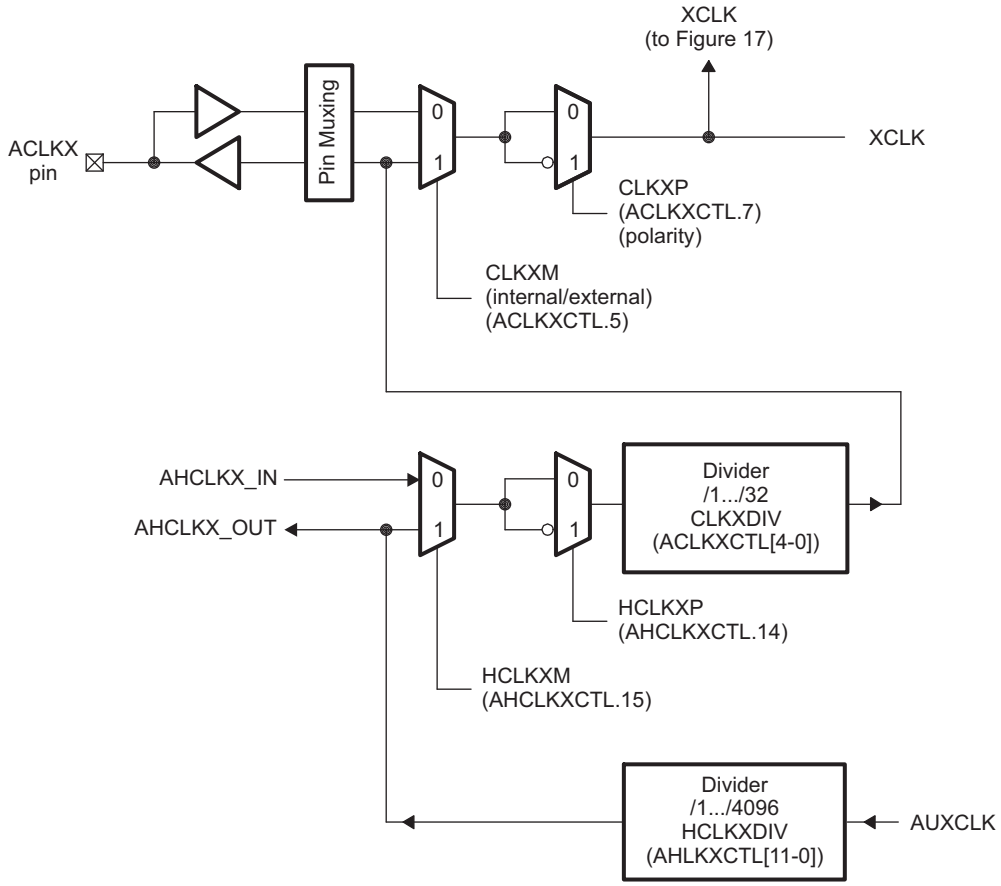
Internally, the McASP always shifts transmit data at the rising edge of the internal transmit clock, XCLK, (Figure 16-16). The CLKXP mux determines if ACLKX needs to be inverted to become XCLK. If CLKXP = 0, the CLKXP mux directly passes ACLKX to XCLK. As a result, the McASP shifts transmit data at the rising edge of ACLKX. If CLKXP = 1, the CLKX mux passes the inverted version of ACLKX to XCLK. As a result, the McASP shifts transmit data at the falling edge of ACLKX.

The transmit high-frequency master clock, AHCLKX, may be either externally sourced from the AHCLKX pin or internally generated, as selected by the HCLKXM bit. If internally generated (HCLKXM = 1), the clock is divided down by a programmable high clock divider (HCLKXDIV) from McASP internal clock source AUXCLK. The transmit high-frequency master clock may be (but is not required to be) output on the AHCLKX pin where it is available to other devices in the system.

The transmit clock configuration is controlled by the following registers:

- ACLKXCTL
- AHCLKXCTL

**Figure 16-16. Transmit Clock Generator Block Diagram**




(1) Refer to the device-specific date manual for multiplexing options.


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.2.2.2 Receive Clock

The MCASP receive clock generator is built on a very similar circuit to the transmit clock generator (but independent) circuit. The receiver also has the option to operate synchronously from the ACLKX and AFSX signals. This is achieved when the ASYNC bit in the transmit clock control register (ACLKXCTL) is cleared to 0 (see Figure 16-17). The receiver may be configured with different polarity (CLKRP) and frame sync data delay options from those options of the transmitter.

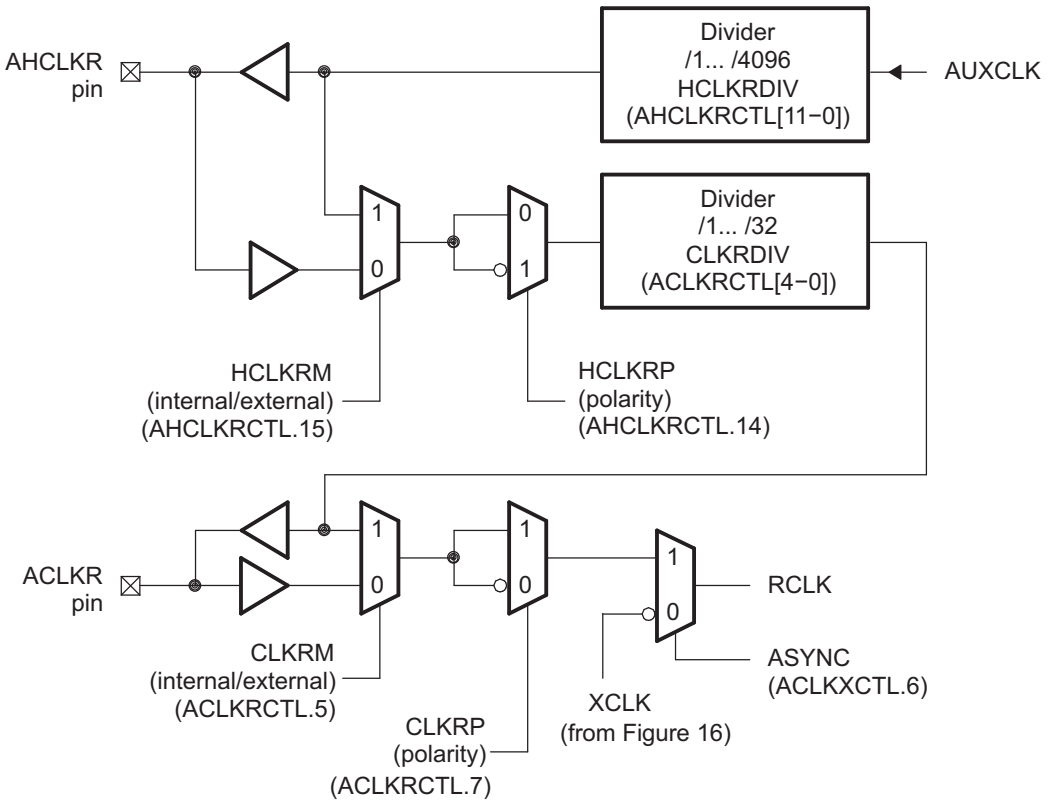
The receive clock configuration is controlled by the following registers:

- ACLKRCTL
- AHCLKRCTL

In case the receive bit clock, ACLKR, is generated internally (but asynchronously to XCLK), the CLKRM bit (from ACLKRCTL register) must be set to 1. Thus, the clock is divided down by a programmable bit clock divider (CLKRDIV bit field in ACLKRCTL register) from the source signal. If the receive high-frequency master clock, AHCLKR, is also sourced internally, the HCLKRM bit (AHCLKRCTL register) must be set to 1. Thus, the clock is divided down by a programmable high-clock divider (HCLKRDIV bit field in AHCLKRCTL register) from the MCASP internal clock source AUXCLK.

The receive high-frequency master clock, AHCLKR, may be (but is not required to be) output on the AHCLKR pin where it is available to other devices in the system. Regardless if AHCLKR is either internally generated or externally sourced, polarity of the high-frequency clock may be programmed via bit HCLKRP to be either rising or falling edge.

**Figure 16-17. Receive Clock Generator Block Diagram**




(1) Refer to the device-specific date manual for multiplexing options.


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.2.2.3 Frame Sync Generator

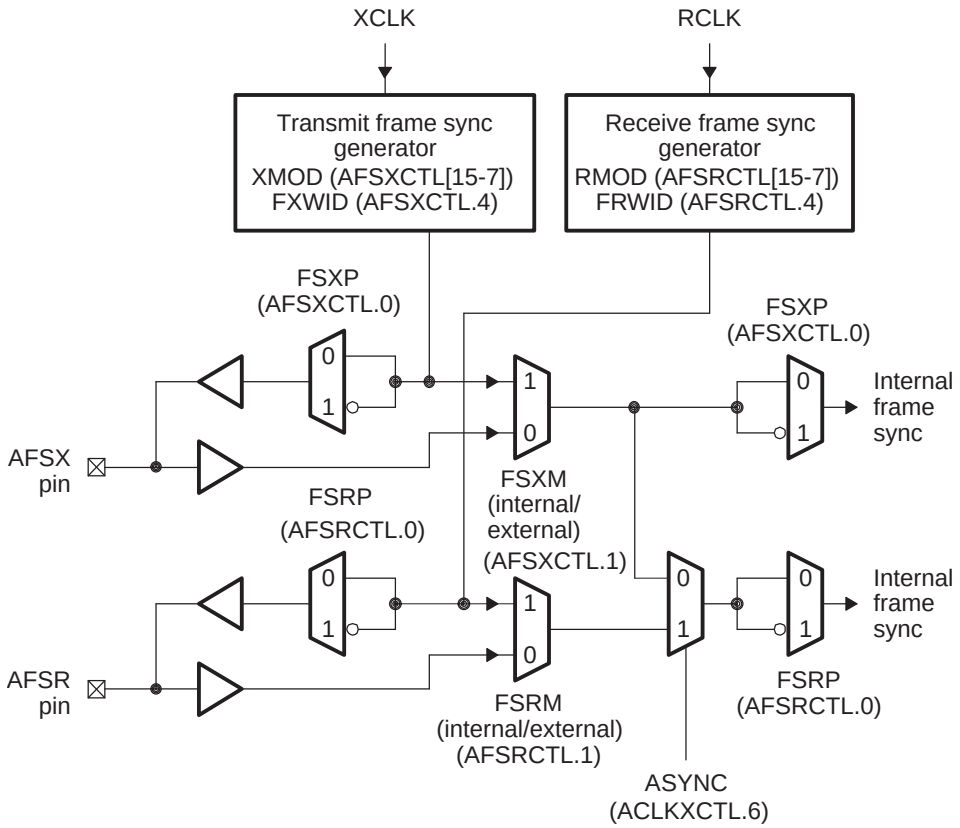
There are two different modes for frame sync: burst and TDM. A block diagram of the frame sync generator is shown in Figure 16-18. The frame sync options are programmed by the receive and transmit frame sync control registers (AFSRCTL and AFSXCTL). The options are:

- Internally-generated or externally-generated via configuring bit AFSXCTL[1] FSXM for transmit or AFSRCTL[1] FSRM for receive logic
- Frame sync polarity: rising edge or falling edge via configuring bit AFSXCTL[0] FSXP for transmit or AFSRCTL[0] FSRP for receive logic
- Frame sync width: single bit or single word via configuring bit AFSXCTL[4] FXWID for transmit or AFSRCTL[4] FRWID for receive logic
- Bit delay: 0, 1, or 2 cycles before the first data bit
- Selecting the source (AFSX or AFSR) of receiver internal frame synchronization. This is done in the same bit, ACLKXCTL[6] ASYNC, used to define the receiver internal clock source.

The transmit frame sync pin is AFSX and the receive frame sync pin is AFSR. A typical usage for these pins is to carry the left/right clock (LRCLK) signal when transmitting and receiving stereo data.

Regardless if the AFSX/AFSR is internally generated or externally sourced, the polarity of AFSX/AFSR is determined by FSXP/FSRP, respectively, to be either rising or falling edge. If FSXP/FSRP = 0, the frame sync polarity is rising edge. If FSRP/FSRP = 1, the frame sync polarity is falling edge.

**Figure 16-18. Frame Sync Generator Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.2.4 Clocking Examples

Some examples of processes using the McASP clocking and frame flexibility are:

- Receive data from a DVD at 48 kHz, but output up-sampled or decoded audio at 96 kHz or 192 kHz. This could be accomplished by inputting a high-frequency master clock (for example, 512 × receive FS), receiving with an internally-generated bit clock ratio of divide-by-8, and transmitting with an internally-generated bit clock ratio of divide-by-4 or divide-by-2.
- Transmit/receive data based on one sample rate (for example, 44.1 kHz), and transmit/receive data at a different sample rate (for example, 48 kHz).

### 16.2.3 Memory Map

See the device-specific data manual for information describing the device memory-map.

### 16.2.4 Signal Descriptions

The signals used on the McASP audio interface are listed in Table 16-3.

**Table 16-3. McASP Interface Signals**

| Pin[1] | I/O/Z | Device Reset (RESET = 0) | Description |
|---|---|---|---|
| | | **Transmitter Control** | |
| AHCLKX | I/O/Z | Input | Transmit high-frequency master clock |
| AFSX | I/O/Z | Input | Transmit frame sync or left/right clock (LRCLK) |
| ACLKX | I/O/Z | Input | Transmit bit clock |
| | | **Receiver Control** | |
| AHCLKR | I/O/Z | Input | Receive high-frequency master clock |
| AFSR | I/O/Z | Input | Receive frame sync or left/right clock (LRCLK) |
| ACLKR | I/O/Z | Input | Receive bit clock |
| | | **Mute** | |
| AMUTE | I/O/Z | Input | Mute output |
| AMUTEIN | I/O/Z | Input | Mute input |
| | | **Data** | |
| AXRn | I/O/Z | Input | TX/RX data pins. Up to 16 data pins are supported (n = 0 to 15). |

(1) Due to pinout restrictions, not all interface signals are supported on all devices or all McASP instances. Refer to your device-specific data manual for details.

### 16.2.5 Pin Multiplexing

The McASP signals share pins with other processor functions. For detailed information on the McASP pin multiplexing and configuration, see the pin multiplexing information in the device-specific data manual.


Copyright © 2011–2016, Texas Instruments Incorporated

## 16.2.6 Transfer Modes

### 16.2.6.1 Burst Transfer Mode

The McASP supports a burst transfer mode, which is useful for nonaudio data such as passing control information between two processors. Burst transfer mode uses a synchronous serial format similar to the TDM mode. The frame sync generation is not periodic or time-driven as in TDM mode, but data driven, and the frame sync is generated for each data word transferred.

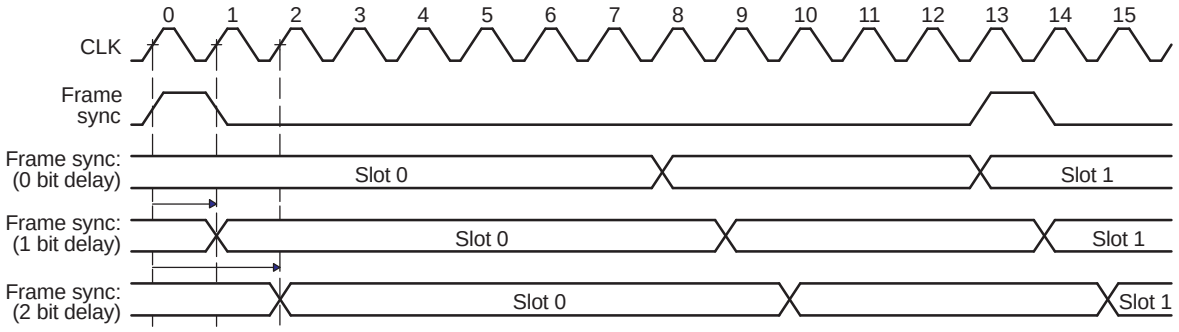
When operating in burst frame sync mode (Figure 16-19), as specified for transmit (XMOD = 0 in AFSXCTL) and receive (RMOD = 0 in AFSRCTL), one slot is shifted for each active edge of the frame sync signal that is recognized. Additional clocks after the slot and before the next frame sync edge are ignored.

In burst frame sync mode, the frame sync delay may be specified as 0, 1, or 2 serial clock cycles. This is the delay between the frame sync active edge and the start of the slot. The frame sync signal lasts for a single bit clock duration (FRWID = 0 in AFSRCTL, FXWID = 0 in AFSXCTL).

For transmit, when generating the transmit frame sync internally, the frame sync begins when the previous transmission has completed and when all the XBUF[n] (for every serializer set to operate as a transmitter) has been updated with new data.

For receive, when generating the receive frame sync internally, frame sync begins when the previous transmission has completed and when all the RBUF[n] (for every serializer set to operate as a receiver) has been read.

**Figure 16-19. Burst Frame Sync Mode**




The control registers must be configured as follows for the burst transfer mode. The burst mode specific bit fields are in bold face:

- PFUNC: The clock, frame, data pins must be configured for McASP function.
- PDIR: The clock, frame, data pins must be configured to the direction desired.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 16.2.10.2 to configure this register.
- AMUTE: Not applicable. Leave at default.
- DLBCTL: If loopback mode is desired, configure this register according to Section 16.2.8.5, otherwise leave this register at default.
- DITCTL: DITEN must be left at default 0 to select non-DIT mode. Leave the register at default.
- RMASK/XMASK: Mask desired bits according to Section 16.2.7.2 and Section 16.2.8.3.
- RFMT/XFMT: Program all fields according to data format desired. See Section 16.2.8.3.
- AFSRCTL/AFSXCTL: Clear **RMOD/XMOD** bits to 0 to indicate burst mode. Clear **FRWID/FXWID** bits to 0 for single bit frame sync duration. Configure other fields as desired.
- ACLKRCTL/ACLKXCTL: Program all fields according to bit clock desired. See Section 16.2.2.
- AHCLKRCTL/AHCLKXCTL: Program all fields according to high-frequency clock desired. See Section 16.2.2.


Copyright © 2011–2016, Texas Instruments Incorporated

- RTDM/XTDM: Program RTDMS0/XTDMS0 to 1 to indicate one active slot only. Leave other fields at default.
- RINTCTL/XINTCTL: Program all fields according to interrupts desired.
- RCLKCHK/XCLKCHK: Not applicable. Leave at default.
- SRCTLn: Program SRMOD to inactive/transmitter/receiver as desired. DISMOD is not applicable and should be left at default.
- DITCSRA[n], DITCSRB[n], DITUDRA[n], DITUDRB[n]: Not applicable. Leave at default.

#### 16.2.6.2 Time-Division Multiplexed (TDM) Transfer Mode

The McASP time-division multiplexed (TDM) transfer mode supports the TDM format discussed in Section 16.1.6.1.

Transmitting data in the TDM transfer mode requires a minimum set of pins:

- ACLKX - transmit bit clock.
- AFSX - transmit frame sync (or commonly called left/right clock).
- One or more serial data pins, AXRn, whose serializers have been configured to transmit.

The transmitter has the option to receive the ACLKX bit clock as an input, or to generate the ACLKX bit clock by dividing down the AHCLKX high-frequency master clock. The transmitter can either generate AHCLKX internally or receive AHCLKX as an input. See Section 16.2.2.1.

Similarly, to receive data in the TDM transfer mode requires a minimum set of pins:

- ACLKR - receive bit clock.
- AFSR - receive frame sync (or commonly called left/right clock).
- One or more serial data pins, AXRn, whose serializers have been configured to receive.

The receiver has the option to receive the ACLKR bit clock as an input or to generate the ACLKR bit clock by dividing down the AHCLKR high-frequency master clock. The receiver can either generate AHCLKR internally or receive AHCLKR as an input. See Section 16.2.2.2 and Section 16.2.2.3.

The control registers must be configured as follows for the TDM mode. The TDM mode specific bit fields are in bold face:

- PFUNC: The clock, frame, data pins must be configured for McASP function.
- PDIR: The clock, frame, data pins must be configured to the direction desired.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 16.2.10.2 to configure this register.
- AMUTE: Program all fields according to mute control desired.
- DLBCTL: If loopback mode is desired, configure this register according to Section 16.2.8.5, otherwise leave this register at default.
- DITCTL: DITEN must be left at default 0 to select TDM mode. Leave the register at default.
- RMASK/XMASK: Mask desired bits according to Section 16.2.7.2 and Section 16.2.8.3.
- RFMT/XFMT: Program all fields according to data format desired. See Section 16.2.8.3.
- AFSRCTL/AFSXCTL: Set **RMOD/XMOD** bits to 2-32 for TDM mode. Configure other fields as desired.
- ACLKRCTL/ACLKXCTL: Program all fields according to bit clock desired. See Section 16.2.2.
- AHCLKRCTL/AHCLKXCTL: Program all fields according to high-frequency clock desired. See Section 16.2.2.
- RTDM/XTDM: Program all fields according to the time slot characteristics desired.
- RINTCTL/XINTCTL: Program all fields according to interrupts desired.
- RCLKCHK/XCLKCHK: Program all fields according to clock checking desired.
- SRCTLn: Program all fields according to serializer operation desired.
- DITCSRA[n], DITCSRB[n], DITUDRA[n], DITUDRB[n]: Not applicable. Leave at default.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.6.2.1 TDM Time Slots

TDM mode on the McASP can extend to support multiprocessor applications, with up to 32 time slots per frame. For each of the time slots, the McASP may be configured to participate or to be inactive by configuring XTDM and/or RTDM (this allows multiple processors to communicate on the same TDM serial bus).

The TDM sequencer (separate ones for transmit and receive) functions in this mode. The TDM sequencer counts the slots beginning with the frame sync. For each slot, the TDM sequencer checks the respective bit in either XTDM or RTDM to determine if the McASP should transmit/receive in that time slot.

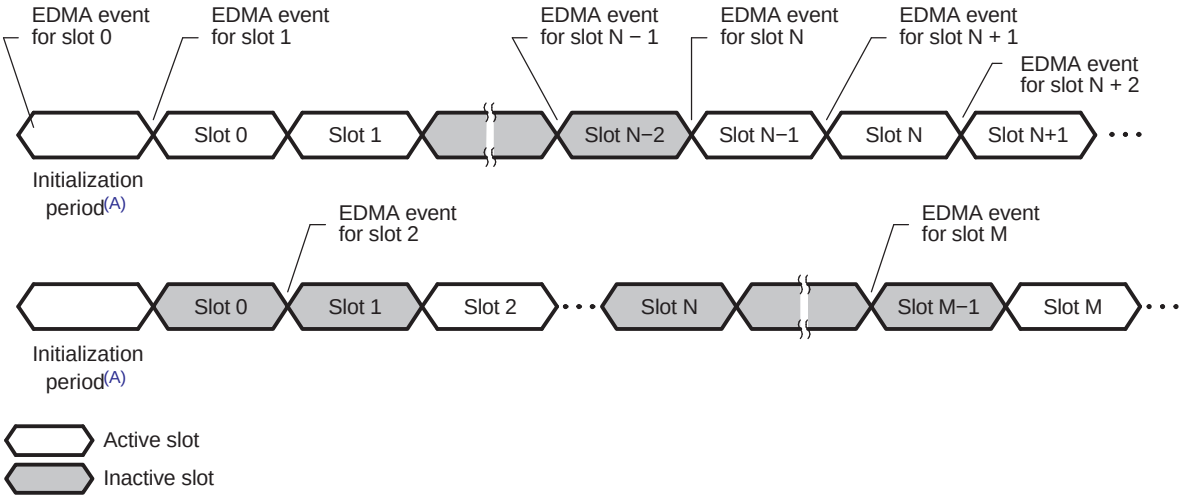
If the transmit/receive bit is active, the McASP functions normally during that time slot; otherwise, the McASP is inactive during that time slot; no update to the buffer occurs, and no event is generated. Transmit pins are automatically set to a high-impedance state, 0, or 1 during that slot, as determined by bit DISMOD in SRCTL[n].

Figure 16-20 shows when the transmit DMA event AXEVT is generated. See Section 16.2.8.1.1 for details on data ready and the initialization period indication. The transmit DMA event for an active time slot (slot N) is generated during the previous time slot (slot N - 1), regardless if the previous time slot (slot N - 1) is active or inactive.

During an active transmit time slot (slot N), if the next time slot (slot N + 1) is configured to be active, the copy from XRBUF[n] to XRSR[n] generates the DMA event for time slot N + 1. If the next time slot (slot N + 1) is configured to be inactive, then the DMA event will be delayed to time slot M - 1. In this case, slot M is the next active time slot. The DMA event for time slot M is generated during the first bit time of slot M - 1.

The receive DMA request generation does not need this capability, since the receive DMA event is generated after data is received in the buffer (looks back in time). If a time slot is disabled, then no data is copied to the buffer for that time slot and no DMA event is generated.

**Figure 16-20. Transmit DMA Event (AXEVT) Generation in TDM Time Slots**




A See Section 16.2.10.2, step 7a.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.6.2.2 Special 384 Slot TDM Mode for Connection to External DIR

The McASP receiver also supports a 384 time slot TDM mode (DIR mode), to support S/PDIF, AES-3, IEC-60958 receiver ICs whose natural block (block corresponds to McASP frame) size is 384 samples. The advantage to using the 384 time slot TDM mode is that interrupts may be generated synchronous to the S/PDIF, AES-3, IEC-60958, such as the last slot interrupt.

The receive TDM time slot register (RTDM) should be programmed to all 1s during reception of a DIR block. Other TDM functionalities (for example, inactive slots) are not supported (only the slot counter counts the 384 subframes in a block).

To receive data in the DIR mode, the following pins are typically needed:

- ACLKR - receive bit clock.
- AFSR - receive frame sync (or commonly called left/right clock). In this mode, AFSR should be connected to a DIR which outputs a start of block signal, instead of LRCLK.
- One or more serial data pins, AXRn, whose serializers have been configured to receive.

For this special DIR mode, the control registers can be configured just as for TDM mode, except set RMOD in AFSRCTL to 384 to receive 384 time slots.

### 16.2.6.3 Digital Audio Interface Transmit (DIT) Transfer Mode

In addition to the TDM and burst transfer modes, which are suitable for transmitting audio data between ICs inside the same system, the digital audio interface transmit (DIT) transfer mode of the McASP also supports transmission of audio data in the S/PDIF, AES-3, or IEC-60958 format. These formats are designed to carry audio data between different systems through an optical or coaxial cable. The DIT mode only applies to serializers configured as transmitters, not receivers. Refer to Section 16.1.6.2 for a description of the S/PDIF format.

#### 16.2.6.3.1 Transmit DIT Encoding

The McASP operation in DIT mode is basically identical to the 2 time slot TDM mode, but the data transmitted is output as a biphase mark encoded bit stream, with preamble, channel status, user data, validity, and parity automatically stuffed into the bit stream by the McASP. The McASP includes separate validity bits for even/odd subframes and two 384-bit RAM modules to hold channel status and user data bits.

The transmit TDM time slot register (XTDM) should be programmed to all 1s during DIT mode. TDM functionality is not supported in DIT mode, except that the TDM slot counter counts the DIT subframes.

To transmit data in the DIT mode, the following pins are typically needed:

- AHCLKX - transmit high-frequency master clock.
- One or more serial data pins, AXRn, whose serializers have been configured to transmit.

AHCLKX is optional (the internal clock source may be used instead), but if used as a reference, the processor provides a clock check circuit that continually monitors the AHCLKX input for stability.

If the McASP is configured to transmit in the DIT mode on more than one serial data pin, the bit streams on all pins will be synchronized. In addition, although they will carry unique audio data, they will carry the same channel status, user data, and validity information.

The actual 24-bit audio data must always be in bit positions 23-0 after passing through the first three stages of the transmit format unit.


Copyright © 2011–2016, Texas Instruments Incorporated

For left-aligned Q31 data, the following transmit format unit settings process the data into right aligned 24-bit audio data ready for transmission:

- XROT = 010 (rotate right by 8 bits).
- XRVRS = 0 (no bit reversal, LSB first).
- XMASK = FFFF FF00h-FFFF 0000h (depending upon whether 24, 23, 22, 21, 20, 19, 18, 17, or 16 valid audio data bits are present).
- XPAD = 00 (pad extra bits with 0).

For right-aligned data, the following transmit format unit settings process the data into right aligned 24-bit audio data ready for transmission:

- XROT = 000 (rotate right by 0 bits).
- XRVRS = 0 (no bit reversal, LSB first).
- XMASK = 00FF FFFFh to 0000 FFFFh (depending upon whether 24, 23, 22, 21, 20, 19, 18, 17, or 16 valid audio data bits are present).
- XPAD = 00 (pad extra bits with 0).

#### *16.2.6.3.2 Transmit DIT Clock and Frame Sync Generation*

The DIT transmitter only works in the following configuration:

- In transmit frame control register (AFSXCTL):
  - Internally-generated transmit frame sync, FSXM = 1.
  - Rising-edge frame sync, FSXP = 0.
  - Bit-width frame sync, FXWID = 0.
  - 384-slot TDM, XMOD = 1 1000 0000b.
- In transmit clock control register (ACLKXCTL), ASYNC = 1.
- In transmit bitstream format register (XFMT), XSSZ = 1111 (32-bit slot size).

All combinations of AHCLKX and ACLKX are supported.

This is a summary of the register configurations required for DIT mode. The DIT mode specific bit fields are in bold face:

- PFUNC: The data pins (AXRn) must be configured for McASP function. If AHCLKX is used, it must also be configured for McASP function.
- PDIR: The data pins (AXRn) must be configured as outputs. If AHCLKX is used as an input reference, it should be configured as input. If internal clock source AUXCLK is used as the reference clock, it may be output on the AHCLKX pin by configuring AHCLKX as an output.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable for DIT operation. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 16.2.10.2 to configure this register.
- AMUTE: Program all fields according to mute control desired.
- DLBCTL: Not applicable. Loopback is not supported for DIT mode. Leave at default.
- DITCTL: **DITEN** bit must be set to 1 to enable DIT mode. Configure other bits as desired.
- RMASK: Not applicable. Leave at default.
- RFMT: Not applicable. Leave at default.
- AFSRCTL: Not applicable. Leave at default.
- ACLKRCTL: Not applicable. Leave at default.
- AHCLKRCTL: Not applicable. Leave at default.
- RTDM: Not applicable. Leave at default.
- RINTCTL: Not applicable. Leave at default.
- RCLKCHK: Not applicable. Leave at default.
- **XMASK**: Mask desired bits according to the discussion in this section, depending upon left-aligned or right-aligned internal data.


Copyright © 2011–2016, Texas Instruments Incorporated

- **XFMT: XDATDLY** = 0. **XRVRS** = 0. **XPAD** = 0. **XPBIT** = default (not applicable). **XSSZ** = Fh (32-bit slot). XBUSEL = configured as desired. **XROT** bit is configured according to the discussion in this section, either 0 or 8-bit rotate.
- **AFSXCTL**: Configure the bits according to the discussion in this section.
- **ACLKXCTL**: **ASYNC** = 1. Program CLKXDIV bits to obtain the bit clock rate desired. Configure CLKXP and CLKXM bits as desired, because CLKX is not actually used in the DIT protocol.
- **AHCLKXCTL**: Program all fields according to high-frequency clock desired.
- **XTDM**: Set to FFFF FFFFh for all active slots for DIT transfers.
- XINTCTL: Program all fields according to interrupts desired.
- XCLKCHK: Program all fields according to clock checking desired.
- SRCTLn: Set **SRMOD** = 1 (transmitter) for the DIT pins. DISMOD field is don't care for DIT mode.
- **DITCSRA[n], DITCSRB[n]**: Program the channel status bits as desired.
- **DITUDRA[n], DITUDRB[n]**: Program the user data bits as desired.

#### *16.2.6.3.3 DIT Channel Status and User Data Register Files*

The channel status registers (DITCSRA*n* and DITCSRB*n*) and user data registers (DITUDRA*n* and DITUDRB*n*) are not double buffered. Typically the programmer uses one of the synchronizing interrupts, such as last slot, to create an event at a safe time so the register may be updated. In addition, the CPU reads the transmit TDM slot counter to determine which word of the register is being used.

It is a requirement that the software avoid writing to the word of user data and channel status that are being used to encode the current time slot; otherwise, it will be indeterminate whether the old or new data is used to encode the bitstream.

The DIT subframe format is defined in Section 16.1.6.2.2. The channel status information (C) and User Data (U) are defined in these DIT control registers:

- DITCSRA0 to DITCSRA5: The 192 bits in these six registers contain the channel status information for the LEFT channel within each frame.
- DITCSRB0 to DITCSRB5: The 192 bits in these six registers contain the channel status information for the RIGHT channel within each frame.
- DITUDRA0 to DITUDRA5: The 192 bits in these six registers contain the user data information for the LEFT channel within each frame.
- DITUDRB0 to DITUDRB5: The 192 bits in these six registers contain the user data information for the RIGHT channel within each frame.

The S/PDIF block format is shown in Figure 16-12. There are 192 frames within a block (frame 0 to frame 191). Within each frame there are two subframes (subframe 1 and 2 for left and right channels, respectively). The channel status and user data information sent on each subframe is summarized in Table 16-4.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-4. Channel Status and User Data for Each DIT Block**

| Frame | Subframe | Preamble | Channel Status defined in: | User Data defined in: |
|---|---|---|---|---|
| **Defined by DITCSRA0, DITCSRB0, DITUDRA0, DITUDRB0** | | | | |
| 0 | 1 (L) | B | DITCSRA0[0] | DITUDRA0[0] |
| 0 | 2 (R) | W | DITCSRB0[0] | DITUDRB0[0] |
| 1 | 1 (L) | M | DITCSRA0[1] | DITUDRA0[1] |
| 1 | 2 (R) | W | DITCSRB0[1] | DITUDRB0[1] |
| 2 | 1 (L) | M | DITCSRA0[2] | DITUDRA0[2] |
| 2 | 2 (R) | W | DITCSRB0[2] | DITUDRB0[2] |
| … | … | … | … | … |
| 31 | 1 (L) | M | DITCSRA0[31] | DITUDRA0[31] |
| 31 | 2 (R) | W | DITCSRB0[31] | DITUDRB0[31] |
| **Defined by DITCSRA1, DITCSRB1, DITUDRA1, DITUDRB1** | | | | |
| 32 | 1 (L) | M | DITCSRA1[0] | DITUDRA1[0] |
| 32 | 2 (R) | W | DITCSRB1[0] | DITUDRB1[0] |
| … | … | … | … | … |
| 63 | 1 (L) | M | DITCSRA1[31] | DITUDRA1[31] |
| 63 | 2 (R) | W | DITCSRB1[31] | DITUDRB1[31] |
| **Defined by DITCSRA2, DITCSRB2, DITUDRA2, DITUDRB2** | | | | |
| 64 | 1 (L) | M | DITCSRA2[0] | DITUDRA2[0] |
| 64 | 2 (R) | W | DITCSRB2[0] | DITUDRB2[0] |
| … | … | … | … | … |
| 95 | 1 (L) | M | DITCSRA2[31] | DITUDRA2[31] |
| 95 | 2 (R) | W | DITCSRB2[31] | DITUDRB2[31] |
| **Defined by DITCSRA3, DITCSRB3, DITUDRA3, DITUDRB3** | | | | |
| 96 | 1 (L) | M | DITCSRA3[0] | DITUDRA3[0] |
| 96 | 2 (R) | W | DITCSRB3[0] | DITUDRB3[0] |
| … | … | … | … | … |
| 127 | 1 (L) | M | DITCSRA3[31] | DITUDRA3[31] |
| 127 | 2 (R) | W | DITCSRB3[31] | DITUDRB3[31] |
| **Defined by DITCSRA4, DITCSRB4, DITUDRA4, DITUDRB4** | | | | |
| 128 | 1 (L) | M | DITCSRA4[0] | DITUDRA4[0] |
| 128 | 2 (R) | W | DITCSRB4[0] | DITUDRB4[0] |
| … | … | … | … | … |
| 159 | 1 (L) | M | DITCSRA4[31] | DITUDRA4[31] |
| 159 | 2 (R) | W | DITCSRB4[31] | DITUDRB4[31] |
| **Defined by DITCSRA5, DITCSRB5, DITUDRA5, DITUDRB5** | | | | |
| 160 | 1 (L) | M | DITCSRA5[0] | DITUDRA5[0] |
| 160 | 2 (R) | W | DITCSRB5[0] | DITUDRB5[0] |
| … | … | … | … | … |
| 191 | 1 (L) | M | DITCSRA5[31] | DITUDRA5[31] |
| 191 | 2 (R) | W | DITCSRB5[31] | DITUDRB5[31] |


Copyright © 2011–2016, Texas Instruments Incorporated

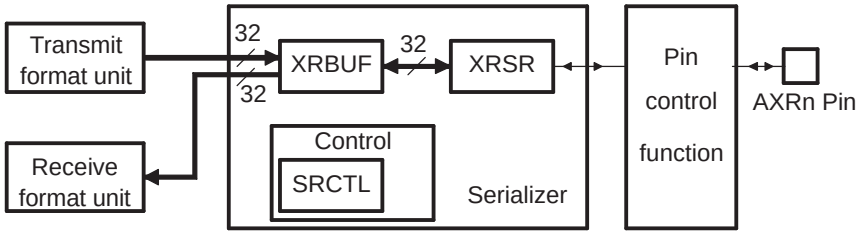
## 16.2.7 General Architecture

### 16.2.7.1 Serializers

Figure 16-21 shows the block diagram of the serializer and its interface to other units within the McASP. The serializers take care of shifting serial data in and out of the McASP. Each serializer consists of a shift register (XRSR), a data buffer (XRBUF), a control register (SRCTL), and logic to support the data alignment options of the McASP. For each serializer (XRSRn), there is a dedicated serial data pin (AXRn) and a dedicated control register (SRCTL[n]). The control register allows the serializer to be configured as a transmitter, receiver, or as inactive. When configured as a transmitter the serializer shifts out data to the serial data pin AXRn. When configured as a receiver, the serializer shifts in data from the AXRn pin. The serializer is clocked from the transmit/receive section clock (ACLKX/ACLKR) if configured to transmit/receive respectively.

All serializers that are configured to transmit operate in lock-step. Similarly, all serializers that are configured to receive also operate in lock-step. This means that at most there are two zones per McASP, one for transmit and one for receive.

**Figure 16-21. Individual Serializer and Connections Within McASP**




For receive, data is shifted in through the AXRn pin to the shift register XRSR. Once the entire slot of data is collected in the XRSR, the data is copied to the data buffer XRBUF. The data is now ready to be read by the processor through the RBUF register, which is an alias of the XRBUF for receive. When the processor reads from the RBUF, the McASP passes the data from RBUF through the receive format unit and returns the formatted data to the processor.

For transmit, the processor services the McASP by writing data into the XBUF register, which is an alias of the XRBUF for transmit. The data automatically passes through the transmit format unit before actually reaching the XRBUF register in the serializer. The data is then copied from XRBUF to XRSR, and shifted out from the AXRn synchronously to the serial clock.

In DIT mode, in addition to the data, the serializer shifts out other DIT-specific information accordingly (preamble, user data, etc.).

The serializer configuration is controlled by SRCTL[n]. A serializer n is configured as inactive via setting bitfield SRCTLn[1:0] SRMOD to 0. For a transmitting serializer, the SRCTLn[3:2] DISMOD bitfield defines the AXRn pin output state during inactive slots (logic high, logic low, or 3-state). A serializer n is configured in a transmit function via setting bitfield SRCTLn[1:0] SRMOD to 1. A serializer n is configured in a receive function via setting bitfield SRCTLn[1:0] SRMOD to 2h.


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.2.7.2 Format Unit

The McASP has two data formatting units, one for transmit and one for receive. These units automatically remap the data bits within the transmitted and received words between a natural format for the processor (such as a Q31 representation) and the required format for the external serial device (such as "I2S format"). During the remapping process, the format unit also can mask off certain bits or perform sign extension.

Since all transmitters share the same data formatting unit, the McASP only supports one transmit format at a time. For example, the McASP will not transmit in "I2S format" on serializer 0, while transmitting "Left Justified" on serializer 1. Likewise, the receiver section of the McASP only supports one data format at a time, and this format applies to all receiving serializers. However, the McASP can transmit in one format while receiving in a completely different format.

This formatting unit consists of three stages:

- Bit mask and pad (masks off bits, performs sign extension)
- Rotate right (aligns data within word)
- Bit reversal (selects between MSB first or LSB first)

The MCASP transmitter supports serial formats of:

- Slot (or Time slot) size = 8, 12, 16, 20, 24, 28, 32 bits
- Word size ≤ Slot size
- Alignment: when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot
- Order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB
  - LSB: least-significant bit of word is shifted out last, last bit is MSB

Hardware support for these serial formats comes from the programmable options in the bitstream format register, (R/X)FMT:

- XRVRS: bit reverse (1) or no bit reverse (0)
- XROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits
- XSSZ: receive slot size of 8, 12, 16, 20, 24, 28, or 32 bits

Figure 16-22 shows a block diagram of the receive formatting unit, and Figure 16-23 shows the transmit formatting unit. Note that the order in which data flows through the three stages is different between the transmit and receive formatting units.

As shown in Figure 16-23, the data to the transmit format unit (TFU) can come from the configuration port (CFG) or the data port (DATA). The selection is made through the RFMT[3] RBUSEL bit. According to port type selected, data transfer has different behavior.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 16-22. Receive Format Unit**

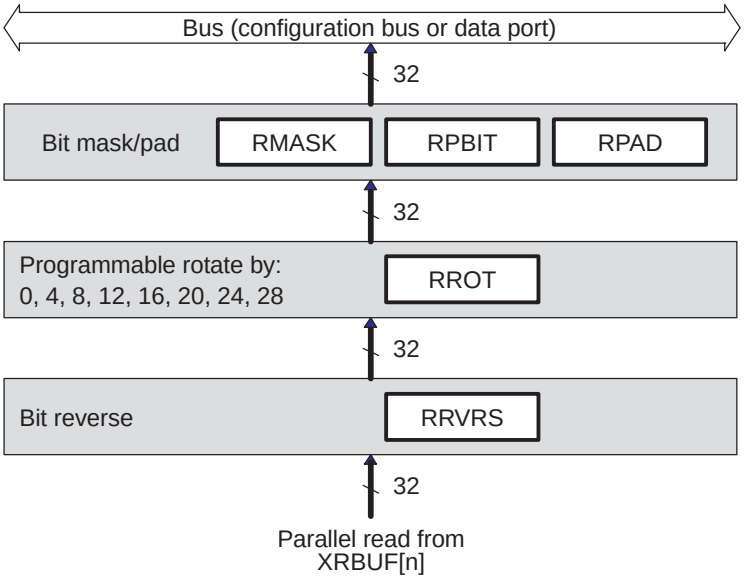
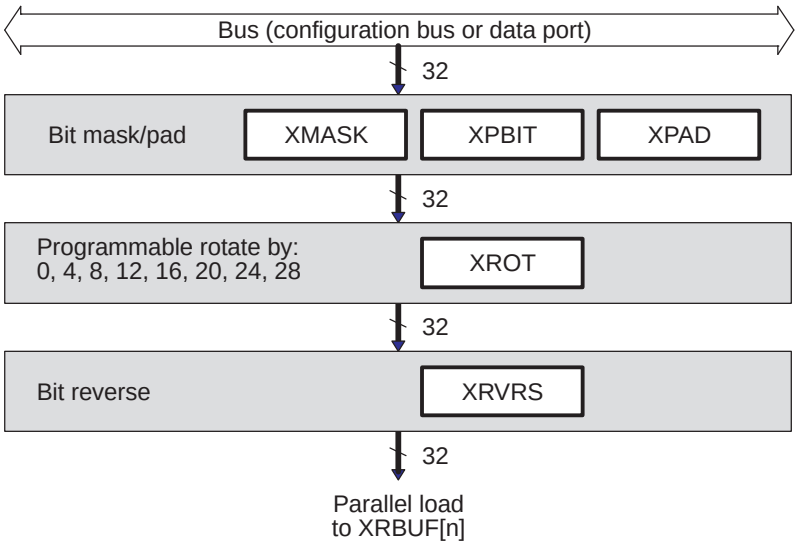



**Figure 16-23. Transmit Format Unit**





Copyright © 2011–2016, Texas Instruments Incorporated

The bit mask and pad stage includes a full 32-bit mask register, allowing selected individual bits to either pass through the stage unchanged, or be masked off. The bit mask and pad then pad the value of the masked off bits by inserting either a 0, a 1, or one of the original 32 bits as the pad value. The last option allows for sign-extension when the sign bit is selected to pad the remaining bits.

The rotate right stage performs bitwise rotation by a multiple of 4 bits (between 0 and 28 bits), programmable by the (R/X)FMT register. Note that this is a rotation process, not a shifting process, so bit 0 gets shifted back into bit 31 during the rotation.

The bit reversal stage either passes all 32 bits directly through, or swaps them. This allows for either MSB or LSB first data formats. If bit reversal is not enabled, then the McASP will naturally transmit and receive in an LSB first order.

Finally, note that the (R/X)DATDLY bits in (R/X)FMT also determine the data format. For example, the difference between I2S format and left-justified is determined by the delay between the frame sync edge and the first data bit of a given time slot. For I2S format, (R/X)DATDLY should be set to a 1-bit delay, whereas for left-justified format, it should be set to a 0-bit delay.

The combination of all the options in (R/X)FMT means that the McASP supports a wide variety of data formats, both on the serial data lines, and in the internal processor representation.

Section 16.2.8.3 provides more detail and specific examples. The examples use internal representation in integer and Q31 notation, but other fractional notations are also possible.

#### 16.2.7.3 State Machine

The receive and transmit sections have independent state machines. Each state machine controls the interactions between the various units in the respective section. In addition, the state machine keeps track of error conditions and serial port status.

No serial transfers can occur until the respective state machine is released from reset. See initialization sequence for details (Section 16.2.10).

The receive state machine is controlled by the RFMT register, and it reports the McASP status and error conditions in the RSTAT register. Similarly, the transmit state machine is controlled by the XFMT register, and it reports the McASP status and error conditions in the XSTAT register.

#### 16.2.7.4 TDM Sequencer

There are separate TDM sequencers for the transmit section and the receive section. Each TDM sequencer keeps track of the slot count. In addition, the TDM sequencer checks the bits of (R/X)TDM and determines if the McASP should receive/transmit in that time slot.

If the McASP should participate (transmit/receive bit is active) in the time slot, the McASP functions normally. If the McASP should not participate (transmit/receive bit is inactive) in the time slot, no transfers between the XRBUF and XRSR registers in the serializer would occur during that time slot. In addition, the serializers programmed as transmitters place their data output pins in a predetermined state (logic low, high, or high impedance) as programmed by each serializer control register (SRCTL). Refer also to Section 16.2.6.2 for details on how DMA event or interrupt generations are handled during inactive time slots in TDM mode.

The receive TDM sequencer is controlled by register RTDM and reports current receive slot to RSLOT. The transmit TDM sequencer is controlled by register XTDM and reports current transmit slot to XSLOT.

#### 16.2.7.5 Clock Check Circuit

A common source of error in audio systems is a serial clock failure due to instabilities in the off-chip DIR circuit. To detect a clock error quickly, a clock-check circuit is included in the McASP for both transmit and receive clocks, since both may be sourced from off chip.

The clock check circuit can detect and recover from transmit and receive clock failures. See Section 16.2.8.4.6 for implementation and programming details.


Copyright © 2011–2016, Texas Instruments Incorporated

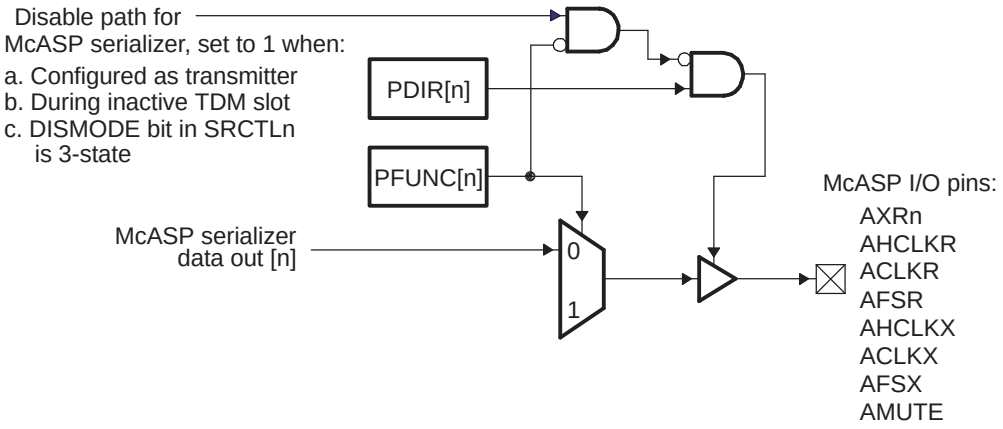
### 16.2.7.6 Pin Function Control

All McASP pins except AMUTEIN are bidirectional input/output pins. In addition, these bidirectional pins function either as McASP or general-purpose I/O (GPIO) pins. The following registers control the pin functions:

- Pin function register (PFUNC): selects pin to function as McASP or GPIO.
- Pin direction register (PDIR): selects pin to be input or output.
- Pin data input register (PDIN): shows data input at the pin.
- Pin data output register (PDOUT): data to be output at the pin if the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1). Not applicable when the pin is configured as McASP pin (PFUNC[n] = 0).
- Pin data set register (PDSET): alias of PDOUT. Writing a 1 to PDSET[n] sets the respective PDOUT[n] to 1. Writing a 0 has no effect. Applicable only when the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1).
- Pin data clear register (PDCLR): alias of PDOUT. Writing a 1 to PDCLR[n] clears the respective PDOUT[n] to 0. Writing a 0 has no effect. Applicable only when the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1).

See the register descriptions in Section 16.3 for details on the mapping of each McASP pin to the register bits. Figure 16-24 shows the pin control block diagram.

**Figure 16-24. McASP I/O Pin Control Block Diagram**




#### *16.2.7.6.1 McASP Pin Control-Transmit and Receive*

You must correctly set the McASP GPIO registers PFUNC and PDIR, even when McASP pins are used for their serial port (non-GPIO) function.

Serial port functions include:

- Clock pins (ACLKX, ACLKR, AHCLKX, AHCLKR, AFSX, AFSR) used as clock inputs and outputs.
- Serializer data pins (AXRn) used to transmit or receive.
- AMUTE used as a mute output signal.

When using these pins in their serial port function, you must clear PFUNC[n] to 0 for each pin.

Also, certain outputs require PDIR[n] = 1, such as clock pins used as clock outputs, serializer data pins used to transmit, and AMUTE used as mute output.

Clock inputs and serializers configured to receive must have PDIR[n] = 0.

PFUNC and PDIR do not control the AMUTEIN signal, it is usually tied to a device level interrupt pin (consult device datasheet). If used as a mute input, this pin needs to be configured as an input in the appropriate peripheral.


Copyright © 2011–2016, Texas Instruments Incorporated

Finally, there is an important advantage to having separate control of pin direction (by PDIR), and the choice of internal versus external clocking (by CLKRM/CLKXM). Depending on the specific device and usage, you might select an external clock (CLKRM = 0), while enabling the internal clock divider, and the clock pin as an output in the PDIR register (PDIR[ACLKR] = 1). In this case, the bit clock is an output (PDIR[ACLKR] = 1) and, therefore, routed to the ACLKR pin. However, because CLKRM = 0, the bit clock is then routed back to the McASP module as an "external" clock source. This may result in less skew between the clock inside the McASP and the clock in the external device, thus producing more balanced setup and hold times for a particular system. As a result, this may allow a higher serial clock rate interface.

## 16.2.8 Operation

This section discusses the operation of the McASP.

### 16.2.8.1 Data Transmission and Reception

The processor services the McASP by writing data to the XBUF register(s) for transmit operations, and by reading data from the RBUF register(s) for receive operations. The McASP sets status flag and notifies the processor whenever data is ready to be serviced. Section 16.2.8.1.1 discusses data ready status in detail.

The XBUF and RBUF registers can be accessed through one of the two peripheral ports of the device:

- The data port (DAT): This port is dedicated for data transfers on the device.
- The configuration bus (CFG): This port is used for both data transfers and peripheral configuration control on the device.

Section 16.2.8.1.2 and Section 16.2.8.1.3 discuss how to perform transfers through the data port and the configuration bus.

Either the CPU or the DMA can be used to service the McASP through any of these two peripheral ports. The CPU and DMA usages are discussed in Section 16.2.8.1.4 and Section 16.2.12.2.

#### 16.2.8.1.1 Data Ready Status and Event/Interrupt Generation

##### 16.2.8.1.1.1 Transmit Data Ready

The transmit data ready flag XDATA bit in the XSTAT register reflects the status of the XBUF register. The XDATA flag is set when data is transferred from the XRBUF[n] buffers to the XRSR[n] shift registers, indicating that the XBUF is empty and ready to accept new data from the processor. The transmit data ready event is individually indicated per serializer in its corresponding control register SRCTLn[4] XRDY status bit. When this bit is set to 1, it notifies to the host that this serializer Tx buffer must be serviced (written). When XBUFn is written to by the host, the SRCTLn[4] XRDY is deasserted to 0. As XDATA global flag is an OR-event of all active serializers XRDY flags, it indicates to software the moment, when write service operation has to be initiated by the MCASP host (XDATA = 1). The XRDY flags have to be sequentially scanned by user software to determine which serializer TXBUFn register has to be currently written. Once all requested TXBUFn are written, the serializers control XRDY flags are cleared to 0. As a consequence, XDATA flag is deasserted to 0 to indicate to software that write operation is completed for all serializers. The global XDATA flag can be cleared when the XSTAT[5] XDATA bit is written to 1 or once TXBUFn registers of all the serializers, that have previously raised their XRDY flags, are written with corresponding active slot data by the host.

Whenever XDATA is set, an DMA event AXEVT is automatically generated to notify the DMA of the XBUF empty status. An interrupt AXINT is also generated if XDATA interrupt is enabled in the XINTCTL register (See Section 16.2.11.1 for details).

For DMA requests, the McASP does not require XSTAT to be read between DMA events. This means that even if XSTAT already has the XDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

Since all serializers act in lockstep, only one DMA event is generated to indicate that all active transmit serializers are ready to be written to with new data.

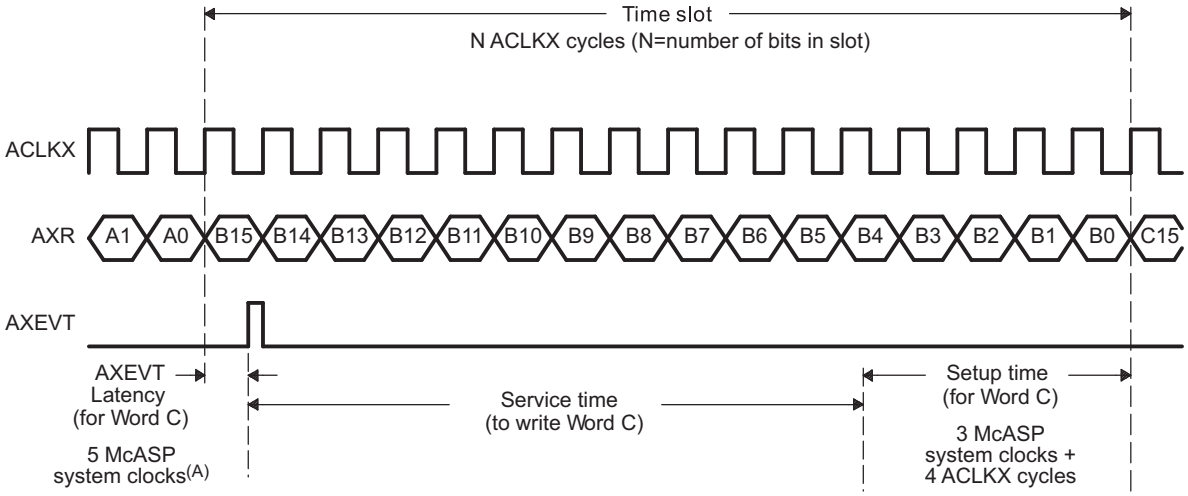

Copyright © 2011–2016, Texas Instruments Incorporated

Figure 16-25 shows the timing details of when AXEVT is generated at the McASP boundary. In this example, as soon as the last bit (bit A0) of Word A is transmitted, the McASP sets the XDATA flag and generates an AXEVT event. However, it takes up to 5 McASP system clocks (AXEVT Latency) before AXEVT is active at the McASP boundary. Upon AXEVT, the processor can begin servicing the McASP by writing Word C into the XBUF (Processor Service Time). The processor must write Word C into the XBUF no later than the setup time required by the McASP (Setup Time).

The maximum Processor Service Time (Figure 16-25) can be calculated as:

Processor Service Time = Time Slot - AXEVT Latency - Setup Time

**Figure 16-25. Processor Service Time Upon Transmit DMA Event (AXEVT)**




A Refer to the device-specific data manual for the McASP system clock source. This is not the same as AUXCLK.


Copyright © 2011–2016, Texas Instruments Incorporated

***Example 16-1. Processor Service Time Calculation for Transmit DMA Event (AXEVT)***

The following is an example to show how to calculate Processor Service Time. Assume the following setup:

- Device: C64x+ DSP at 300 MHz.
- McASP transmits in I2S format at 192 kHz frame rate. Assume slot size is 32 bit.

With the above setup, we obtain the following parameters corresponding to Figure 16-25:

- Calculation of McASP system clock cycle:
  - C64x+ DSP uses SYSCLK2 as the McASP system clock. It runs at 150 MHz (half of device frequency).
  - Therefore, McASP system clock cycle = 1/150 MHz = 6.7 ns.
- Calculation of ACLKX clock cycle:
  - This example has two 32-bit slots per frame, for a total of 64 bits per frame.
  - ACLKX clock cycle is (1/192 kHz)/64 = 81.4 ns.
- Time Slot between AXEVT events:
  - For I2S format, McASP generates two AXEVT events per 192 kHz frame.
  - Therefore, Time Slot between AXEVT events is (1/192 kHz)/2 = 2604 ns.
- AXEVT Latency:
  = 5 McASP system clocks
  = 6.7 ns × 5 = 33.5 ns
- Setup Time
  = 3 McASP system clocks + 4 ACLKX cycles
  = (6.7 ns × 3) + (81.4 ns × 4)
  = 345.7 ns
- Processor Service Time
  = Time Slot - AXEVT Latency - Setup Time
  = 2604 ns - 33.5 ns - 345.7 ns
  = 2225 ns


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.8.1.1.2 Receive Data Ready

Similarly, the receive data ready flag, RDATA, in the RSTAT register reflects the data ready status of XRBUFn buffers for all of the active slot receiving serializers. The RDATA flag is set whenever data is transferred from a receiving serializer shift register XRSRn to its corresponding XRBUFn data buffer. Thus, the RDATA bit indicates the global event that some of the receivers data buffer, RXBUFn, already contains received data (that is, a buffer is full) and is ready to transfer it to the host (MPU). The receive data ready event is individually indicated per serializer in its corresponding control register SRCTLn [5] RRDY status bit. When this bit is set to 1, it notifies to host that this serializer Rx buffer must be serviced (read). When RXBUF[n] is read from the host, the SRCTLn [5] RRDY is deasserted to 0. As RDATA global flag is an OR-event of all active serializers RRDY flags, it indicates to software the moment, when read service operation has to be initiated by the MCASP host (RDATA = 1). The RRDY flags have to be sequentially scanned by user software to determine which serializer RXBUFn register has to be currently read. Once all requested RXBUFn are read, the serializers control RRDY flags are cleared to 0. As a consequence, RDATA flag is deasserted to 0, to indicate to software that read operation is completed for all n serializers. The global RDATA flag can be cleared when the RSTAT[5] RDATA bit is written to 1 or once RXBUFn registers of all the serializers, that have previously raised their RRDY flags, are read by the host. (See Section 16.2.11.2 for details).

For DMA requests, the McASP does not require RSTAT to be read between DMA events. This means that even if RSTAT already has the RDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

Since all serializers act in lockstep, only one DMA event is generated to indicate that all active receive serializers are ready to receive new data.

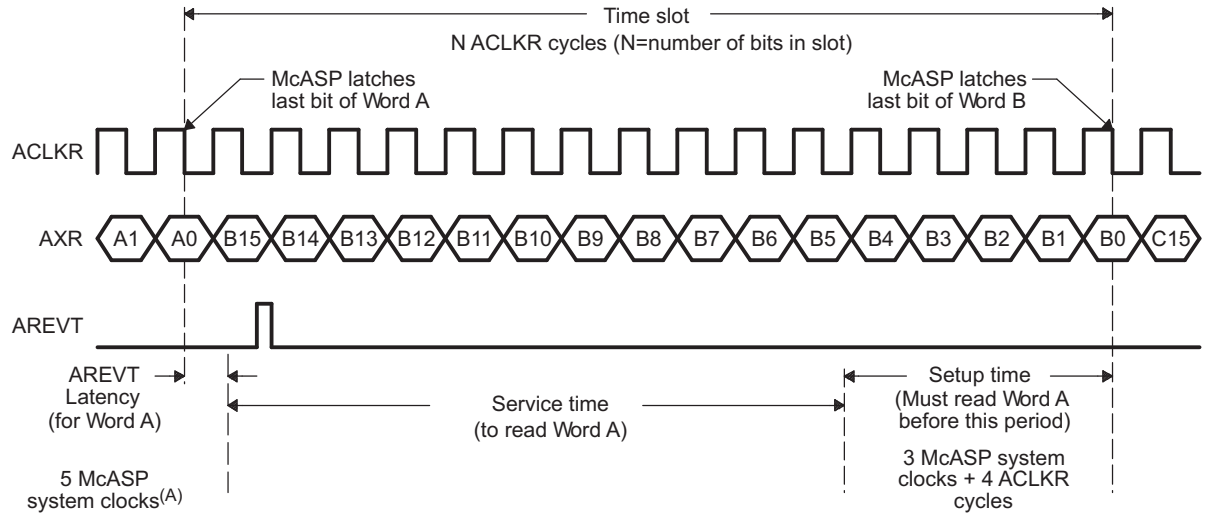
Figure 16-26 shows the timing details of when AREVT is generated at the McASP boundary. In this example, as soon as the last bit (bit A0) of Word A is received, the McASP sets the RDATA flag and generates an AREVT event. However, it takes up to 5 McASP system clocks (AREVT Latency) before AREVT is active at the McASP boundary. Upon AREVT, the processor can begin servicing the McASP by reading Word A from the RBUF (Processor Service Time). The processor must read Word A from the XBUF no later than the setup time required by the McASP (Setup Time).

The maximum Processor Service Time (Figure 16-26) can be calculated as:

Processor Service Time = Time Slot - AREVT Latency - Setup Time

The Processor Service Time calculation for receive is similar to the calculation for transmit. See Example 16-1 for Processor Service Time calculation using transmit as an example.

**Figure 16-26. Processor Service Time Upon Receive DMA Event (AREVT)**




A The device uses SYSCLK2 as the McASP system clock source.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.8.1.2 Transfers Through the Data Port (DAT)

**NOTE:** The data port (DAT) addresses are located in L3 and L4 tables in the device data sheet for the appropriate data peripheral registers. To perform internal transfers through the data port, clear XBUSEL/RBUSEL bit to 0 in the respective XFMT/RFMT registers. Failure to do so will result in software malfunction.

Typically, you will access the McASP XRBUF registers through the data port. To access through the data port, simply have the CPU or DMA access the XRBUF through its data port location. Refer to the device-specific data manual for the exact memory address. Through the data port, the DMA/CPU can service all the serializers through a single address. The McASP automatically cycles through the appropriate serializers.

For transmit operations through the data port, the DMA/CPU should write to the same XBUF data port address to service all of the active transmit serializers. In addition, the DMA/CPU should write to the XBUF for all active transmit serializers in incremental (although not necessarily consecutive) order. For example, if serializers 0, 4, and 5, are set up as active transmitters, the DMA/CPU should write to the XBUF data port address four times with data for serializers 0, 4, and 5, upon each transmit data ready event. This exact servicing order must be followed so that data appears in the appropriate serializers.

Similarly, for receive operations through the data port, the DMA/CPU should read from the same RBUF data port address to service all of the active receive serializers. In addition, reads from the active receive serializers through the data port return data in incremental (although not necessarily consecutive) order. For example, if serializers 1, 2, and 3, are set up as active receivers, the DMA/CPU should read from the RBUF data port address four times to obtain data for serializers 1, 2, and 3, in this exact order, upon each receive data ready event.

When transmitting, the DMA/CPU must write data to each serializer configured as "active" (active slot in XTDM register) and "transmit" (Tx enabled in SCRCTLn register) within each time slot. Failure to do so results in a buffer underrun condition (Section 16.2.8.4.2). Similarly, when receiving, data must be read from each serializer configured as "active" (active slot in RTDM register) and "receive" (Rx enabled in SCRCTLn register) within each time slot. Failure to do results in a buffer overrun condition (Section 16.2.8.4.3).

To perform internal transfers through the data port, clear XBUSEL/RBUSEL bit to 0 in the respective XFMT/RFMT registers.

#### 16.2.8.1.3 Transfers Through the Configuration Bus (CFG)

**NOTE:** To perform internal transfers through the configuration bus, set XBUSEL/RBUSEL bit to 1 in the respective XFMT/RFMT registers. Failure to do so will result in software malfunction.

In this method, the DMA/CPU accesses the XRBUF registers through the configuration bus address. The exact XRBUF register address for any particular serializer is determined by adding the offset for that particular serializer to the base address for the particular McASP (found in the device-specific data manual). XRBUF for the serializers configured as transmitters is given the name XBUF*n*. For example, the XRBUF associated with transmit serializer 2 is named XBUF2. Similarly, XRBUF for the serializers configured as receivers is given the name RBUF*n*.

Accessing the XRBUF registers through the data port is different because the CPU/DMA only needs to access one single address. When accessing through the configuration bus, the CPU/DMA must provide the exact XBUF*n* or RBUF*n* address for each access.

When transmitting, DMA/CPU must write data to each serializer configured as "active" and "transmit" within each time slot. Failure to do so results in a buffer underrun condition (Section 16.2.8.4.2). Similarly when receiving, data must be read from each serializer configured as "active" and "receive" within each time slot. Failure to do results in a buffer overrun condition (Section 16.2.8.4.3).

To perform internal transfers through the configuration bus, set XBUSEL/RBUSEL bit to 1 in the respective XFMT/RFMT registers.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.8.1.4 Using the CPU for McASP Servicing

The CPU can be used to service the McASP through interrupt (upon AXINT/ARINT interrupts) or through polling the XDATA bit in the XSTAT register and the RDATA bit in the RSTAT register. As discussed in Section 16.2.8.1.2 and Section 16.2.8.1.3, the CPU can access XRBUF serializer buffer through either the data port or through the configuration bus.

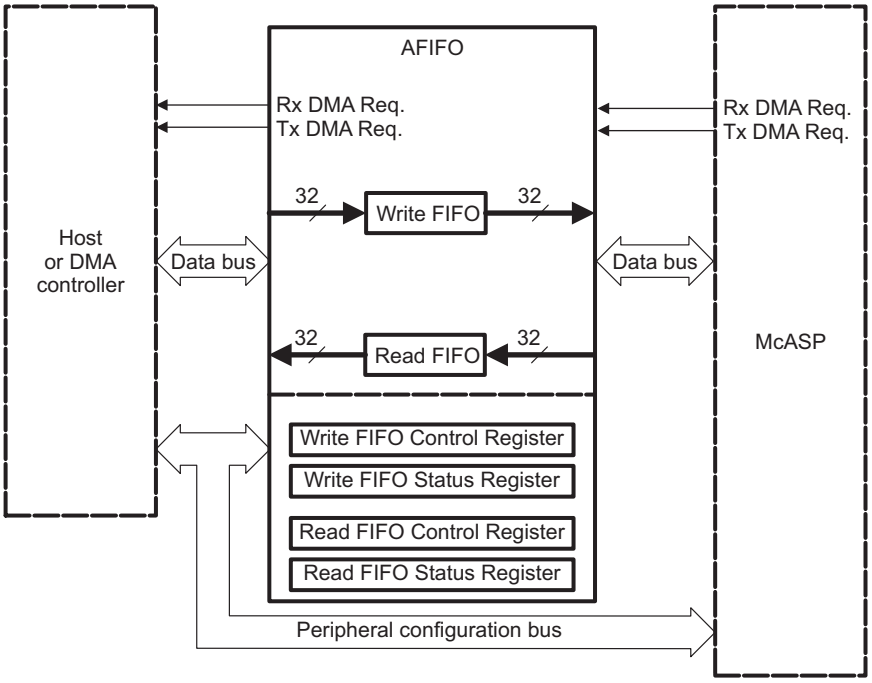
To use the CPU to service the McASP through interrupts, the XDATA/RDATA bit must be enabled in the respective XINTCTL/RINTCTL registers, to generate interrupts AXINT/ARINT to the CPU upon data ready.

### 16.2.8.2 McASP Audio FIFO (AFIFO)

The AFIFO contains two FIFOs: one Read FIFO (RFIFO), and one Write FIFO (WFIFO). To ensure backward compatibility with existing software, both the Read and Write FIFOs are disabled by default. See Figure 16-27 for a high-level block diagram of the AFIFO.

The AFIFO may be enabled/disabled and configured via the WFIFOCTL and RFIFOCTL registers. Note that if the Read or Write FIFO is to be enabled, it must be enabled prior to initializing the receive/transmit section of the McASP (see Section 16.2.10.2 for details).

**Figure 16-27. McASP Audio FIFO (AFIFO) Block Diagram**




#### 16.2.8.2.1 AFIFO Data Transmission

When the Write FIFO is disabled, transmit DMA requests pass through directly from the McASP to the host/DMA controller. Whether the WFIFO is enabled or disabled, the McASP generates transmit DMA requests as needed; the AFIFO is “invisible” to the McASP.

When the Write FIFO is enabled, transmit DMA requests from the McASP are sent to the AFIFO, which in turn generates transmit DMA requests to the host/DMA controller.

If the Write FIFO is enabled, upon a transmit DMA request from the McASP, the WFIFO writes *WNUMDMA* 32-bit words to the McASP if and when there are at least *WNUMDMA* words in the Write FIFO. If there are not, the WFIFO waits until this condition has been satisfied. At that point, it writes *WNUMDMA* words to the McASP. (See description for WFIFOCTL.WNUMDMA in Section 16.3.43.)


Copyright © 2011–2016, Texas Instruments Incorporated

If the host CPU writes to the Write FIFO, independent of a transmit DMA request, the WFIFO will accept host writes until full. After this point, excess data will be discarded.

Note that when the WFIFO is first enabled, it will immediately issue a transmit DMA request to the host. This is because it begins in an empty state, and is therefore ready to accept data.

#### 16.2.8.2.1.1 Transmit DMA Event Pacer

The AFIFO may be configured to delay making a transmit DMA request to the host until the Write FIFO has enough space for a specified number of words. In this situation, the number of transmit DMA requests to the host or DMA controller is reduced.

If the Write FIFO has space to accept *WNUMEVT* 32-bit words, it generates a transmit DMA request to the host and then waits for a response. Once *WNUMEVT* words have been written to the FIFO, it checks again to see if there is space for *WNUMEVT* 32-bit words. If there is space, it generates another transmit DMA request to the host, and so on. In this fashion, the Write FIFO will attempt to stay filled.

Note that if transmit DMA event pacing is desired, WFIFOCTL.WNUMEVT should be set to a non-zero integer multiple of the value in WFIFOCTL.WNUMDMA. If transmit DMA event pacing is not desired, then the value in WFIFOCTL.WNUMEVT should be set equal to the value in WFIFOCTL.WNUMDMA.

### 16.2.8.2.2 AFIFO Data Reception

When the Read FIFO is disabled, receive DMA requests pass through directly from McASP to the host/DMA controller. Whether the RFIFO is enabled or disabled, the McASP generates receive DMA requests as needed; the AFIFO is “invisible” to the McASP.

When the Read FIFO is enabled, receive DMA requests from the McASP are sent to the AFIFO, which in turn generates receive DMA requests to the host/DMA controller.

If the Read FIFO is enabled and the McASP makes a receive DMA request, the RFIFO reads *RNUMDMA* 32-bit words from the McASP, if and when the RFIFO has space for *RNUMDMA* words. If it does not, the RFIFO waits until this condition has been satisfied; at that point, it reads *RNUMDMA* words from the McASP. (See description for RFIFOCTL.RNUMDMA in Section 16.3.45.)

If the host CPU reads the Read FIFO, independent of a receive DMA request, and the RFIFO at that time contains less than *RNUMEVT* words, those words will be read correctly, emptying the FIFO.

#### 16.2.8.2.2.1 Receive DMA Event Pacer

The AFIFO may be configured to delay making a receive DMA request to the host until the Read FIFO contains a specified number of words. In this situation, the number of receive DMA requests to the host or DMA controller is reduced.

If the Read FIFO contains at least *RNUMEVT* 32-bit words, it generates a receive DMA request to the host and then waits for a response. Once *RNUMEVT* 32-bit words have been read from the RFIFO, the RFIFO checks again to see if it contains at least another *RNUMEVT* words. If it does, it generates another receive DMA request to the host, and so on. In this fashion, the Read FIFO will attempt to stay empty.

Note that if receive DMA event pacing is desired, RFIFOCTL.RNUMEVT should be set to a non-zero integer multiple of the value in RFIFOCTL.RNUMDMA. If receive DMA event pacing is not desired, then the value in RFIFOCTL.RNUMEVT should be set equal to the value in RFIFOCTL.RNUMDMA.

### 16.2.8.2.3 Arbitration Between Transmit and Receive DMA Requests

If both the WFIFO and the RFIFO are enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the transmit DMA request. Once a transfer is in progress, it is allowed to complete.

If only the WFIFO is enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the transmit DMA request. Once a transfer is in progress, it is allowed to complete.

If only the RFIFO is enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the receive DMA request. Once a transfer is in progress, it is allowed to complete.


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.2.8.3 Formatter

#### *16.2.8.3.1 Transmit Bit Stream Data Alignment*

The McASP transmitter supports serial formats of:

- Slot or time slot size = 8, 12, 16, 20, 24, 28, 32 bits.
- Word size ≤ Slot size.
- Alignment when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad.
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot.
- Order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB.
  - LSB: least-significant bit of word is shifted out last, last bit is MSB.

Hardware support for these serial formats comes from the programmable options in the transmit bitstream format register (XFMT):

- XRVRS: bit reverse (1) or no bit reverse (0).
- XROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits.
- XSSZ: transmit slot size of 8, 12, 16, 20, 24, 28, or 32 bits.

XSSZ should always be programmed to match the slot size of the serial stream. The word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the XROT field.

Table 16-5 and Figure 16-28 show the XRVRS and XROT fields for each serial format and for both integer and Q31 fractional internal representations.

This discussion assumes that all slot size (SLOT in Table 16-5) and word size (WORD in Table 16-5) options are multiples of 4, since the transmit rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be transmitted in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1.

The transmit bit mask/pad unit operates on data as an initial step of the transmit format unit (see Figure 16-23), and the data is aligned in the same representation as it is written to the transmitter by the processor (typically Q31 or integer).

**Table 16-5. Transmit Bitstream Data Alignment**

| Figure 16-28 | Bit Stream Order | Bit Stream Alignment | Internal Numeric Representation | XFMT Bit | |
|---|---|---|---|---|---|
| | | | | XROT[1] | XRVRS |
| (a)[2] | MSB first | Left aligned | Q31 fraction | 0 | 1 |
| (b) | MSB first | Right aligned | Q31 fraction | SLOT - WORD | 1 |
| (c) | LSB first | Left aligned | Q31 fraction | 32 - WORD | 0 |
| (d) | LSB first | Right aligned | Q31 fraction | 32 - SLOT | 0 |
| (e)[2] | MSB first | Left aligned | Integer | WORD | 1 |
| (f) | MSB first | Right aligned | Integer | SLOT | 1 |
| (g) | LSB first | Left aligned | Integer | 0 | 0 |
| (h) | LSB first | Right aligned | Integer | (32 - (SLOT - WORD)) % 32 | 0 |

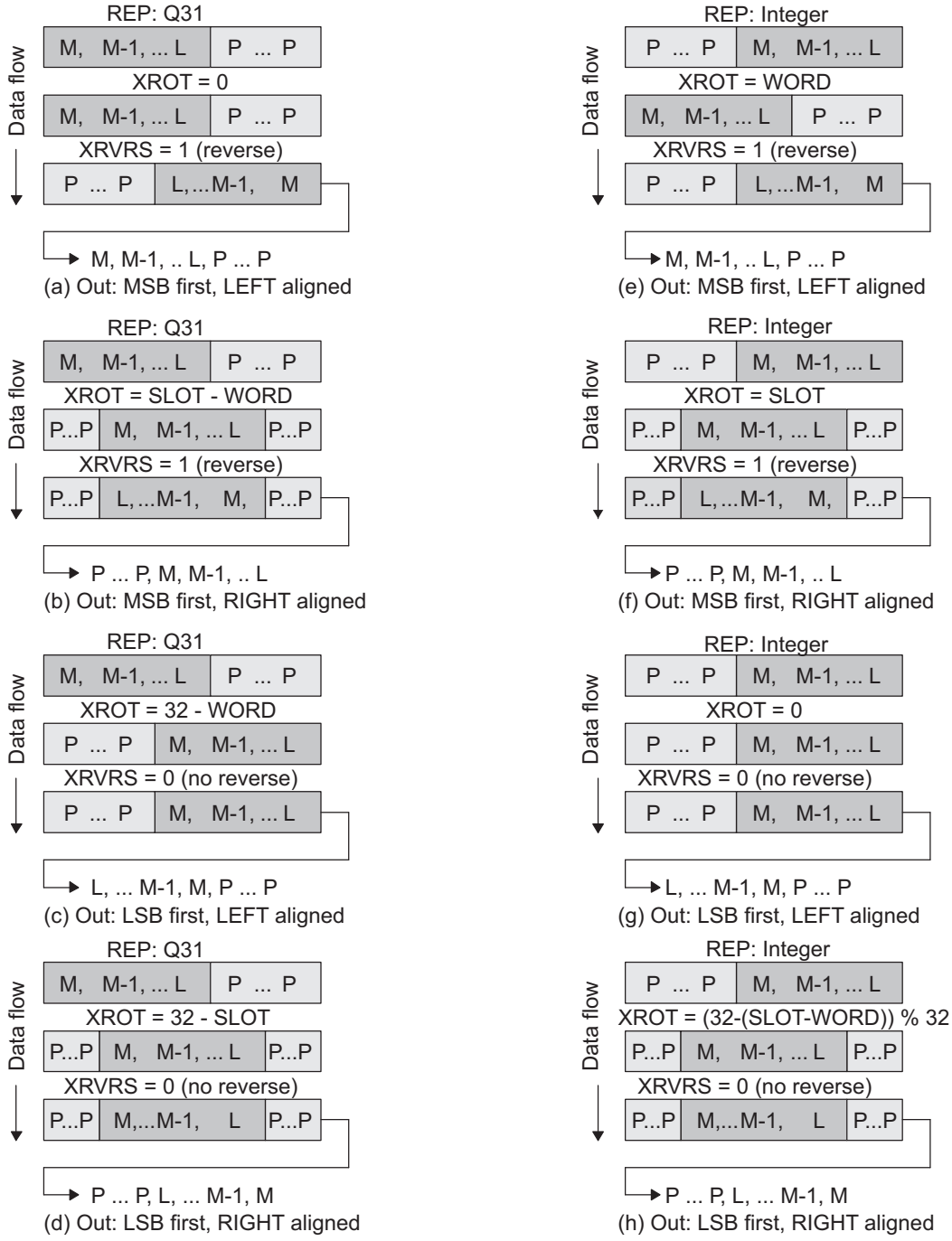
(1) WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator

(2) To transmit in I2S format, use MSB first, left aligned, and also select XDATDLY = 01 (1 bit delay)



Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 16-28. Data Flow Through Transmit Format Unit, Illustrated**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.8.3.2 Receive Bit Stream Data Alignment

The McASP receiver supports serial formats of:

- Slot or time slot size = 8, 12, 16, 20, 24, 28, 32 bits.
- Word size ≤ Slot size.
- Alignment when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad.
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot.
- Order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB.
  - LSB: least-significant bit of word is shifted out last, last bit is MSB.

Hardware support for these serial formats comes from the programmable options in the receive bitstream format register (RFMT):

- RRVRS: bit reverse (1) or no bit reverse (0).
- RROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits.
- RSSZ: receive slot size of 8, 12, 16, 20, 24, 28, or 32 bits.

RSSZ should always be programmed to match the slot size of the serial stream. The word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the RROT field.

Table 16-6 and Figure 16-29 show the RRVRS and RROT fields for each serial format and for both integer and Q31 fractional internal representations.

This discussion assumes that all slot size and word size options are multiples of 4; since the receive rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be transmitted in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1.

The receive bit mask/pad unit operates on data as the final step of the receive format unit (see Figure 16-22), and the data is aligned in the same representation as it is read from the receiver by the processor (typically Q31 or integer).

**Table 16-6. Receive Bitstream Data Alignment**

| Figure 16-29 | Bit Stream Order | Bit Stream Alignment | Internal Numeric Representation | RFMT Bit | |
|---|---|---|---|---|---|
| | | | | RROT[1] | RRVRS |
| (a)[2] | MSB first | Left aligned | Q31 fraction | SLOT | 1 |
| (b) | MSB first | Right aligned | Q31 fraction | WORD | 1 |
| (c) | LSB first | Left aligned | Q31 fraction | (32 - (SLOT - WORD)) % 32 | 0 |
| (d) | LSB first | Right aligned | Q31 fraction | 0 | 0 |
| (e)[2] | MSB first | Left aligned | Integer | SLOT - WORD | 1 |
| (f) | MSB first | Right aligned | Integer | 0 | 1 |
| (g) | LSB first | Left aligned | Integer | 32 - SLOT | 0 |
| (h) | LSB first | Right aligned | Integer | 32 - WORD | 0 |

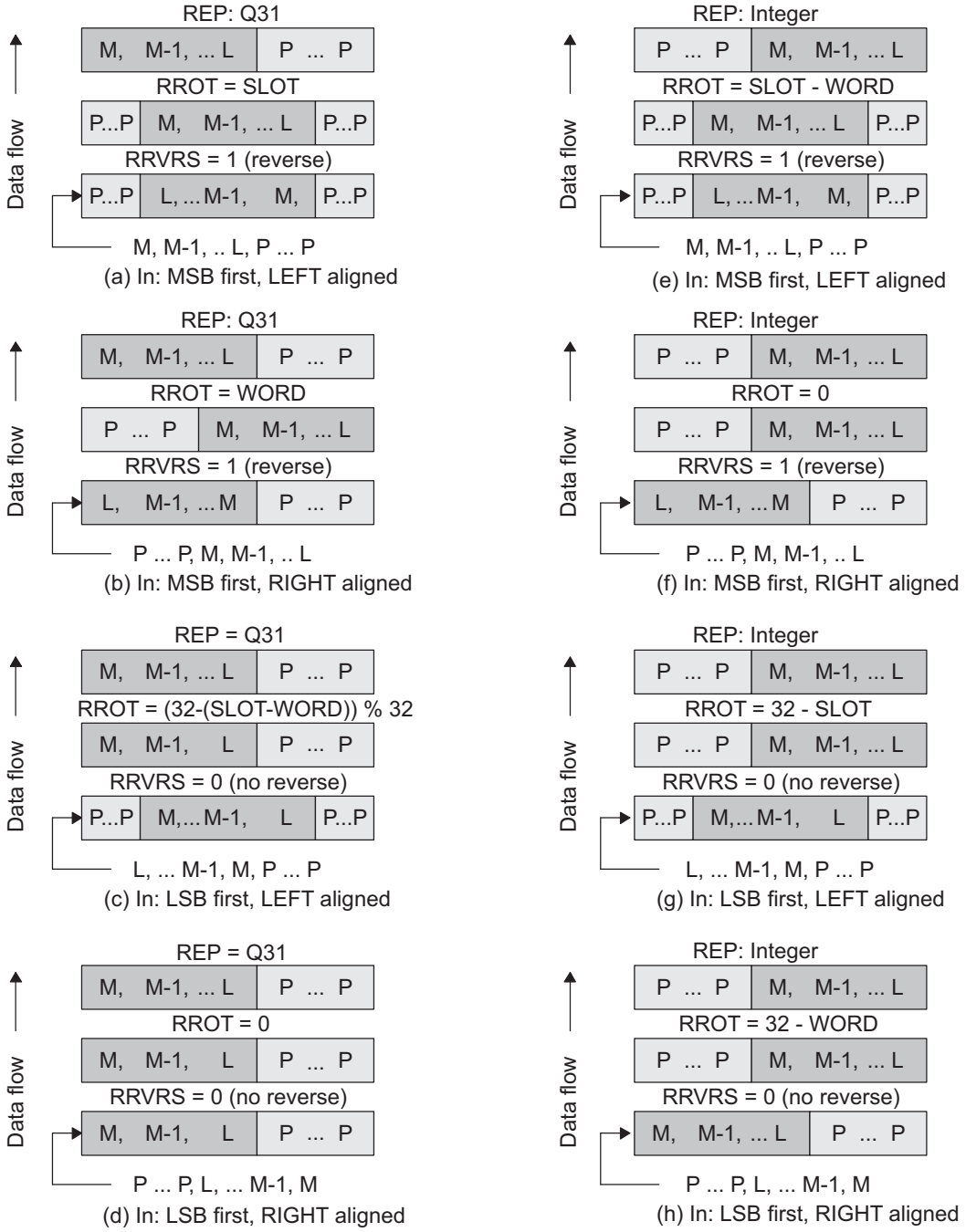
(1) WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator

(2) To transmit in I2S format, select MSB first, left aligned, and also select RDATDLY = 01 (1 bit delay)



Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 16-29. Data Flow Through Receive Format Unit, Illustrated**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.8.4 Error Handling and Management

To support the design of a robust audio system, the McASP includes error-checking capability for the serial protocol, data underrun, and data overrun. In addition, the McASP includes a timer that continually measures the high-frequency master clock every 32 AHCLKX/AHCLKR clock cycles. The timer value can be read to get a measurement of the clock frequency and has a minimum and maximum range setting that can set an error flag if the master clock goes out of a specified range.

Upon the detection of any one or more errors (software selectable), or the assertion of the AMUTEIN input pin, the AMUTE output pin may be asserted to a high or low level to immediately mute the audio output. In addition, an interrupt may be generated if desired, based on any one or more of the error sources.

##### *16.2.8.4.1 Unexpected Frame Sync Error*

An unexpected frame sync occurs when:

- In burst mode, when the next active edge of the frame sync occurs early such that the current slot will not be completed by the time the next slot is scheduled to begin.
- In TDM mode, a further constraint is that the frame sync must occur exactly during the correct bit clock (not a cycle earlier or later) and only before slot 0. An unexpected frame sync occurs if this condition is not met.

When an unexpected frame sync occurs, there are two possible actions depending upon when the unexpected frame sync occurs:

1. Early: An early unexpected frame sync occurs when the McASP is in the process of completing the current frame and a new frame sync is detected (not including overlap that occurs due to a 1 or 2 bit frame sync delay). When an early unexpected frame sync occurs:
   - Error interrupt flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Current frame is not resynchronized. The number of bits in the current frame is completed. The next frame sync, which occurs after the current frame is completed, will be resynchronized.
2. Late: A late unexpected frame sync occurs when there is a gap or delay between the last bit of the previous frame and the first bit of the next frame. When a late unexpected frame sync occurs (as soon as the gap is detected):
   - Error interrupt flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Resynchronization occurs upon the arrival of the next frame sync.

Late frame sync is detected the same way in both burst mode and TDM mode; however, in burst mode, late frame sync is not meaningful and its interrupt enable should not be set.

##### *16.2.8.4.2 Buffer Underrun Error - Transmitter*

A buffer underrun can only occur for serializers programmed to be transmitters. A buffer underrun occurs when the serializer is instructed by the transmit state machine to transfer data from XRBUF[n] to XRSR[n], but XRBUF[n] has not yet been written with new data since the last time the transfer occurred. When this occurs, the transmit state machine sets the XUNDRN flag.

An underrun is checked only once per time slot. The XUNDRN flag is set when an underrun condition occurs. Once set, the XUNDRN flag remains set until the processor explicitly writes a 1 to the XUNDRN bit to clear the XUNDRN bit.

In DIT mode, a pair of BMC zeros is shifted out when an underrun occurs (four bit times at 128 × fs). By shifting out a pair of zeros, a clock may be recovered on the receiver. To recover, reset the McASP and start again with the proper initialization.

In TDM mode, during an underrun case, a long stream of zeros are shifted out causing the DACs to mute. To recover, reset the McASP and start again with the proper initialization.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.8.4.3 Buffer Overrun Error - Receiver

A buffer overrun can only occur for serializers programmed to be receivers. A buffer overrun occurs when the serializer is instructed to transfer data from XRSR[n] to XRBUF[n], but XRBUF[n] has not yet been read by either the DMA or the processor. When this occurs, the receiver state machine sets the ROVRN flag. However, the individual serializer writes over the data in the XRBUF[n] register (destroying the previous sample) and continues shifting.

An overrun is checked only once per time slot. The ROVRN flag is set when an overrun condition occurs. It is possible that an overrun occurs on one time slot but then the processor catches up and does not cause an overrun on the following time slots. However, once the ROVRN flag is set, it remains set until the processor explicitly writes a 1 to the ROVRN bit to clear the ROVRN bit.

#### 16.2.8.4.4 DMA Error - Transmitter

A transmit DMA error, as indicated by the XDMAERR flag in the XSTAT register, occurs when the DMA (or CPU) writes more words to the DAT port of the McASP than it should. For each DMA event, the DMA should write exactly as many words as there are serializers enabled as transmitters.

XDMAERR indicates that the DMA (or CPU) wrote too many words to the McASP for a given transmit DMA event. Writing too few words results in a transmit underrun error setting XUNDRN in XSTAT.

While XDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or CPU. You should reinitialize both the McASP transmitter and the DMA to resynchronize them.

#### 16.2.8.4.5 DMA Error - Receiver

A receive DMA error, as indicated by the RDMAERR flag in the RSTAT register, occurs when the DMA (or CPU) reads more words from the DAT port of the McASP than it should. For each DMA event, the DMA should read exactly as many words as there are serializers enabled as receivers.

RDMAERR indicates that the DMA (or CPU) read too many words from the McASP for a given receive DMA event. Reading too few words results in a receiver overrun error setting ROVRN in RSTAT.

While RDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or CPU. You should reinitialize both the McASP receiver and the DMA to resynchronize them.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.8.4.6 Clock Failure Detection

##### 16.2.8.4.6.1 Clock-Failure Check Startup

It is expected, initially, that the clock-failure circuits will generate an error until at least one measurement has been taken. Therefore, the clock failure interrupts, clock switch, and mute functions should not immediately be enabled, but be enabled only after a specific startup procedure. The startup procedure is:

1. For the transmit clock failure check:
   (a) Configure transmit clock failure detect logic (XMIN, XMAX, XPS) in the transmit clock check control register (XCLKCHK).
   (b) Clear transmit clock failure flag (XCKFAIL) in the transmit status register (XSTAT).
   (c) Wait until first measurement is taken (> 32 AHCLKX clock periods).
   (d) Verify no clock failure is detected.
   (e) Repeat steps b–d until clock is running and is no longer issuing clock failure errors.
   (f) After the transmit clock is measured and falls within the acceptable range, the following may be enabled:
      (i) transmit clock failure interrupt enable bit (XCKFAIL) in the transmitter interrupt control register (XINTCTL).
      (ii) mute option (XCKFAIL) in the mute control register (AMUTE).
2. For the receive clock failure check:
   (a) Configure receive clock failure detect logic (RMIN, RMAX, RPS) in the receive clock check control register (RCLKCHK).
   (b) Clear receive clock failure flag (RCKFAIL) in the receive status register (RSTAT).
   (c) Wait until first measurement is taken (> 32 AHCLKR clock periods).
   (d) Verify no clock failure is detected.
   (e) Repeat steps b–d until clock is running and is no longer issuing clock failure errors.
   (f) After the receive clock is measured and falls within the acceptable range, the following may be enabled:
      (i) receive clock failure interrupt enable bit (RCKFAIL) in the receiver interrupt control register (RINTCTL).
      (ii) mute option (RCKFAIL) in the mute control register (AMUTE).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.8.4.6.2 Transmit Clock Failure Check and Recovery

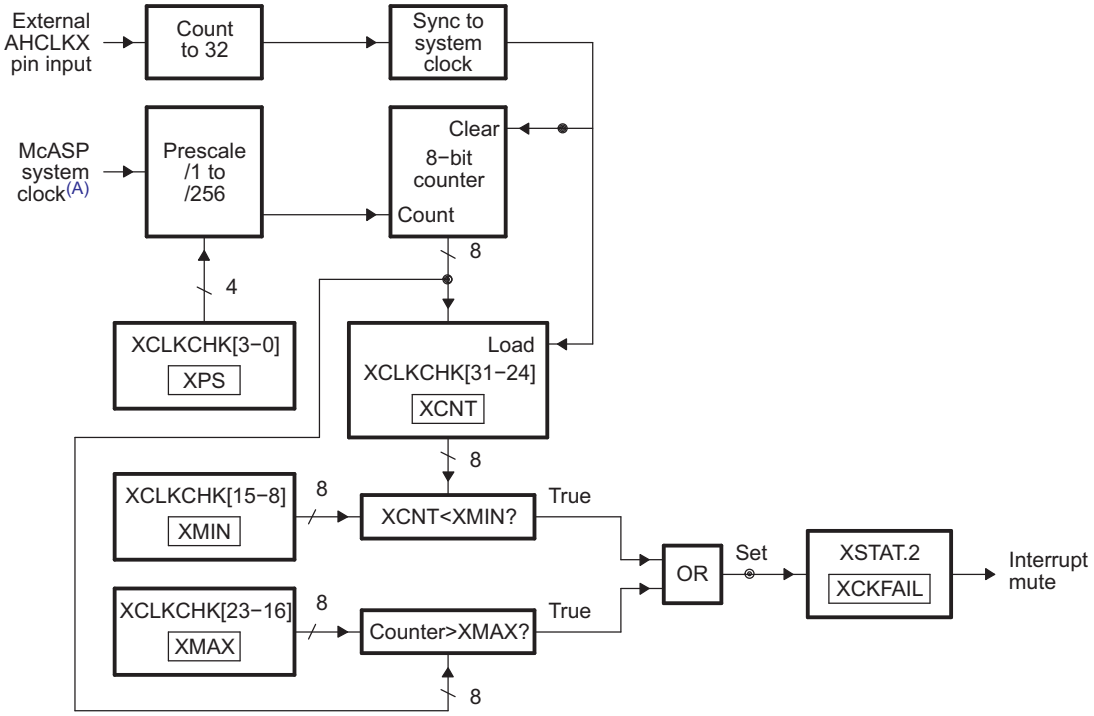
The transmit clock failure check circuit (Figure 16-30) works off both the internal McASP system clock and the external high-frequency serial clock (AHCLKX). It continually counts the number of system clocks for every 32 high rate serial clock (AHCLKX) periods, and stores the count in XCNT of the transmit clock check control register (XCLKCHK) every 32 high rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (XMIN), and automatically flags an interrupt (XCKFAIL in XSTST) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is smaller than XMIN. The logic continually compares the current count (from the running system clock counter) against the maximum allowable boundary (XMAX). This is in case the external clock completely stops, so that the counter value is not copied to XCNT. An out-of-range maximum condition occurs when the count is greater than XMAX. Note that the XMIN and XMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate either that an unstable clock was detected, or that the audio source has changed and a new sample rate is being used.

In order for the transmit clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset regardless if AHCLKX is internally generated or externally sourced.

**Figure 16-30. Transmit Clock Failure Detection Circuit Block Diagram**




A Refer to the device data manual for the McASP system clock source. This is not the same as AUXCLK.

If a clock failure is detected, the transmit clock failure flag (XCKFAIL) in XSTAT is set. This causes an interrupt if the transmit clock failure interrupt enable bit (XCKFAIL) in XINTCTL is set.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.8.4.6.3 Receive Clock Failure Check and Recovery

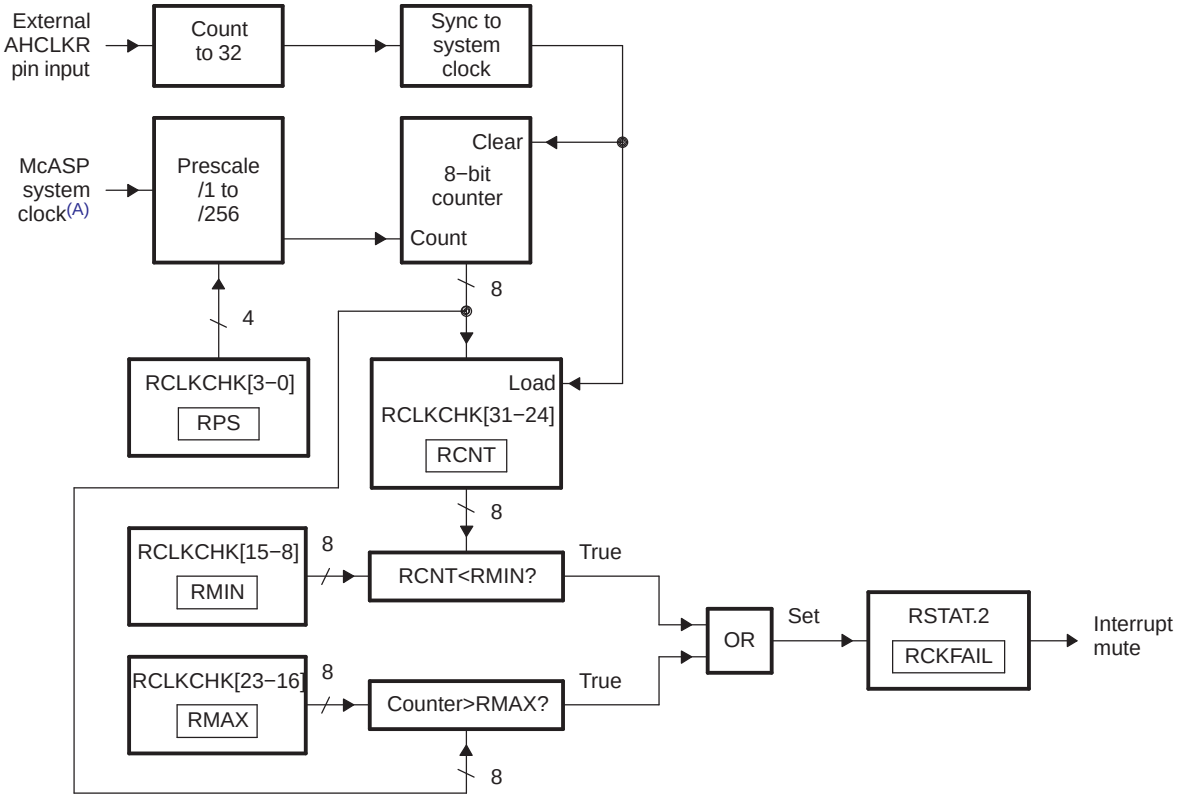
The receive clock failure check circuit (Figure 16-31) works off both the internal McASP system clock and the external high-frequency serial clock (AHCLKR). It continually counts the number of system clocks for every 32 high rate serial clock (AHCLKR) periods, and stores the count in RCNT of the receive clock check control register (RCLKCHK) every 32 high rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (RMIN) and automatically flags an interrupt (RCKFAIL in RSTAT) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is smaller than RMIN. The logic continually compares the current count (from the running system clock counter) against the maximum allowable boundary (RMAX). This is in case the external clock completely stops, so that the counter value is not copied to RCNT. An out-of-range maximum condition occurs when the count is greater than RMAX. Note that the RMIN and RMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate either that an unstable clock was detected or that the audio source has changed and a new sample rate is being used.

In order for the receive clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset regardless if AHCLKR is internally generated or externally sourced.

**Figure 16-31. Receive Clock Failure Detection Circuit Block Diagram**




A Refer to device data manual for the McASP system clock source. This is not the same as AUXCLK.


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.2.8.5 Loopback Modes

The McASP features a digital loopback mode (DLB) that allows testing of the McASP code in TDM mode with a single processor device. In loopback mode, output of the transmit serializers is connected internally to the input of the receive serializers. Therefore, you can check the receive data against the transmit data to ensure that the McASP settings are correct. Digital loopback mode applies to TDM mode only (2 to 32 slots in a frame). It does not apply to DIT mode (XMOD = 180h) or burst mode (XMOD = 0).

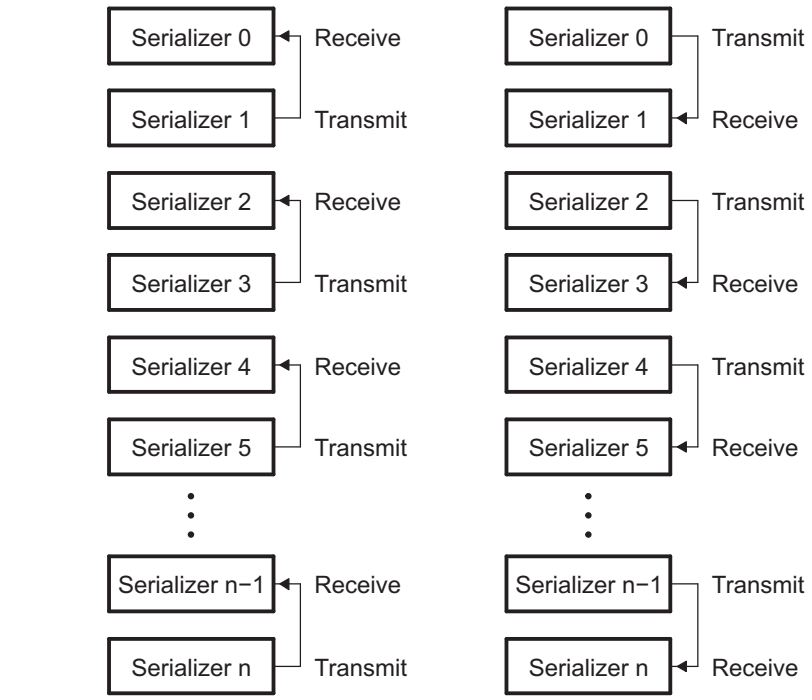
Figure 16-32 shows the basic logical connection of the serializers in loopback mode. Two types of loopback connections are possible, selected by the ORD bit in the digital loopback control register (DLBCTL) as follows:

- ORD = 0: Outputs of odd serializers are connected to inputs of even serializers. If this mode is selected, you should configure odd serializers to be transmitters and even serializers to be receivers.
- ORD = 1: Outputs of even serializers are connected to inputs of odd serializers. If this mode is selected, you should configure even serializers to be transmitters and odd serializers to be receivers.

Data can be externally visible at the I/O pin of the transmit serializer if the pin is configured as a McASP output pin by setting the corresponding PFUNC bit to 0 and PDIR bit to 1.

In loopback mode, the transmit clock and frame sync are used by both the transmit and receive sections of the McASP. The transmit and receive sections operate synchronously. This is achieved by setting the MODE bit of the DLBCTL register to 01b and the ASYNC bit of the ACLKXCTL register to 0.

**Figure 16-32. Serializers in Loopback Mode**




(a) DLBEN = 1 (loopback enabled) and ORD = 0 (even receive, odd transmit)

(b) DLBEN = 1 (loopback enabled) and ORD = 1 (odd receive, even transmit)


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.8.5.1 Loopback Mode Configurations

This is a summary of the settings required for digital loopback mode for TDM format:

- The DLBEN bit in DLBCTL must be set to 1 to enable loopback mode.
- The MODE bits in DLBCTL must be set to 01b for both the transmit and receive sections to use the transmit clock and frame sync generator.
- The ORD bit in DLBCTL must be programmed appropriately to select odd or even serializers to be transmitters or receivers. The corresponding serializers must be configured accordingly.
- The ASYNC bit in ACLKXCTL must be cleared to 0 to ensure synchronous transmit and receive operations.
- RMOD field in AFSRCTL and XMOD field in AFSXCTL must be set to 2h to 20h to indicate TDM mode. Loopback mode does not apply to DIT or burst mode.

### 16.2.9 Reset Considerations

The McASP has two reset sources: software reset and hardware reset.

#### 16.2.9.1 Software Reset Considerations

The transmitter and receiver portions of the McASP may be put in reset through the global control register (GBLCTL). Note that a valid serial clock must be supplied to the desired portion of the McASP (transmit and/or receive) in order to assert the software reset bits in GBLCTL. see Section 16.2.10.2for details on how to ensure reset has occurred.

The entire McASP module may also be reset through the Power and Sleep Controller (PSC). Note that from the McASP perspective, this reset appears as a hardware reset to the entire module.

#### 16.2.9.2 Hardware Reset Considerations

When the McASP is reset due to device reset, the entire serial port (including the transmitter and receiver state machines, and other registers) is reset.

### 16.2.10 Setup and Initialization

This section discusses steps necessary to use the McASP module.

#### 16.2.10.1 Considerations When Using a McASP

The following is a list of things to be considered for systems using a McASP:

#### 16.2.10.1.1 Clocks

For each receive and transmit section:

- External or internal generated bit clock and high frequency clock?
- If internally generated, what is the bit clock speed and the high frequency clock speed?
- Clock polarity?
- External or internal generated frame sync?
- If internally generated, what is frame sync speed?
- Frame sync polarity?
- Frame sync width?
- Transmit and receive sync or asynchronous?

#### 16.2.10.1.2 Data Pins

For each pin of each McASP:

- McASP or GPIO?
- Input or output?


Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.10.1.3 Data Format

For each transmit and receive data:

- Internal numeric representation (integer, Q31 fraction)?
- I2S or DIT (transmit only)?
- Time slot delay (0, 1, or 2 bit)?
- Alignment (left or right)?
- Order (MSB first, LSB first)?
- Pad (if yes, pad with what value)?
- Slot size?
- Rotate?
- Mask?

#### 16.2.10.1.4 Data Transfers

- Internal: DMA or CPU?
- External: TDM or burst?
- Bus: configuration bus (CFG) or data port (DAT)?

### 16.2.10.2 Transmit/Receive Section Initialization

You must follow the following steps to properly configure the McASP. If external clocks are used, they should be present prior to the following initialization steps.

1. Reset McASP to default values by setting GBLCTL = 0. Poll the bits to ensure that the active reset value (0) is successfully latched into the GBLCTL register.
2. Configure all McASP registers except GBLCTL in the following order:
   (a) Receive registers: RMASK, RFMT, AFSRCTL, ACLKRCTL, AHCLKRCTL, RTDM, RINTCTL, RCLKCHK. If external clocks AHCLKR and/or ACLKR are used, they must be running already for proper synchronization of the GBLCTL register.
   (b) Transmit registers: XMASK, XFMT, AFSXCTL, ACLKXCTL, AHCLKXCTL, XTDM, XINTCTL, XCLKCHK. If external clocks AHCLKX and/or ACLKX are used, they must be running already for proper synchronization of the GBLCTL register.
   (c) Serializer registers: SRCTL[n].
   (d) Global registers: Registers PFUNC, PDIR, DITCTL, DLBCTL, AMUTE. Note that PDIR should only be programmed after the clocks and frames are set up in the steps above. This is because the moment a clock pin is configured as an output in PDIR, the clock pin starts toggling at the rate defined in the corresponding clock control register. Therefore you must ensure that the clock control register is configured appropriately before you set the pin to be an output. A similar argument applies to the frame sync pins. Also note that the reset state for the transmit high-frequency clock divide register (HCLKXDIV) is divide-by-1, and the divide-by-1 clocks are not gated by the transmit high-frequency clock divider reset enable (XHCLKRST).
   (e) DIT registers: For DIT mode operation, set up registers DITCSRA[n], DITCSRB[n], DITUDRA[n], and DITUDRB[n].
3. Start the respective high-frequency serial clocks AHCLKX and/or AHCLKR. This step is necessary even if external high-frequency serial clocks are used:
   (a) Take the respective internal high-frequency serial clock divider(s) out of reset by setting the RHCLKRST bit for the receiver and/or the XHCLKRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be held at 0.
   (b) Software must read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.


Copyright © 2011–2016, Texas Instruments Incorporated

4. Start the respective serial clocks ACLKX and/or ACLKR. This step can be skipped if external serial clocks are used and they are running:
   (a) Take the respective internal serial clock divider(s) out of reset by setting the RCLKRST bit for the receiver and/or the XCLKRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   (b) Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
5. Setup data acquisition as required:
   (a) If DMA is used to service the McASP, set up data acquisition as desired and start the DMA in this step, before the McASP is taken out of reset.
   (b) If CPU interrupt is used to service the McASP, enable the transmit and/ or receive interrupt as required.
   (c) If CPU polling is used to service the McASP, no action is required in this step.
6. Activate serializers.
   (a) Before starting, clear the respective transmitter and receiver status registers by writing XSTAT = FFFFh and RSTAT = FFFFh.
   (b) Take the respective serializers out of reset by setting the RSRCLR bit for the receiver and/or the XSRCLR bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   (c) Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
7. Verify that all transmit buffers are serviced. Skip this step if the transmitter is not used. Also, skip this step if time slot 0 is selected as inactive (special cases, see Figure 16-20, second waveform). As soon as the transmit serializer is taken out of reset, XDATA in the XSTAT register is set, indicating that XBUF is empty and ready to be serviced. The XDATA status causes an DMA event AXEVT to be generated, and can cause an interrupt AXINT to be generated if it is enabled in the XINTCTL register.
   (a) If DMA is used to service the McASP, the DMA automatically services the McASP upon receiving AXEVT. Before proceeding in this step, you should verify that the XDATA bit in the XSTAT is cleared to 0, indicating that all transmit buffers are already serviced by the DMA.
   (b) If CPU interrupt is used to service the McASP, interrupt service routine is entered upon the AXINT interrupt. The interrupt service routine should service the XBUF registers. Before proceeding in this step, you should verify that the XDATA bit in XSTAT is cleared to 0, indicating that all transmit buffers are already serviced by the CPU.
   (c) If CPU polling is used to service the McASP, the XBUF registers should be written to in this step.
8. Release state machines from reset.
   (a) Take the respective state machine(s) out of reset by setting the RSMRST bit for the receiver and/or the XSMRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   (b) Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
9. Release frame sync generators from reset. Note that it is necessary to release the internal frame sync generators from reset, even if an external frame sync is being used, because the frame sync error detection logic is built into the frame sync generator.
   (a) Take the respective frame sync generator(s) out of reset by setting the RFRST bit for the receiver, and/or the XFRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   (b) Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.


Copyright © 2011–2016, Texas Instruments Incorporated

10. Upon the first frame sync signal, McASP transfers begin. The McASP synchronizes to an edge on the frame sync pin, not the level on the frame sync pin. This makes it easy to release the state machine and frame sync generators from reset.

    (a) For example, if you configure the McASP for a rising edge transmit frame sync, then you do not need to wait for a low level on the frame sync pin before releasing the McASP transmitter state machine and frame sync generators from reset.

#### 16.2.10.3 Separate Transmit and Receive Initialization

In many cases, it is desirable to separately initialize the McASP transmitter and receiver. For example, you may delay the initialization of the transmitter until the type of data coming in on the receiver is recognized. Or a change in the incoming data stream on the receiver may necessitate a reinitialization of the transmitter.

In this case, you may still follow the sequence outlined in Section 16.2.10.2, but use it for each section (transmit, receive) individually. The GBLCTL register is aliased to RGBLCTL and XGBLCTL to facilitate separate initialization of transmit and receive sections.

#### 16.2.10.4 Importance of Reading Back GBLCTL

In Section 16.2.10.2, steps 3b, 4b, 6c, 8b, and 9b state that GBLCTL should be read back until the bits that were written are successfully latched. This is important, because the transmitter and receiver state machines run off of the respective bit clocks, which are typically about tens to hundreds of times slower than the processor's internal bus clock. Therefore, it takes many cycles between when the processor writes to GBLCTL (or RGBLCTL and XGBLCTL), and when the McASP actually recognizes the write operation. If you skip this step, then the McASP may never see the reset bits in the global control registers get asserted and de-asserted; resulting in an uninitialized McASP.

Therefore, the logic in McASP has been implemented such that once the processor writes GBLCTL, RGBLCTL, or XGBLCTL, the resulting write is not visible by reading back GBLCTL until the McASP has recognized the change. This typically requires two bit clocks plus two processor bus clocks to occur.

Also, if the bit clocks can be completely stopped, any software that polls GBLCTL should be implemented with a time-out. If GBLCTL does not have a time-out, and the bit clock stops, the changes written to GBLCTL will not be reflected until the bit clock restarts.

Finally, please note that while RGBLCTL and XGBLCTL allow separate changing of the receive and transmit halves of GBLCTL, they also immediately reflect the updated value (useful for debug purposes). Only GBLCTL can be used for the read back step.

#### 16.2.10.5 Synchronous Transmit and Receive Operation (ASYNC = 0)

When ASYNC = 0 in ACLKXCTL, the transmit and receive sections operate synchronously from the transmit section clock and transmit frame sync signals (Figure 16-16). The receive section may have a different (but compatible in terms of slot size) data format.

When ASYNC = 0, the receive frame sync generator is internally disabled. If the AFSX pin is configured as an output, it serves as the frame sync signal for both transmit and receive. The AFSR pin should not be used because the transmit frame sync generator output, which is used by both the transmitter and the receiver when ASYNC = 0, is not propagated to the AFSR pin (Figure 16-18).

When ASYNC = 0, the transmit and receive sections must share some common settings, since they both use the same clock and frame sync signals:

- DITEN = 0 in DITCTL (TDM mode is enabled).
- The total number of bits per frame must be the same (that is, RSSZ × RMOD must equal to XSSZ × XMOD).
- Both transmit and receive should either be specified as burst or TDM mode, but not mixed.
- The settings in ACLKRCTL are irrelevant.
- FSXM must match FSRM.
- FXWID must match FRWID.


Copyright © 2011–2016, Texas Instruments Incorporated

For all other settings, the transmit and receive sections may be programmed independently.

#### 16.2.10.6 Asynchronous Transmit and Receive Operation (ASYNC = 1)

When ASYNC = 1 in ACLKXCTL, the transmit and receive sections operate completely independently and have separate clock and frame sync signals (Figure 16-16, Figure 16-17, and Figure 16-18). The events generated by each section come asynchronously.

### *16.2.11 Interrupts*

#### 16.2.11.1 Transmit Data Ready Interrupt

The transmit data ready interrupt (XDATA) is generated if XDATA is 1 in the XSTAT register and XDATA is also enabled in XINTCTL. Section 16.2.8.1.1 provides details on when XDATA is set in the XSTAT register.

A transmit start of frame interrupt (XSTAFRM) is triggered by the recognition of transmit frame sync. A transmit last slot interrupt (XLAST) is a qualified version of the data ready interrupt (XDATA). It has the same behavior as the data ready interrupt, but is further qualified by having the data requested belonging to the last slot (the slot that just ended was next-to-last TDM slot, current slot is last slot).

#### 16.2.11.2 Receive Data Ready Interrupt

The receive data ready interrupt (RDATA) is generated if RDATA is 1 in the RSTAT register and RDATA is also enabled in RINTCTL. Section 16.2.8.1.2 provides details on when RDATA is set in the RSTAT register.

A receiver start of frame interrupt (RSTAFRM) is triggered by the recognition of a receiver frame sync. A receiver last slot interrupt (RLAST) is a qualified version of the data ready interrupt (RDATA). It has the same behavior as the data ready interrupt, but is further qualified by having the data in the buffer come from the last TDM time slot (the slot that just ended was last TDM slot).

#### 16.2.11.3 Error Interrupts

Upon detection, the following error conditions generate interrupt flags:

- In the receive status register (RSTAT):
  - Receiver overrun (ROVRN).
  - Unexpected receive frame sync (RSYNCERR).
  - Receive clock failure (RCKFAIL).
  - Receive DMA error (RDMAERR).
- In the transmit status register (XSTAT):
  - Transmit underrun (XUNDRN).
  - Unexpected transmit frame sync (XSYNCERR).
  - Transmit clock failure (XCKFAIL).
  - Transmit DMA error (XDMAERR).

Each interrupt source also has a corresponding enable bit in the receive interrupt control register (RINTCTL) and transmit interrupt control register (XINTCTL). If the enable bit is set in RINTCTL or XINTCTL, an interrupt is requested when the interrupt flag is set in RSTAT or XSTAT. If the enable bit is not set, no interrupt request is generated. However, the interrupt flag may be polled.


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.2.11.4 Audio Mute (AMUTE) Function

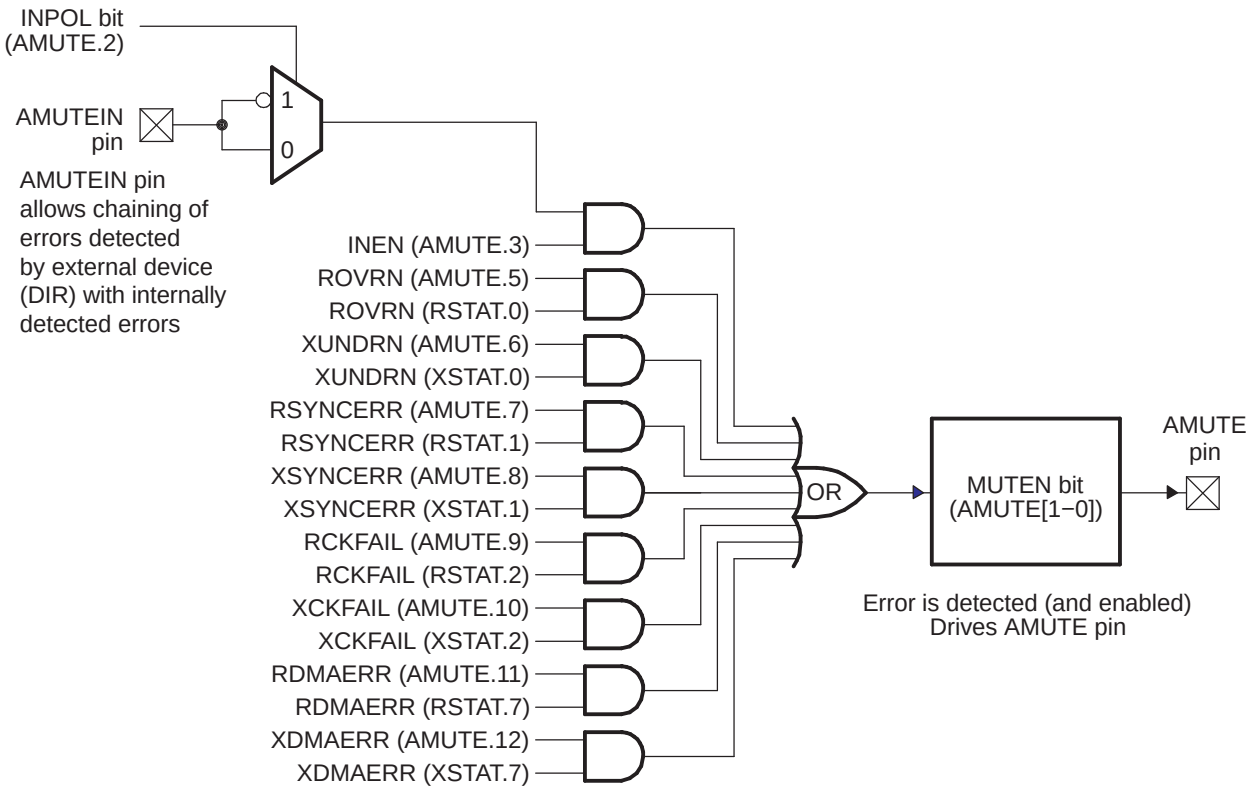
The McASP includes an automatic audio mute function (Figure 16-33) that asserts in hardware the AMUTE pin to a preprogrammed output state, as selected by the MUTEN bit in the audio mute control register (AMUTE). The AMUTE pin is asserted when one of the interrupt flags is set or an external device issues an error signal on the AMUTEIN input. Typically, the AMUTEIN input is shared with a device interrupt pin (for example EXT_INT4).

The AMUTEIN input allows the on-chip logic to consider a mute input from other devices in the system, so that all errors may be considered. The AMUTEIN input has a programmable polarity to allow it to adapt to different devices, as selected by the INPOL bit in AMUTE, and it must be enabled explicitly.

In addition to the external AMUTEIN input, the AMUTE pin output may be asserted when one of the error interrupt flags is set and its mute function is enabled in AMUTE.

When one or more of the errors is detected and enabled, the AMUTE pin is driven to an active state that is selected by MUTEN in AMUTE. The active polarity of the AMUTE pin is programmable by MUTEN (and the inactive polarity is the opposite of the active polarity). The AMUTE pin remains driven active until software clears all the error interrupt flags that are enabled to mute, and until the AMUTEIN is inactive.

**Figure 16-33. Audio Mute (AMUTE) Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 16.2.11.5 Multiple Interrupts

This only applies to interrupts and not to DMA requests. The following terms are defined:

- **Active Interrupt Request:** a flag in RSTAT or XSTAT is set and the interrupt is enabled in RINTCTL or XINTCTL.
- **Outstanding Interrupt Request:** An interrupt request has been issued on one of the McASP transmit/receive interrupt ports, but that request has not yet been serviced.
- **Serviced:** The CPU writes to RSTAT or XSTAT to clear one or more of the active interrupt request flags.

The first interrupt request to become active for the transmitter with the interrupt flag set in XSTAT and the interrupt enabled in XINTCTL generates a request on the McASP transmit interrupt port AXINT.

If more than one interrupt request becomes active in the same cycle, a single interrupt request is generated on the McASP transmit interrupt port. Subsequent interrupt requests that become active while the first interrupt request is outstanding do not immediately generate a new request pulse on the McASP transmit interrupt port.

The transmit interrupt is serviced with the CPU writing to XSTAT. If any interrupt requests are active after the write, a new request is generated on the McASP transmit interrupt port.

The receiver operates in a similar way, but using RSTAT, RINTCTL, and the McASP receive interrupt port ARINT.

One outstanding interrupt request is allowed on each port, so a transmit and a receive interrupt request may both be outstanding at the same time.

### *16.2.12 EDMA Event Support*

#### 16.2.12.1 EDMA Events

There are six EDMA events.

#### 16.2.12.2 Using the DMA for McASP Servicing

The most typical scenario is to use the DMA to service the McASP through the data port, although the DMA can also service the McASP through the configuration bus. Two possibilities exist for using the DMA events to service the McASP:

1. Use **AXEVT/AREVT**: Triggered upon each XDATA/RDATA transition from 0 to 1.
2. Use **AXEVTO/AREVTO** and **AXEVTE/AREVTE**: Alternating AXEVT/AREVT events for odd/even slots. Upon AXEVT/AREVT, AXEVTO/AREVTO is triggered if the event is for an odd channel, and AXEVTE/AREVTE is triggered if the event is for an even channel.

**NOTE:** Check the device-specific data manual to see if AXEVTO/AREVTO and AXEVTE/AREVTE are supported. These are optional.

Figure 16-34 and Figure 16-35 show an example audio system with six audio channels (LF, RF, LS, RS, C, and LFE) transmitted from three AXRn pins on the McASP. Figure 16-34 and Figure 16-35 show when events AXEVT, AXEVTO, and AXEVTE are triggered. Figure 16-34 and Figure 16-35 also apply for the receive audio channels and show when events AREVT, AREVTO, and AREVTE are triggered.

You can either use the DMA to service the McASP upon events AXEVT and AREVT (Figure 16-34) or upon events AXEVTO, AREVTO, AXEVTE, and AREVTE (Figure 16-35).


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 16-34. DMA Events in an Audio Example–Two Events (Scenario 1)**

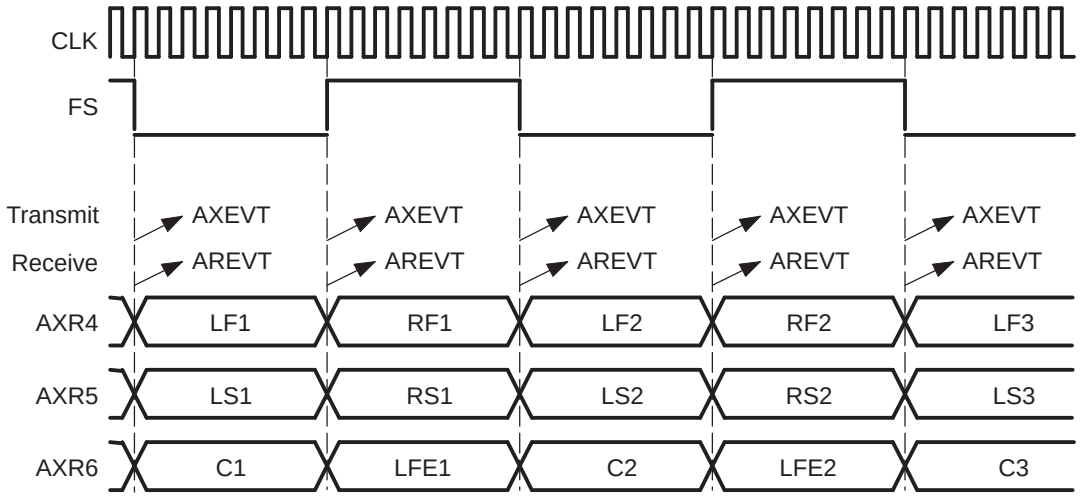
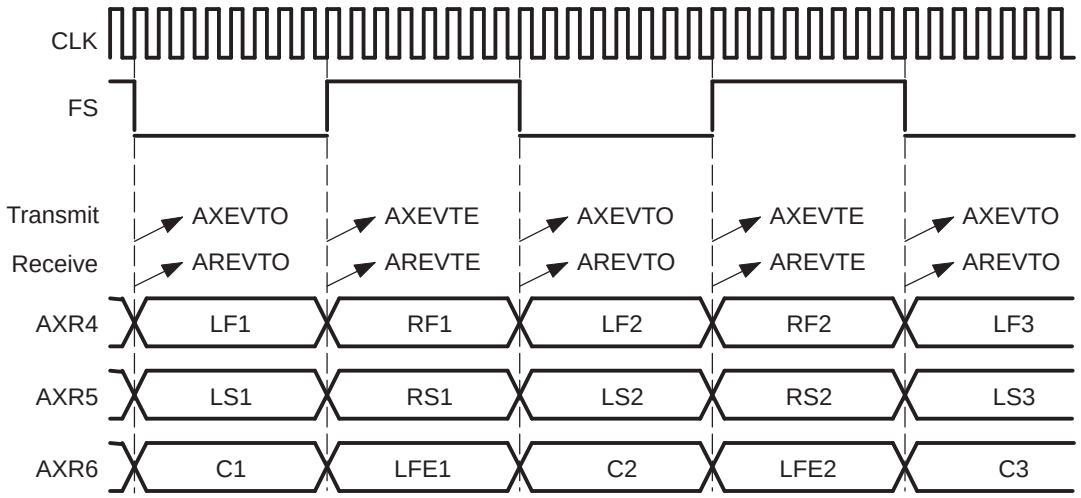



**Figure 16-35. DMA Events in an Audio Example–Four Events (Scenario 2)**




In scenario 1 (Figure 16-34), a DMA event AXEVT/AREVT is triggered on each time slot. In the example, AXEVT is triggered for each of the transmit audio channel time slot (Time slot for channels LF, LS, and C; and time slot for channels RF, RS, LFE). Similarly, AREVT is triggered for each of the receive audio channel time slot. Scenario 1 allows for the use of a single DMA to transmit all audio channels, and a single DMA to receive all audio channels.

In scenario 2 (Figure 16-35), two alternating DMA events are triggered for each time slot. In the example, AXEVTE (even) is triggered for the time slot for the even audio channels (LF, LS, C) and AXEVTO (odd) is triggered for the time slot for the odd audio channels (RF, RS, LFE). AXEVTO and AXEVTE alternate in time. The same is true in the receive direction with the use of AREVTO and AREVTE. This scenario allows for the use of two DMA channels (odd and even) to transmit all audio channels, and two DMA channels to receive all audio channels.


Copyright © 2011–2016, Texas Instruments Incorporated

Here are some guidelines on using the different DMA events:

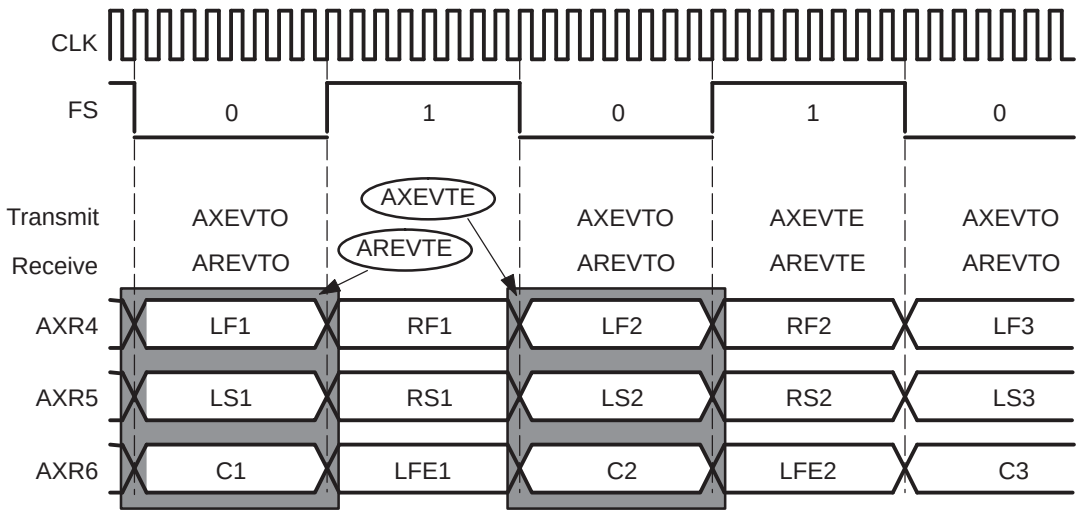
- Either use AXEVT, or the combination of AXEVTO and AXEVTE, to service the McASP. Never use all three at the same time. Similarly for receive, either use AREVT, or the combination of AREVTO and AREVTE.
- The McASP generates transmit DMA events independently from receive DMA events; therefore, separate schemes can be used for transmit and receive DMA. For example, scenario 1 could be used for the transmit data (AXEVT) and scenario 2 could be used for the receive data (AREVTO, AREVTE), and conversely.

Note the difference between DMA event generation and the CPU interrupt generation. DMA events are generated automatically upon data ready; whereas CPU interrupt generation needs to be enabled in the XINTCTL/RINTCTL register.

In Figure 16-35, scenario 2, each transmit DMA request is for data in the next time slot, while each receive DMA request is for data in the previous time slot. For example, Figure 16-36 shows a circled AXEVTE event for an even time slot transmit DMA request. The transmitter always requests a DMA transfer for data it will need to transmit during the next time slot. So, in this example, the circled event AXEVTE is a request for data for samples LF2, LS2, and C2.

On the other hand, the circled AREVTE event is an even time slot receive DMA request. The receiver always requests a DMA transfer for data it received during the previous time slot. In this example, the circled event AREVTE is a request for samples LF1, LS1, and C1.

**Figure 16-36. DMA Events in an Audio Example**




### 16.2.13 Power Management

The McASP can be placed in reduced power modes to conserve power during periods of low activity.

### 16.2.14 Emulation Considerations

**NOTE:** The receive buffer registers (RBUF*n*) and transmit buffer registers (XBUF*n*) should not be accessed by the emulator when the McASP is running. Such an access will cause the RRDY/XRDY bit in the serializer control register *n* (SRCTL*n*) to be updated.

The McASP does not support the emulation suspend. If the McASP is placed into a suspended state, software needs to ensure that the registers/internal logic is reset to its default values by performing a module power cycle before restarting the initialization sequence.


Copyright © 2011–2016, Texas Instruments Incorporated

## 16.3 McASP Registers

Table 16-7 lists the control registers for the McASP. The control registers are accessed through the configuration bus of the device. The receive buffer registers (RBUF*n*) and transmit buffer registers (XBUF*n*) can also be accessed through the data port of the device, as listed in Table 16-8.

Control registers for the McASP Audio FIFO (AFIFO) are summarized in Table 16-9. Note that the AFIFO Write FIFO (WFIFO) and Read FIFO (RFIFO) have independent control and status registers. The AFIFO control registers are accessed through the peripheral configuration port.

**Table 16-7. McASP Registers Accessed Through Configuration Bus**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 0h | REV | Revision identification register | Section 16.3.1 |
| 10h | PFUNC | Pin function register | Section 16.3.2 |
| 14h | PDIR | Pin direction register | Section 16.3.3 |
| 18h | PDOUT | Pin data output register | Section 16.3.4 |
| 1Ch | PDIN | Read returns: Pin data input register | Section 16.3.5 |
| 1Ch | PDSET | Writes affect: Pin data set register (alternate write address: PDOUT) | Section 16.3.6 |
| 20h | PDCLR | Pin data clear register (alternate write address: PDOUT) | Section 16.3.7 |
| 44h | GBLCTL | Global control register | Section 16.3.8 |
| 48h | AMUTE | Audio mute control register | Section 16.3.9 |
| 4Ch | DLBCTL | Digital loopback control register | Section 16.3.10 |
| 50h | DITCTL | DIT mode control register | Section 16.3.11 |
| 60h | RGBLCTL | Receiver global control register: Alias of GBLCTL, only receive bits are affected - allows receiver to be reset independently from transmitter | Section 16.3.12 |
| 64h | RMASK | Receive format unit bit mask register | Section 16.3.13 |
| 68h | RFMT | Receive bit stream format register | Section 16.3.14 |
| 6Ch | AFSRCTL | Receive frame sync control register | Section 16.3.15 |
| 70h | ACLKRCTL | Receive clock control register | Section 16.3.16 |
| 74h | AHCLKRCTL | Receive high-frequency clock control register | Section 16.3.17 |
| 78h | RTDM | Receive TDM time slot 0-31 register | Section 16.3.18 |
| 7Ch | RINTCTL | Receiver interrupt control register | Section 16.3.19 |
| 80h | RSTAT | Receiver status register | Section 16.3.20 |
| 84h | RSLOT | Current receive TDM time slot register | Section 16.3.21 |
| 88h | RCLKCHK | Receive clock check control register | Section 16.3.22 |
| 8Ch | REVTCTL | Receiver DMA event control register | Section 16.3.23 |
| A0h | XGBLCTL | Transmitter global control register. Alias of GBLCTL, only transmit bits are affected - allows transmitter to be reset independently from receiver | Section 16.3.24 |
| A4h | XMASK | Transmit format unit bit mask register | Section 16.3.25 |
| A8h | XFMT | Transmit bit stream format register | Section 16.3.26 |
| ACh | AFSXCTL | Transmit frame sync control register | Section 16.3.27 |
| B0h | ACLKXCTL | Transmit clock control register | Section 16.3.28 |
| B4h | AHCLKXCTL | Transmit high-frequency clock control register | Section 16.3.29 |
| B8h | XTDM | Transmit TDM time slot 0-31 register | Section 16.3.30 |
| BCh | XINTCTL | Transmitter interrupt control register | Section 16.3.31 |
| C0h | XSTAT | Transmitter status register | Section 16.3.32 |
| C4h | XSLOT | Current transmit TDM time slot register | Section 16.3.33 |
| C8h | XCLKCHK | Transmit clock check control register | Section 16.3.34 |
| CCh | XEVTCTL | Transmitter DMA event control register | Section 16.3.35 |
| 100h | DITCSRA0 | Left (even TDM time slot) channel status register (DIT mode) 0 | Section 16.3.37 |
| 104h | DITCSRA1 | Left (even TDM time slot) channel status register (DIT mode) 1 | Section 16.3.37 |
| 108h | DITCSRA2 | Left (even TDM time slot) channel status register (DIT mode) 2 | Section 16.3.37 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-7. McASP Registers Accessed Through Configuration Bus (continued)**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 10Ch | DITCSRA3 | Left (even TDM time slot) channel status register (DIT mode) 3 | Section 16.3.37 |
| 110h | DITCSRA4 | Left (even TDM time slot) channel status register (DIT mode) 4 | Section 16.3.37 |
| 114h | DITCSRA5 | Left (even TDM time slot) channel status register (DIT mode) 5 | Section 16.3.37 |
| 118h | DITCSRB0 | Right (odd TDM time slot) channel status register (DIT mode) 0 | Section 16.3.38 |
| 11Ch | DITCSRB1 | Right (odd TDM time slot) channel status register (DIT mode) 1 | Section 16.3.38 |
| 120h | DITCSRB2 | Right (odd TDM time slot) channel status register (DIT mode) 2 | Section 16.3.38 |
| 124h | DITCSRB3 | Right (odd TDM time slot) channel status register (DIT mode) 3 | Section 16.3.38 |
| 128h | DITCSRB4 | Right (odd TDM time slot) channel status register (DIT mode) 4 | Section 16.3.38 |
| 12Ch | DITCSRB5 | Right (odd TDM time slot) channel status register (DIT mode) 5 | Section 16.3.38 |
| 130h | DITUDRA0 | Left (even TDM time slot) channel user data register (DIT mode) 0 | Section 16.3.39 |
| 134h | DITUDRA1 | Left (even TDM time slot) channel user data register (DIT mode) 1 | Section 16.3.39 |
| 138h | DITUDRA2 | Left (even TDM time slot) channel user data register (DIT mode) 2 | Section 16.3.39 |
| 13Ch | DITUDRA3 | Left (even TDM time slot) channel user data register (DIT mode) 3 | Section 16.3.39 |
| 140h | DITUDRA4 | Left (even TDM time slot) channel user data register (DIT mode) 4 | Section 16.3.39 |
| 144h | DITUDRA5 | Left (even TDM time slot) channel user data register (DIT mode) 5 | Section 16.3.39 |
| 148h | DITUDRB0 | Right (odd TDM time slot) channel user data register (DIT mode) 0 | Section 16.3.40 |
| 14Ch | DITUDRB1 | Right (odd TDM time slot) channel user data register (DIT mode) 1 | Section 16.3.40 |
| 150h | DITUDRB2 | Right (odd TDM time slot) channel user data register (DIT mode) 2 | Section 16.3.40 |
| 154h | DITUDRB3 | Right (odd TDM time slot) channel user data register (DIT mode) 3 | Section 16.3.40 |
| 158h | DITUDRB4 | Right (odd TDM time slot) channel user data register (DIT mode) 4 | Section 16.3.40 |
| 15Ch | DITUDRB5 | Right (odd TDM time slot) channel user data register (DIT mode) 5 | Section 16.3.40 |
| 180h | SRCTL0 | Serializer control register 0 | Section 16.3.36 |
| 184h | SRCTL1 | Serializer control register 1 | Section 16.3.36 |
| 188h | SRCTL2 | Serializer control register 2 | Section 16.3.36 |
| 18Ch | SRCTL3 | Serializer control register 3 | Section 16.3.36 |
| 190h | SRCTL4 | Serializer control register 4 | Section 16.3.36 |
| 194h | SRCTL5 | Serializer control register 5 | Section 16.3.36 |
| 198h | SRCTL6 | Serializer control register 6 | Section 16.3.36 |
| 19Ch | SRCTL7 | Serializer control register 7 | Section 16.3.36 |
| 1A0h | SRCTL8 | Serializer control register 8 | Section 16.3.36 |
| 1A4h | SRCTL9 | Serializer control register 9 | Section 16.3.36 |
| 1A8h | SRCTL10 | Serializer control register 10 | Section 16.3.36 |
| 1ACh | SRCTL11 | Serializer control register 11 | Section 16.3.36 |
| 1B0h | SRCTL12 | Serializer control register 12 | Section 16.3.36 |
| 1B4h | SRCTL13 | Serializer control register 13 | Section 16.3.36 |
| 1B8h | SRCTL14 | Serializer control register 14 | Section 16.3.36 |
| 1BCh | SRCTL15 | Serializer control register 15 | Section 16.3.36 |
| 200h | XBUF0 | Transmit buffer register for serializer 0 | Section 16.3.41 |
| 204h | XBUF1 | Transmit buffer register for serializer 1 | Section 16.3.41 |
| 208h | XBUF2 | Transmit buffer register for serializer 2 | Section 16.3.41 |
| 20Ch | XBUF3 | Transmit buffer register for serializer 3 | Section 16.3.41 |
| 210h | XBUF4 | Transmit buffer register for serializer 4 | Section 16.3.41 |
| 214h | XBUF5 | Transmit buffer register for serializer 5 | Section 16.3.41 |
| 218h | XBUF6 | Transmit buffer register for serializer 6 | Section 16.3.41 |
| 21Ch | XBUF7 | Transmit buffer register for serializer 7 | Section 16.3.41 |
| 220h | XBUF8 | Transmit buffer register for serializer 8 | Section 16.3.41 |
| 224h | XBUF9 | Transmit buffer register for serializer 9 | Section 16.3.41 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-7. McASP Registers Accessed Through Configuration Bus (continued)**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 228h | XBUF10 | Transmit buffer register for serializer 10 | Section 16.3.41 |
| 22Ch | XBUF11 | Transmit buffer register for serializer 11 | Section 16.3.41 |
| 230h | XBUF12 | Transmit buffer register for serializer 12 | Section 16.3.41 |
| 234h | XBUF13 | Transmit buffer register for serializer 13 | Section 16.3.41 |
| 238h | XBUF14 | Transmit buffer register for serializer 14 | Section 16.3.41 |
| 23Ch | XBUF15 | Transmit buffer register for serializer 15 | Section 16.3.41 |
| 280h | RBUF0 | Receive buffer register for serializer 0 | Section 16.3.42 |
| 284h | RBUF1 | Receive buffer register for serializer 1 | Section 16.3.42 |
| 288h | RBUF2 | Receive buffer register for serializer 2 | Section 16.3.42 |
| 28Ch | RBUF3 | Receive buffer register for serializer 3 | Section 16.3.42 |
| 290h | RBUF4 | Receive buffer register for serializer 4 | Section 16.3.42 |
| 294h | RBUF5 | Receive buffer register for serializer 5 | Section 16.3.42 |
| 298h | RBUF6 | Receive buffer register for serializer 6 | Section 16.3.42 |
| 29Ch | RBUF7 | Receive buffer register for serializer 7 | Section 16.3.42 |
| 2A0h | RBUF8 | Receive buffer register for serializer 8 | Section 16.3.42 |
| 2A4h | RBUF9 | Receive buffer register for serializer 9 | Section 16.3.42 |
| 2A8h | RBUF10 | Receive buffer register for serializer 10 | Section 16.3.42 |
| 2ACh | RBUF11 | Receive buffer register for serializer 11 | Section 16.3.42 |
| 2B0h | RBUF12 | Receive buffer register for serializer 12 | Section 16.3.42 |
| 2B4h | RBUF13 | Receive buffer register for serializer 13 | Section 16.3.42 |
| 2B8h | RBUF14 | Receive buffer register for serializer 14 | Section 16.3.42 |
| 2BCh | RBUF15 | Receive buffer register for serializer 15 | Section 16.3.42 |

**Table 16-8. McASP Registers Accessed Through Data Port**

| Hex Address | Register Name | Register Description |
|---|---|---|
| Read Accesses | RBUF[1] | Receive buffer data peripheral register address. Cycles through receive serializers, skipping over transmit serializers and inactive serializers. Starts at the lowest serializer at the beginning of each time slot. DAT BUS only if XBUSEL = 0. See the device specific data manual for address location. |
| Write Accesses | XBUF[1] | Transmit buffer data peripheral register address. Cycles through transmit serializers, skipping over receive and inactive serializers. Starts at the lowest serializer at the beginning of each time slot. DAT BUS only if RBUSEL = 0. See the device specific data manual for address location. |

(1) RBUF and XBUF are at the same address location. Reads access RBUF and writes access XBUF.

**Table 16-9. McASP AFIFO Registers Accessed Through Peripheral Configuration Port**

| Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 1000h | WFIFOCTL | Write FIFO control register | Section 16.3.43 |
| 1004h | WFIFOSTS | Write FIFO status register | Section 16.3.44 |
| 1008h | RFIFOCTL | Read FIFO control register | Section 16.3.45 |
| 100Ch | RFIFOSTS | Read FIFO status register | Section 16.3.46 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.1 Revision Identification Register (REV)

The revision identification register (REV) contains identification data for the peripheral. The REV is shown in Figure 16-37 and described in Table 16-10.

**Figure 16-37. Revision Identification Register (REV)**

| 31 | 0 |
|---|---|
| REV | |
| R-4430 7302h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 16-10. Revision Identification Register (REV) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | REV | 4430 7302h | Identifies revision of peripheral. |

### 16.3.2 Pin Function Register (PFUNC)

The pin function register (PFUNC) specifies the function of AXRn, ACLKX, AHCLKX, AFSX, ACLKR, AHCLKR, and AFSR pins as either a McASP pin or a general-purpose input/output (GPIO) pin. The PFUNC is shown in Figure 16-38 and described in Table 16-11.

**CAUTION**

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

**Figure 16-38. Pin Function Register (PFUNC)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| AFSR | AHCLKR | ACLKR | AFSX | AHCLKX | ACLKX | AMUTE | Reserved |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 |

| 23 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| AXR15 | AXR14 | AXR13 | AXR12 | AXR11 | AXR10 | AXR9 | AXR8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| AXR7 | AXR6 | AXR5 | AXR4 | AXR3 | AXR2 | AXR1 | AXR0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-11. Pin Function Register (PFUNC) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | AFSR | | Determines if AFSR pin functions as McASP or GPIO. |
| | | 0 | Pin functions as McASP pin. |
| | | 1 | Pin functions as GPIO pin. |
| 30 | AHCLKR | | Determines if AHCLKR pin functions as McASP or GPIO. |
| | | 0 | Pin functions as McASP pin. |
| | | 1 | Pin functions as GPIO pin. |
| 29 | ACLKR | | Determines if ACLKR pin functions as McASP or GPIO. |
| | | 0 | Pin functions as McASP pin. |
| | | 1 | Pin functions as GPIO pin. |
| 28 | AFSX | | Determines if AFSX pin functions as McASP or GPIO. |
| | | 0 | Pin functions as McASP pin. |
| | | 1 | Pin functions as GPIO pin. |
| 27 | AHCLKX | | Determines if AHCLKX pin functions as McASP or GPIO. |
| | | 0 | Pin functions as McASP pin. |
| | | 1 | Pin functions as GPIO pin. |
| 26 | ACLKX | | Determines if ACLKX pin functions as McASP or GPIO. |
| | | 0 | Pin functions as McASP pin. |
| | | 1 | Pin functions as GPIO pin. |
| 25 | AMUTE | | Determines if AMUTE pin functions as McASP or GPIO. |
| | | 0 | Pin functions as McASP pin. |
| | | 1 | Pin functions as GPIO pin. |
| 24-16 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 15-0 | AXR[15-0] | | Determines if AXRn pin functions as McASP or GPIO. |
| | | 0 | Pin functions as McASP pin. |
| | | 1 | Pin functions as GPIO pin. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.3 Pin Direction Register (PDIR)

The pin direction register (PDIR) specifies the direction of AXRn, ACLKX, AHCLKX, AFSX, ACLKR, AHCLKR, and AFSR pins as either an input or an output pin. The PDIR is shown in Figure 16-39 and described in Table 16-12. Refer to the device-specific data manual for the number of pins available on your device.

Regardless of the pin function register (PFUNC) setting, each PDIR bit must be set to 1 for the specified pin to be enabled as an output and each PDIR bit must be cleared to 0 for the specified pin to be an input.

For example, if the McASP is configured to use an internally-generated bit clock and the clock is to be driven out to the system, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be set to 1 (an output).

When AXRn is configured to transmit, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be set to 1 (an output). Similarly, when AXRn is configured to receive, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be cleared to 0 (an input).

**CAUTION**

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

**Figure 16-39. Pin Direction Register (PDIR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| AFSR | AHCLKR | ACLKR | AFSX | AHCLKX | ACLKX | AMUTE | Reserved |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 |

| 23 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| AXR15 | AXR14 | AXR13 | AXR12 | AXR11 | AXR10 | AXR9 | AXR8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| AXR7 | AXR6 | AXR5 | AXR4 | AXR3 | AXR2 | AXR1 | AXR0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-12. Pin Direction Register (PDIR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | AFSR | | Determines if AFSR pin functions as an input or output. |
| | | 0 | Pin functions as input. |
| | | 1 | Pin functions as output. |
| 30 | AHCLKR | | Determines if AHCLKR pin functions as an input or output. |
| | | 0 | Pin functions as input. |
| | | 1 | Pin functions as output. |
| 29 | ACLKR | | Determines if ACLKR pin functions as an input or output. |
| | | 0 | Pin functions as input. |
| | | 1 | Pin functions as output. |
| 28 | AFSX | | Determines if AFSX pin functions as an input or output. |
| | | 0 | Pin functions as input. |
| | | 1 | Pin functions as output. |
| 27 | AHCLKX | | Determines if AHCLKX pin functions as an input or output. |
| | | 0 | Pin functions as input. |
| | | 1 | Pin functions as output. |
| 26 | ACLKX | | Determines if ACLKX pin functions as an input or output. |
| | | 0 | Pin functions as input. |
| | | 1 | Pin functions as output. |
| 25 | AMUTE | | Determines if AMUTE pin functions as an input or output. |
| | | 0 | Pin functions as input. |
| | | 1 | Pin functions as output. |
| 24-16 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 15-0 | AXR[15-0] | | Determines if AXRn pin functions as an input or output. |
| | | 0 | Pin functions as input. |
| | | 1 | Pin functions as output. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.4 Pin Data Output Register (PDOUT)

The pin data output register (PDOUT) holds a value for data out at all times, and may be read back at all times. The value held by PDOUT is not affected by writing to PDIR and PFUNC. However, the data value in PDOUT is driven out onto the McASP pin only if the corresponding bit in PFUNC is set to 1 (GPIO function) and the corresponding bit in PDIR is set to 1 (output).

When reading data, it returns the corresponding PDOUT[n] bit value and does not return the input from I/O pin. When writing data, it writes to the corresponding PDOUT[n] bit.

The PDOUT is shown in Figure 16-40 and described in Table 16-13.

PDOUT has these aliases or alternate addresses:

- PDSET - when written to at this address, writing a 1 to a bit in PDSET sets the corresponding bit in PDOUT to 1; writing a 0 has no effect and keeps the bits in PDOUT unchanged.
- PDCLR - when written to at this address, writing a 1 to a bit in PDCLR clears the corresponding bit in PDOUT to 0; writing a 0 has no effect and keeps the bits in PDOUT unchanged.

There is only one set of data out bits, PDOUT[31-0]. The other registers, PDSET and PDCLR, are just different addresses for the same control bits, with different behaviors during writes.

**CAUTION**

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

**Figure 16-40. Pin Data Output Register (PDOUT)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| AFSR | AHCLKR | ACLKR | AFSX | AHCLKX | ACLKX | AMUTE | Reserved |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 |

| 23 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| AXR15 | AXR14 | AXR13 | AXR12 | AXR11 | AXR10 | AXR9 | AXR8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| AXR7 | AXR6 | AXR5 | AXR4 | AXR3 | AXR2 | AXR1 | AXR0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-13. Pin Data Output Register (PDOUT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | AFSR | | Determines drive on AFSR output pin when the corresponding PFUNC[31] and PDIR[31] bits are set to 1. |
| | | 0 | Pin drives low. |
| | | 1 | Pin drives high. |
| 30 | AHCLKR | | Determines drive on AHCLKR output pin when the corresponding PFUNC[30] and PDIR[30] bits are set to 1. |
| | | 0 | Pin drives low. |
| | | 1 | Pin drives high. |
| 29 | ACLKR | | Determines drive on ACLKR output pin when the corresponding PFUNC[29] and PDIR[29] bits are set to 1. |
| | | 0 | Pin drives low. |
| | | 1 | Pin drives high. |
| 28 | AFSX | | Determines drive on AFSX output pin when the corresponding PFUNC[28] and PDIR[28] bits are set to 1. |
| | | 0 | Pin drives low. |
| | | 1 | Pin drives high. |
| 27 | AHCLKX | | Determines drive on AHCLKX output pin when the corresponding PFUNC[27] and PDIR[27] bits are set to 1. |
| | | 0 | Pin drives low. |
| | | 1 | Pin drives high. |
| 26 | ACLKX | | Determines drive on ACLKX output pin when the corresponding PFUNC[26] and PDIR[26] bits are set to 1. |
| | | 0 | Pin drives low. |
| | | 1 | Pin drives high. |
| 25 | AMUTE | | Determines drive on AMUTE output pin when the corresponding PFUNC[25] and PDIR[25] bits are set to 1. |
| | | 0 | Pin drives low. |
| | | 1 | Pin drives high. |
| 24-16 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 15-0 | AXR[15-0] | | Determines drive on AXR[n] output pin when the corresponding PFUNC[n] and PDIR[n] bits are set to 1. |
| | | 0 | Pin drives low. |
| | | 1 | Pin drives high. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.5 Pin Data Input Register (PDIN)

The pin data input register (PDIN) holds the I/O pin state of each of the McASP pins. PDIN allows the actual value of the pin to be read, regardless of the state of PFUNC and PDIR. The value after reset for registers 1 through 15 and 24 through 31 depends on how the pins are being driven. The PDIN is shown in Figure 16-41 and described in Table 16-14.

**CAUTION**

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

**Figure 16-41. Pin Data Input Register (PDIN)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| AFSR | AHCLKR | ACLKR | AFSX | AHCLKX | ACLKX | AMUTE | Reserved |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 |

| 23 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| AXR15 | AXR14 | AXR13 | AXR12 | AXR11 | AXR10 | AXR9 | AXR8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| AXR7 | AXR6 | AXR5 | AXR4 | AXR3 | AXR2 | AXR1 | AXR0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-14. Pin Data Input Register (PDIN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | AFSR | | Logic level on AFSR pin. |
| | | 0 | Pin is logic low. |
| | | 1 | Pin is logic high. |
| 30 | AHCLKR | | Logic level on AHCLKR pin. |
| | | 0 | Pin is logic low. |
| | | 1 | Pin is logic high. |
| 29 | ACLKR | | Logic level on ACLKR pin. |
| | | 0 | Pin is logic low. |
| | | 1 | Pin is logic high. |
| 28 | AFSX | | Logic level on AFSX pin. |
| | | 0 | Pin is logic low. |
| | | 1 | Pin is logic high. |
| 27 | AHCLKX | | Logic level on AHCLKX pin. |
| | | 0 | Pin is logic low. |
| | | 1 | Pin is logic high. |
| 26 | ACLKX | | Logic level on ACLKX pin. |
| | | 0 | Pin is logic low. |
| | | 1 | Pin is logic high. |
| 25 | AMUTE | | Logic level on AMUTE pin. |
| | | 0 | Pin is logic low. |
| | | 1 | Pin is logic high. |
| 24-16 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 15-0 | AXR[15-0] | | Logic level on AXR[n] pin. |
| | | 0 | Pin is logic low. |
| | | 1 | Pin is logic high. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.6 Pin Data Set Register (PDSET)

The pin data set register (PDSET) is an alias of the pin data output register (PDOUT) for writes only. Writing a 1 to the PDSET bit sets the corresponding bit in PDOUT and, if PFUNC = 1 (GPIO function) and PDIR = 1 (output), drives a logic high on the pin. PDSET is useful for a multitasking system because it allows you to set to a logic high only the desired pin(s) within a system without affecting other I/O pins controlled by the same McASP. The PDSET is shown in Figure 16-42 and described in Table 16-15.

**CAUTION**

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

**Figure 16-42. Pin Data Set Register (PDSET)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| AFSR | AHCLKR | ACLKR | AFSX | AHCLKX | ACLKX | AMUTE | Reserved |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 |

| 23 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| AXR15 | AXR14 | AXR13 | AXR12 | AXR11 | AXR10 | AXR9 | AXR8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AXR7 | AXR6 | AXR5 | AXR4 | AXR3 | AXR2 | AXR1 | AXR0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-15. Pin Data Set Register (PDSET) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | AFSR | | Allows the corresponding AFSR bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[31] bit is set to 1. |
| 30 | AHCLKR | | Allows the corresponding AHCLKR bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[30] bit is set to 1. |
| 29 | ACLKR | | Allows the corresponding ACLKR bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[29] bit is set to 1. |
| 28 | AFSX | | Allows the corresponding AFSX bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[28] bit is set to 1. |
| 27 | AHCLKX | | Allows the corresponding AHCLKX bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[27] bit is set to 1. |
| 26 | ACLKX | | Allows the corresponding ACLKX bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[26] bit is set to 1. |
| 25 | AMUTE | | Allows the corresponding AMUTE bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[25] bit is set to 1. |
| 24-16 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 15-0 | AXR[15-0] | | Allows the corresponding AXR[n] bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[n] bit is set to 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.7 Pin Data Clear Register (PDCLR)

The pin data clear register (PDCLR) is an alias of the pin data output register (PDOUT) for writes only. Writing a 1 to the PDCLR bit clears the corresponding bit in PDOUT and, if PFUNC = 1 (GPIO function) and PDIR = 1 (output), drives a logic low on the pin. PDCLR is useful for a multitasking system because it allows you to clear to a logic low only the desired pin(s) within a system without affecting other I/O pins controlled by the same McASP. The PDCLR is shown in Figure 16-43 and described in Table 16-16.

**CAUTION**

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

**Figure 16-43. Pin Data Clear Register (PDCLR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| AFSR | AHCLKR | ACLKR | AFSX | AHCLKX | ACLKX | AMUTE | Reserved |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 |

| 23 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| AXR15 | AXR14 | AXR13 | AXR12 | AXR11 | AXR10 | AXR9 | AXR8 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| AXR7 | AXR6 | AXR5 | AXR4 | AXR3 | AXR2 | AXR1 | AXR0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-16. Pin Data Clear Register (PDCLR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | AFSR | | Allows the corresponding AFSR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[31] bit is cleared to 0. |
| 30 | AHCLKR | | Allows the corresponding AHCLKR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[30] bit is cleared to 0. |
| 29 | ACLKR | | Allows the corresponding ACLKR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[29] bit is cleared to 0. |
| 28 | AFSX | | Allows the corresponding AFSX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[28] bit is cleared to 0. |
| 27 | AHCLKX | | Allows the corresponding AHCLKX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[27] bit is cleared to 0. |
| 26 | ACLKX | | Allows the corresponding ACLKX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[26] bit is cleared to 0. |
| 25 | AMUTE | | Allows the corresponding AMUTE bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[25] bit is cleared to 0. |
| 24-16 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 15-0 | AXR[15-0] | | Allows the corresponding AXR[n] bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. |
| | | 0 | No effect. |
| | | 1 | PDOUT[n] bit is cleared to 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.8 Global Control Register (GBLCTL)

The global control register (GBLCTL) provides initialization of the transmit and receive sections. The GBLCTL is shown in Figure 16-44 and described in Table 16-17.

The bit fields in GBLCTL are synchronized and latched by the corresponding clocks (ACLKX for bits 12-8 and ACLKR for bits 4-0). Before GBLCTL is programmed, you must ensure that serial clocks are running. If the corresponding external serial clocks, ACLKX and ACLKR, are not yet running, you should select the internal serial clock source in AHCLKXCTL, AHCLKRCTL, ACLKXCTL, and ACLKRCTL before GBLCTL is programmed. Also, after programming any bits in GBLCTL you should not proceed until you have read back from GBLCTL and verified that the bits are latched in GBLCTL.

**Figure 16-44. Global Control Register (GBLCTL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| Reserved | | XFRST | XSMRST | XSRCLR | XHCLKRST | XCLKRST |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | RFRST | RSMRST | RSRCLR | RHCLKRST | RCLKRST |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-17. Global Control Register (GBLCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-13 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 12 | XFRST | | Transmit frame sync generator reset enable bit. |
| | | 0 | Transmit frame sync generator is reset. |
| | | 1 | Transmit frame sync generator is active. When released from reset, the transmit frame sync generator begins counting serial clocks and generating frame sync as programmed. |
| 11 | XSMRST | | Transmit state machine reset enable bit. |
| | | 0 | Transmit state machine is held in reset. AXRn pin state:<br>If PFUNC[n] = 0 and PDIR[n] = 1; then the serializer drives the AXRn pin to the state specified for inactive time slot (as determined by DISMOD bits in SRCTL). |
| | | 1 | Transmit state machine is released from reset. When released from reset, the transmit state machine immediately transfers data from XRBUF[n] to XRSR[n]. The transmit state machine sets the underrun flag (XUNDRN) in XSTAT, if XRBUF[n] have not been preloaded with data before reset is released. The transmit state machine also immediately begins detecting frame sync and is ready to transmit. Transmit TDM time slot begins at slot 0 after reset is released. |
| 10 | XSRCLR | | Transmit serializer clear enable bit. By clearing then setting this bit, the transmit buffer is flushed to an empty state (XDATA = 1). If XSMRST = 1, XSRCLR = 1, XDATA = 1, and XBUF is not loaded with new data before the start of the next active time slot, an underrun will occur. |
| | | 0 | Transmit serializers are cleared. |
| | | 1 | Transmit serializers are active. When the transmit serializers are first taken out of reset (XSRCLR changes from 0 to 1), the transmit data ready bit (XDATA) in XSTAT is set to indicate XBUF is ready to be written. |
| 9 | XHCLKRST | | Transmit high-frequency clock divider reset enable bit. |
| | | 0 | Transmit high-frequency clock divider is held in reset and passes through its input as divide-by-1. |
| | | 1 | Transmit high-frequency clock divider is running. |
| 8 | XCLKRST | | Transmit clock divider reset enable bit. |
| | | 0 | Transmit clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input. |
| | | 1 | Transmit clock divider is running. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-17. Global Control Register (GBLCTL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7-5 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 4 | RFRST | | Receive frame sync generator reset enable bit. |
| | | 0 | Receive frame sync generator is reset. |
| | | 1 | Receive frame sync generator is active. When released from reset, the receive frame sync generator begins counting serial clocks and generating frame sync as programmed. |
| 3 | RSMRST | | Receive state machine reset enable bit. |
| | | 0 | Receive state machine is held in reset. |
| | | 1 | Receive state machine is released from reset. When released from reset, the receive state machine immediately begins detecting frame sync and is ready to receive.<br>Receive TDM time slot begins at slot 0 after reset is released. |
| 2 | RSRCLR | | Receive serializer clear enable bit. By clearing then setting this bit, the receive buffer is flushed. |
| | | 0 | Receive serializers are cleared. |
| | | 1 | Receive serializers are active. |
| 1 | RHCLKRST | | Receive high-frequency clock divider reset enable bit. |
| | | 0 | Receive high-frequency clock divider is held in reset and passes through its input as divide-by-1. |
| | | 1 | Receive high-frequency clock divider is running. |
| 0 | RCLKRST | | Receive clock divider reset enable bit. |
| | | 0 | Receive clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input. |
| | | 1 | Receive clock divider is running. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.9 Audio Mute Control Register (AMUTE)

The audio mute control register (AMUTE) controls the McASP audio mute (AMUTE) output pin. The value after reset for register 4 depends on how the pins are being driven. The AMUTE is shown in Figure 16-45 and described in Table 16-18.

**Figure 16-45. Audio Mute Control Register (AMUTE)**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | XDMAERR | RDMAERR | XCKFAIL | RCKFAIL | XSYNCERR |
| R-0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RSYNCERR | XUNDRN | ROVRN | INSTAT | INEN | INPOL | MUTEN | |
| R/W-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-18. Audio Mute Control Register (AMUTE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-13 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 12 | XDMAERR | | If transmit DMA error (XDMAERR), drive AMUTE active enable bit. |
| | | 0 | Drive is disabled. Detection of transmit DMA error is ignored by AMUTE. |
| | | 1 | Drive is enabled (active). Upon detection of transmit DMA error, AMUTE is active and is driven according to MUTEN bit. |
| 11 | RDMAERR | | If receive DMA error (RDMAERR), drive AMUTE active enable bit. |
| | | 0 | Drive is disabled. Detection of receive DMA error is ignored by AMUTE. |
| | | 1 | Drive is enabled (active). Upon detection of receive DMA error, AMUTE is active and is driven according to MUTEN bit. |
| 10 | XCKFAIL | | If transmit clock failure (XCKFAIL), drive AMUTE active enable bit. |
| | | 0 | Drive is disabled. Detection of transmit clock failure is ignored by AMUTE. |
| | | 1 | Drive is enabled (active). Upon detection of transmit clock failure, AMUTE is active and is driven according to MUTEN bit |
| 9 | RCKFAIL | | If receive clock failure (RCKFAIL), drive AMUTE active enable bit. |
| | | 0 | Drive is disabled. Detection of receive clock failure is ignored by AMUTE. |
| | | 1 | Drive is enabled (active). Upon detection of receive clock failure, AMUTE is active and is driven according to MUTEN bit. |
| 8 | XSYNCERR | | If unexpected transmit frame sync error (XSYNCERR), drive AMUTE active enable bit. |
| | | 0 | Drive is disabled. Detection of unexpected transmit frame sync error is ignored by AMUTE. |
| | | 1 | Drive is enabled (active). Upon detection of unexpected transmit frame sync error, AMUTE is active and is driven according to MUTEN bit. |
| 7 | RSYNCERR | | If unexpected receive frame sync error (RSYNCERR), drive AMUTE active enable bit. |
| | | 0 | Drive is disabled. Detection of unexpected receive frame sync error is ignored by AMUTE. |
| | | 1 | Drive is enabled (active). Upon detection of unexpected receive frame sync error, AMUTE is active and is driven according to MUTEN bit. |
| 6 | XUNDRN | | If transmit underrun error (XUNDRN), drive AMUTE active enable bit. |
| | | 0 | Drive is disabled. Detection of transmit underrun error is ignored by AMUTE. |
| | | 1 | Drive is enabled (active). Upon detection of transmit underrun error, AMUTE is active and is driven according to MUTEN bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-18. Audio Mute Control Register (AMUTE) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 5 | ROVRN | | If receiver overrun error (ROVRN), drive AMUTE active enable bit. |
| | | 0 | Drive is disabled. Detection of receiver overrun error is ignored by AMUTE. |
| | | 1 | Drive is enabled (active). Upon detection of receiver overrun error, AMUTE is active and is driven according to MUTEN bit. |
| 4 | INSTAT | | Status of mute in pin, determines drive on AXRn pin when PFUNC[n] and PDIR[n] bits are set to 1. |
| | | 0 | AMUTEIN pin is inactive. |
| | | 1 | AMUTEIN pin is active. Audio mute in error is detected. |
| 3 | INEN | | Drive AMUTE active when AMUTEIN error is active (INSTAT = 1). |
| | | 0 | Drive is disabled. AMUTEIN is ignored by AMUTE. |
| | | 1 | Drive is enabled (active). INSTAT = 1 drives AMUTE active. |
| 2 | INPOL | | Audio mute in (AMUTEIN) polarity select bit. |
| | | 0 | Polarity is active high. A high on AMUTEIN sets INSTAT to 1. |
| | | 1 | Polarity is active low. A low on AM UTEIN sets INSTAT to 1. |
| 1-0 | MUTEN | 0-3h | AMUTE pin enable bit (unless overridden by GPIO registers). |
| | | 0 | AMUTE pin is disabled, pin goes to tri-state condition. |
| | | 1h | AMUTE pin is driven high if error is detected. |
| | | 2h | AMUTE pin is driven low if error is detected. |
| | | 3h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.10 Digital Loopback Control Register (DLBCTL)

The digital loopback control register (DLBCTL) controls the internal loopback settings of the McASP in TDM mode. The DLBCTL is shown in Figure 16-46 and described in Table 16-19. Note that loopback is NOT supported if McASP is configured in DIT mode.

**Figure 16-46. Digital Loopback Control Register (DLBCTL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | MODE | | ORD | DLBEN |
| R-0 | | R/W-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-19. Digital Loopback Control Register (DLBCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 3-2 | MODE | 0-3h | Loopback generator mode bits. Applies only when loopback mode is enabled (DLBEN = 1). |
| | | 0 | Default and reserved on loopback mode (DLBEN = 1).<br>When in non-loopback mode (DLBEN = 0), MODE should be left at default (00).<br>When in loopback mode (DLBEN = 1), MODE = 00 is reserved and is not applicable. |
| | | 1h | MODE must be set to 01 when McASP operates in loopback mode (DLBEN = 1). This is necessary to allow transmit clock and frame sync generators to be used by both transmit and receive sections. |
| | | 2h-3h | Reserved. |
| 1 | ORD | | Loopback order bit when loopback mode is enabled (DLBEN = 1). |
| | | 0 | Odd serializers N + 1 transmit to even serializers N that receive. The corresponding serializers must be programmed properly. |
| | | 1 | Even serializers N transmit to odd serializers N + 1 that receive. The corresponding serializers must be programmed properly. |
| 0 | DLBEN | | Loopback mode enable bit. |
| | | 0 | Loopback mode is disabled (normal McASP operation) |
| | | 1 | Loopback mode is enabled (TDM mode only). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.11 Digital Mode Control Register (DITCTL)

The DIT mode control register (DITCTL) controls DIT operations of the McASP. The DITCTL is shown in Figure 16-47 and described in Table 16-20.

**Figure 16-47. Digital Mode Control Register (DITCTL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | VB | VA | Rsvd | DITEN |
| R-0 | | R/W-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-20. Digital Mode Control Register (DITCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 3 | VB | | Valid bit for odd time slots (DIT right subframe). |
| | | 0 | V bit is 0 during odd DIT subframes. |
| | | 1 | V bit is 1 during odd DIT subframes. |
| 2 | VA | | Valid bit for even time slots (DIT left subframe). |
| | | 0 | V bit is 0 during even DIT subframes. |
| | | 1 | V bit is 1 during even DIT subframes. |
| 1 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 0 | DITEN | | DIT mode enable bit. DITEN should only be changed while the XSMRST bit in GBLCTL is in reset (and for startup, XSRCLR also in reset). However, it is not necessary to reset the XCLKRST or XHCLKRST bits in GBLCTL to change DITEN. |
| | | 0 | DIT mode is disabled. Transmitter operates in TDM or burst mode. |
| | | 1 | DIT mode is enabled. Transmitter operates in DIT encoded mode. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.12 Receiver Global Control Register (RGBLCTL)

Alias of the global control register (GBLCTL). Writing to the receiver global control register (RGBLCTL) affects only the receive bits of GBLCTL (bits 4-0). Reads from RGBLCTL return the value of GBLCTL. RGBLCTL allows the receiver to be reset independently from the transmitter. The RGBLCTL is shown in Figure 16-48 and described in Table 16-21. See Section 16.3.8 for a detailed description of GBLCTL.

**Figure 16-48. Receiver Global Control Register (RGBLCTL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| Reserved | | XFRST | XSMRST | XSRCLR | XHCLKRST | XCLKRST |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | RFRST | RSMRST | RSRCLR | RHCLKRST | RCLKRST |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-21. Receiver Global Control Register (RGBLCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-13 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 12 | XFRST | x | Transmit frame sync generator reset enable bit. A read of this bit returns the XFRST bit value of GBLCTL. Writes have no effect. |
| 11 | XSMRST | x | Transmit state machine reset enable bit. A read of this bit returns the XSMRST bit value of GBLCTL. Writes have no effect. |
| 10 | XSRCLR | x | Transmit serializer clear enable bit. A read of this bit returns the XSRCLR bit value of GBLCTL. Writes have no effect. |
| 9 | XHCLKRST | x | Transmit high-frequency clock divider reset enable bit. A read of this bit returns the XHCLKRST bit value of GBLCTL. Writes have no effect. |
| 8 | XCLKRST | x | Transmit clock divider reset enable bit. A read of this bit returns the XCLKRST bit value of GBLCTL. Writes have no effect. |
| 7-5 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 4 | RFRST | | Receive frame sync generator reset enable bit. A write to this bit affects the RFRST bit of GBLCTL. |
| | | 0 | Receive frame sync generator is reset. |
| | | 1 | Receive frame sync generator is active. |
| 3 | RSMRST | | Receive state machine reset enable bit. A write to this bit affects the RSMRST bit of GBLCTL. |
| | | 0 | Receive state machine is held in reset. |
| | | 1 | Receive state machine is released from reset. |
| 2 | RSRCLR | | Receive serializer clear enable bit. A write to this bit affects the RSRCLR bit of GBLCTL. |
| | | 0 | Receive serializers are cleared. |
| | | 1 | Receive serializers are active. |
| 1 | RHCLKRST | | Receive high-frequency clock divider reset enable bit. A write to this bit affects the RHCLKRST bit of GBLCTL. |
| | | 0 | Receive high-frequency clock divider is held in reset and passes through its input as divide-by-1. |
| | | 1 | Receive high-frequency clock divider is running. |
| 0 | RCLKRST | | Receive clock divider reset enable bit. A write to this bit affects the RCLKRST bit of GBLCTL. |
| | | 0 | Receive clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input. |
| | | 1 | Receive clock divider is running. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.13 Receive Format Unit Bit Mask Register (RMASK)

The receive format unit bit mask register (RMASK) determines which bits of the received data are masked off and padded with a known value before being read by the CPU or DMA. The RMASK is shown in Figure 16-49 and described in Table 16-22.

**Figure 16-49. Receive Format Unit Bit Mask Register (RMASK)**

| 31 | 0 |
|---|---|
| RMASK[*n*] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 16-22. Receive Format Unit Bit Mask Register (RMASK) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | RMASK[31-0] | | Receive data mask *n* enable bit. |
| | | 0 | Corresponding bit of receive data (after passing through reverse and rotate units) is masked out and then padded with the selected bit pad value (RPAD and RPBIT bits in RFMT). |
| | | 1 | Corresponding bit of receive data (after passing through reverse and rotate units) is returned to CPU or DMA. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.14 Receive Bit Stream Format Register (RFMT)

The receive bit stream format register (RFMT) configures the receive data format. The RFMT is shown in Figure 16-50 and described in Table 16-23.

**Figure 16-50. Receive Bit Stream Format Register (RFMT)**

| 31 | 18 | 17 | 16 |
|---|---|---|---|
| Reserved | | RDATDLY | |
| R-0 | | R/W-0 | |

| 15 | 14 | 13 | 12 | 8 | 7 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| RRVRS | RPAD | | RPBIT | | RSSZ | | RBUSEL | RROT | |
| R/W-0 | R/W-0 | | R/W-0 | | R/W-0 | | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-23. Receive Bit Stream Format Register (RFMT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-18 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 17-16 | RDATDLY | 0-3h | Receive frame sync delay of AXRn. |
| | | 0 | 0-bit delay. The first receive data bit, AXRn, occurs in same ACLKR cycle as the receive frame sync (AFSR). |
| | | 1h | 1-bit delay. The first receive data bit, AXRn, occurs one ACLKR cycle after the receive frame sync (AFSR). |
| | | 2h | 2-bit delay. The first receive data bit, AXRn, occurs two ACLKR cycles after the receive frame sync (AFSR). |
| | | 3h | Reserved. |
| 15 | RRVRS | | Receive serial bitstream order. |
| | | 0 | Bitstream is LSB first. No bit reversal is performed in receive format bit reverse unit. |
| | | 1 | Bitstream is MSB first. Bit reversal is performed in receive format bit reverse unit. |
| 14-13 | RPAD | 0-3h | Pad value for extra bits in slot not belonging to the word. This field only applies to bits when RMASK[n] = 0. |
| | | 0 | Pad extra bits with 0. |
| | | 1h | Pad extra bits with 1. |
| | | 2h | Pad extra bits with one of the bits from the word as specified by RPBIT bits. |
| | | 3h | Reserved. |
| 12-8 | RPBIT | 0-1Fh | RPBIT value determines which bit (as read by the CPU or DMA from RBUF[n]) is used to pad the extra bits. This field only applies when RPAD = 2h. |
| | | 0 | Pad with bit 0 value. |
| | | 1h-1Fh | Pad with bit 1 to bit 31 value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-23. Receive Bit Stream Format Register (RFMT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7-4 | RSSZ | 0-Fh | Receive slot size. |
| | | 0-2h | Reserved |
| | | 3h | Slot size is 8 bits. |
| | | 4h | Reserved |
| | | 5h | Slot size is 12 bits. |
| | | 6h | Reserved |
| | | 7h | Slot size is 16 bits. |
| | | 8h | Reserved |
| | | 9h | Slot size is 20 bits. |
| | | Ah | Reserved |
| | | Bh | Slot size is 24 bits |
| | | Ch | Reserved |
| | | Dh | Slot size is 28 bits. |
| | | Eh | Reserved |
| | | Fh | Slot size is 32 bits. |
| 3 | RBUSEL | | Selects whether reads from serializer buffer XRBUF[n] originate from the configuration bus (CFG) or the data (DAT) port. |
| | | 0 | Reads from XRBUF[n] originate on data port. Reads from XRBUF[n] on configuration bus are ignored. |
| | | 1 | Reads from XRBUF[n] originate on configuration bus. Reads from XRBUF[n] on data port are ignored. |
| 2-0 | RROT | 0-7h | Right-rotation value for receive rotate right format unit. |
| | | 0 | Rotate right by 0 (no rotation). |
| | | 1h | Rotate right by 4 bit positions. |
| | | 2h | Rotate right by 8 bit positions. |
| | | 3h | Rotate right by 12 bit positions. |
| | | 4h | Rotate right by 16 bit positions. |
| | | 5h | Rotate right by 20 bit positions. |
| | | 6h | Rotate right by 24 bit positions. |
| | | 7h | Rotate right by 28 bit positions. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.15 Receive Frame Sync Control Register (AFSRCTL)

The receive frame sync control register (AFSRCTL) configures the receive frame sync (AFSR). The AFSRCTL is shown in Figure 16-51 and described in Table 16-24.

**Figure 16-51. Receive Frame Sync Control Register (AFSRCTL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| RMOD | | Reserved | | FRWID | Reserved | | FSRM | FSRP |
| R/W-0 | | R-0 | | R/W-0 | R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-24. Receive Frame Sync Control Register (AFSRCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 15-7 | RMOD | 0-1FFh | Receive frame sync mode select bits. |
| | | 0 | Burst mode. |
| | | 1h | Reserved. |
| | | 2h-20h | 2-slot TDM (I2S mode) to 32-slot TDM. |
| | | 21h-17Fh | Reserved. |
| | | 180h | 384-slot TDM (external DIR IC inputting 384-slot DIR frames to McASP over I2S interface). |
| | | 181h-1FFh | Reserved. |
| 6-5 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 4 | FRWID | | Receive frame sync width select bit indicates the width of the receive frame sync (AFSR) during its active period. |
| | | 0 | Single bit. |
| | | 1 | Single word. Single word is not supported if RMOD is set to burst mode. |
| 3-2 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 1 | FSRM | | Receive frame sync generation select bit. |
| | | 0 | Externally-generated receive frame sync. |
| | | 1 | Internally-generated receive frame sync. |
| 0 | FSRP | | Receive frame sync polarity select bit. |
| | | 0 | A rising edge on receive frame sync (AFSR) indicates the beginning of a frame. |
| | | 1 | A falling edge on receive frame sync (AFSR) indicates the beginning of a frame. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.16 Receive Clock Control Register (ACLKRCTL)

The receive clock control register (ACLKRCTL) configures the receive bit clock (ACLKR) and the receive clock generator. The ACLKRCTL is shown in Figure 16-52 and described in Table 16-25.

**Figure 16-52. Receive Clock Control Register (ACLKRCTL)**

| 31 | | | | | | 16 |
|---|---|---|---|---|---|---|
| Reserved | | | | | | |
| R-0 | | | | | | |
| **15** | **8** | **7** | **6** | **5** | **4** | **0** |
| Reserved | | CLKRP | Rsvd | CLKRM | CLKRDIV | |
| R-0 | | R/W-0 | R-0 | R/W-1 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-25. Receive Clock Control Register (ACLKRCTL) Field Descriptions**

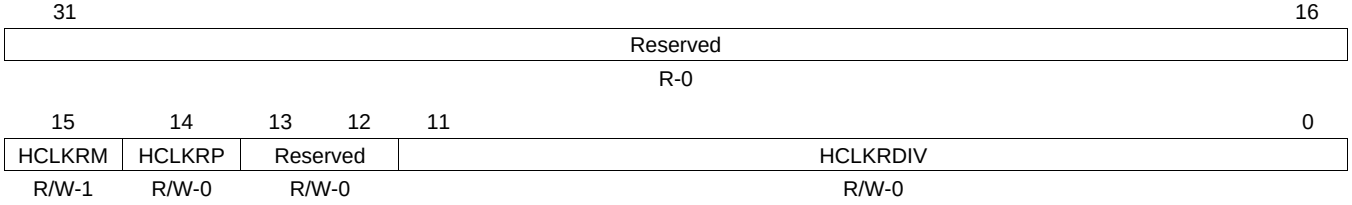
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 7 | CLKRP | | Receive bitstream clock polarity select bit. |
| | | 0 | Falling edge. Receiver samples data on the falling edge of the serial clock, so the external transmitter driving this receiver must shift data out on the rising edge of the serial clock. |
| | | 1 | Rising edge. Receiver samples data on the rising edge of the serial clock, so the external transmitter driving this receiver must shift data out on the falling edge of the serial clock. |
| 6 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 5 | CLKRM | | Receive bit clock source bit. Note that this bit does not have any effect, if ACLKXCTL.ASYNC = 0. |
| | | 0 | External receive clock source from ACLKR pin. |
| | | 1 | Internal receive clock source from output of programmable bit clock divider. |
| 4-0 | CLKRDIV | 0-1Fh | Receive bit clock divide ratio bits determine the divide-down ratio from AHCLKR to ACLKR. Note that this bit does not have any effect, if ACLKXCTL.ASYNC = 0. |
| | | 0 | Divide-by-1. |
| | | 1h | Divide-by-2. |
| | | 2h-1Fh | Divide-by-3 to divide-by-32. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.17 Receive High-Frequency Clock Control Register (AHCLKRCTL)

The receive high-frequency clock control register (AHCLKRCTL) configures the receive high-frequency master clock (AHCLKR) and the receive clock generator. The AHCLKRCTL is shown in Figure 16-53 and described in Table 16-26.

**Figure 16-53. Receive High-Frequency Clock Control Register (AHCLKRCTL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-26. Receive High-Frequency Clock Control Register (AHCLKRCTL) Field Descriptions**

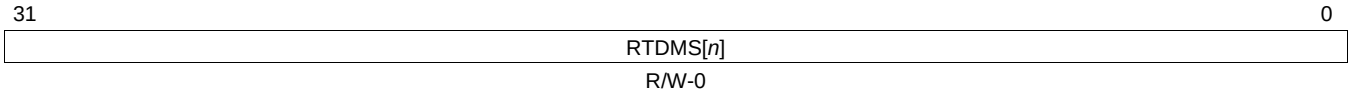
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 15 | HCLKRM | | Receive high-frequency clock source bit. |
| | | 0 | External receive high-frequency clock source from AHCLKR pin. |
| | | 1 | Internal receive high-frequency clock source from output of programmable high clock divider. |
| 14 | HCLKRP | | Receive bitstream high-frequency clock polarity select bit. |
| | | 0 | AHCLKR is not inverted before programmable bit clock divider. In the special case where the receive bit clock (ACLKR) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKRDIV = 0 in ACLKRCTL), AHCLKR is directly passed through to the ACLKR pin. |
| | | 1 | AHCLKR is inverted before programmable bit clock divider. In the special case where the receive bit clock (ACLKR) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKRDIV = 0 in ACLKRCTL), AHCLKR is directly passed through to the ACLKR pin. |
| 13-12 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 11-0 | HCLKRDIV | 0-FFFh | Receive high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKR. |
| | | 0 | Divide-by-1. |
| | | 1h | Divide-by-2. |
| | | 2h-FFFh | Divide-by-3 to divide-by-4096. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.18 Receive TDM Time Slot Register (RTDM)

The receive TDM time slot register (RTDM) specifies which TDM time slot the receiver is active. The RTDM is shown in Figure 16-54 and described in Table 16-27.

**Figure 16-54. Receive TDM Time Slot Register (RTDM)**




LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 16-27. Receive TDM Time Slot Register (RTDM) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | RTDMS[31-0] | | Receiver mode during TDM time slot *n*. |
| | | 0 | Receive TDM time slot *n* is inactive. The receive serializer does not shift in data during this slot. |
| | | 1 | Receive TDM time slot *n* is active. The receive serializer shifts in data during this slot. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.19 Receiver Interrupt Control Register (RINTCTL)

The receiver interrupt control register (RINTCTL) controls generation of the McASP receive interrupt (RINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates RINT. The RINTCTL is shown in Figure 16-55 and described in Table 16-28. See Section 16.3.20 for a description of the interrupt conditions.

**Figure 16-55. Receiver Interrupt Control Register (RINTCTL)**

| 31 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RSTAFRM | Reserved | RDATA | RLAST | RDMAERR | RCKFAIL | RSYNCERR | ROVRN |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-28. Receiver Interrupt Control Register (RINTCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 7 | RSTAFRM | | Receive start of frame interrupt enable bit. |
| | | 0 | Interrupt is disabled. A receive start of frame interrupt does not generate a McASP receive interrupt (RINT). |
| | | 1 | Interrupt is enabled. A receive start of frame interrupt generates a McASP receive interrupt (RINT). |
| 6 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 5 | RDATA | | Receive data ready interrupt enable bit. |
| | | 0 | Interrupt is disabled. A receive data ready interrupt does not generate a McASP receive interrupt (RINT). |
| | | 1 | Interrupt is enabled. A receive data ready interrupt generates a McASP receive interrupt (RINT). |
| 4 | RLAST | | Receive last slot interrupt enable bit. |
| | | 0 | Interrupt is disabled. A receive last slot interrupt does not generate a McASP receive interrupt (RINT). |
| | | 1 | Interrupt is enabled. A receive last slot interrupt generates a McASP receive interrupt (RINT). |
| 3 | RDMAERR | | Receive DMA error interrupt enable bit. |
| | | 0 | Interrupt is disabled. A receive DMA error interrupt does not generate a McASP receive interrupt (RINT). |
| | | 1 | Interrupt is enabled. A receive DMA error interrupt generates a McASP receive interrupt (RINT). |
| 2 | RCKFAIL | | Receive clock failure interrupt enable bit. |
| | | 0 | Interrupt is disabled. A receive clock failure interrupt does not generate a McASP receive interrupt (RINT). |
| | | 1 | Interrupt is enabled. A receive clock failure interrupt generates a McASP receive interrupt (RINT). |
| 1 | RSYNCERR | | Unexpected receive frame sync interrupt enable bit. |
| | | 0 | Interrupt is disabled. An unexpected receive frame sync interrupt does not generate a McASP receive interrupt (RINT). |
| | | 1 | Interrupt is enabled. An unexpected receive frame sync interrupt generates a McASP receive interrupt (RINT). |
| 0 | ROVRN | | Receiver overrun interrupt enable bit. |
| | | 0 | Interrupt is disabled. A receiver overrun interrupt does not generate a McASP receive interrupt (RINT). |
| | | 1 | Interrupt is enabled. A receiver overrun interrupt generates a McASP receive interrupt (RINT). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.20 Receiver Status Register (RSTAT)

The receiver status register (RSTAT) provides the receiver status and receive TDM time slot number. If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes a new interrupt request to be generated. The RSTAT is shown in Figure 16-56 and described in Table 16-29.

**Figure 16-56. Receiver Status Register (RSTAT)**

| 31 | | | | | | | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | RERR |
| R-0 | | | | | | | | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | | 0 |
| RDMAERR | RSTAFRM | RDATA | RLAST | RTDMSLOT | RCKFAIL | RSYNCERR | | ROVRN |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R-0 | R/W1C-0 | R/W1C-0 | | R/W1C-0 |

LEGEND: R/W = Read/Write; R = Read only; W1C = Writing a 1 clears this bit; writing a 0 has no effect.; -*n* = value after reset

**Table 16-29. Receiver Status Register (RSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 8 | RERR | | RERR bit always returns a logic-OR of:ROVRN \| RSYNCERR \| RCKFAIL \| RDMAERR<br>Allows a single bit to be checked to determine if a receiver error interrupt has occurred. |
| | | 0 | No errors have occurred. |
| | | 1 | An error has occurred. |
| 7 | RDMAERR | | Receive DMA error flag. RDMAERR is set when the CPU or DMA reads more serializers through the data port in a given time slot than were programmed as receivers. Causes a receive interrupt (RINT), if this bit is set and RDMAERR in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. |
| | | 0 | Receive DMA error did not occur. |
| | | 1 | Receive DMA error did occur. |
| 6 | RSTAFRM | | Receive start of frame flag. Causes a receive interrupt (RINT), if this bit is set and RSTAFRM in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. |
| | | 0 | No new receive frame sync (AFSR) is detected. |
| | | 1 | A new receive frame sync (AFSR) is detected. |
| 5 | RDATA | | Receive data ready flag. Causes a receive interrupt (RINT), if this bit is set and RDATA in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. |
| | | 0 | No new data in RBUF. |
| | | 1 | Data is transferred from XRSR to RBUF and ready to be serviced by the CPU or DMA. When RDATA is set, it always causes a DMA event (AREVT). |
| 4 | RLAST | | Receive last slot flag. RLAST is set along with RDATA, if the current slot is the last slot in a frame. Causes a receive interrupt (RINT), if this bit is set and RLAST in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. |
| | | 0 | Current slot is not the last slot in a frame. |
| | | 1 | Current slot is the last slot in a frame. RDATA is also set. |
| 3 | RTDMSLOT | | Returns the LSB of RSLOT. Allows a single read of RSTAT to determine whether the current TDM time slot is even or odd. |
| | | 0 | Current TDM time slot is odd. |
| | | 1 | Current TDM time slot is even. |
| 2 | RCKFAIL | | Receive clock failure flag. RCKFAIL is set when the receive clock failure detection circuit reports an error (see *Clock Failure Detection*). Causes a receive interrupt (RINT), if this bit is set and RCKFAIL in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. |
| | | 0 | Receive clock failure did not occur. |
| | | 1 | Receive clock failure did occur. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-29. Receiver Status Register (RSTAT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | RSYNCERR | | Unexpected receive frame sync flag. RSYNCERR is set when a new receive frame sync (AFSR) occurs before it is expected. Causes a receive interrupt (RINT), if this bit is set and RSYNCERR in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. |
| | | 0 | Unexpected receive frame sync did not occur. |
| | | 1 | Unexpected receive frame sync did occur. |
| 0 | ROVRN | | Receiver overrun flag. ROVRN is set when the receive serializer is instructed to transfer data from XRSR to RBUF, but the former data in RBUF has not yet been read by the CPU or DMA. Causes a receive interrupt (RINT), if this bit is set and ROVRN in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. |
| | | 0 | Receiver overrun did not occur. |
| | | 1 | Receiver overrun did occur. |

### 16.3.21 Current Receive TDM Time Slot Registers (RSLOT)

The current receive TDM time slot register (RSLOT) indicates the current time slot for the receive data frame. The RSLOT is shown in Figure 16-57 and described in Table 16-30.

**Figure 16-57. Current Receive TDM Time Slot Registers (RSLOT)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 9 | 8 | 0 |
|---|---|---|---|
| Reserved | | RSLOTCNT | |
| R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 16-30. Current Receive TDM Time Slot Registers (RSLOT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 8-0 | RSLOTCNT | 0-17Fh | Current receive time slot count. Legal values: 0 to 383 (17Fh).<br>TDM function is not supported for > 32 time slots. However, TDM time slot counter may count to 383 when used to receive a DIR block (transferred over TDM format). In I2S mode, this should only be read by the DSP since frame sync initializes it to zero. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.22 Receive Clock Check Control Register (RCLKCHK)

The receive clock check control register (RCLKCHK) configures the receive clock failure detection circuit. The RCLKCHK is shown in Figure 16-58 and described in Table 16-31.

**Figure 16-58. Receive Clock Check Control Register (RCLKCHK)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| RCNT | | RMAX | |
| R-0 | | R/W-0 | |

| 15 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| RMIN | | Reserved | | RPS | |
| R/W-0 | | R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-31. Receive Clock Check Control Register (RCLKCHK) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | RCNT | 0-FFh | Receive clock count value (from previous measurement). The clock circuit continually counts the number of system clocks for every 32 receive high-frequency master clock (AHCLKR) signals, and stores the count in RCNT until the next measurement is taken. |
| 23-16 | RMAX | 0-FFh | Receive clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If the current counter value is greater than RMAX after counting 32 AHCLKR signals, RCKFAIL in RSTAT is set. The comparison is performed using unsigned arithmetic. |
| 15-8 | RMIN | 0-FFh | Receive clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If RCNT is less than RMIN after counting 32 AHCLKR signals, RCKFAIL in RSTAT is set. The comparison is performed using unsigned arithmetic. |
| 7-4 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 3-0 | RPS | 0-Fh | Receive clock check prescaler value. |
| | | 0 | McASP system clock divided by 1. |
| | | 1h | McASP system clock divided by 2. |
| | | 2h | McASP system clock divided by 4. |
| | | 3h | McASP system clock divided by 8. |
| | | 4h | McASP system clock divided by 16. |
| | | 5h | McASP system clock divided by 32. |
| | | 6h | McASP system clock divided by 64. |
| | | 7h | McASP system clock divided by 128. |
| | | 8h | McASP system clock divided by 256. |
| | | 9h-Fh | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

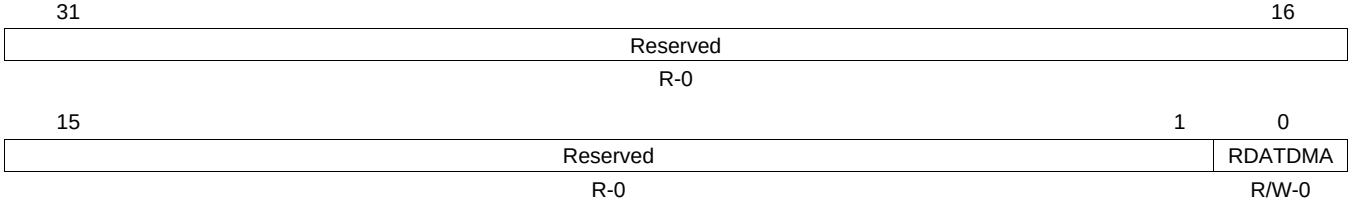
### 16.3.23 Receiver DMA Event Control Register (REVTCTL)

The receiver DMA event control register (REVTCTL) contains a disable bit for the receiver DMA event. The REVTCTL is shown in Figure 16-59 and described in Table 16-32.

**NOTE: Device-specific registers**

Accessing REVTCTL not implemented on a specific device may cause improper device operation.

**Figure 16-59. Receiver DMA Event Control Register (REVTCTL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-32. Receiver DMA Event Control Register (REVTCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 0 | RDATDMA | | Receive data DMA request enable bit. If writing to this bit, always write the default value of 0. |
| | | 0 | Receive data DMA request is enabled. |
| | | 1 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.24 Transmitter Global Control Register (XGBLCTL)

Alias of the global control register (GBLCTL). Writing to the transmitter global control register (XGBLCTL) affects only the transmit bits of GBLCTL (bits 12-8). Reads from XGBLCTL return the value of GBLCTL. XGBLCTL allows the transmitter to be reset independently from the receiver. The XGBLCTL is shown in Figure 16-60 and described in Table 16-33. See Section 16.3.8 for a detailed description of GBLCTL.

**Figure 16-60. Transmitter Global Control Register (XGBLCTL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| Reserved | | XFRST | XSMRST | XSRCLR | XHCLKRST | XCLKRST |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | RFRST | RSMRST | RSRCLR | RHCLKRST | RCLKRST |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-33. Transmitter Global Control Register (XGBLCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-13 | Reserved | 0-FFh | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 12 | XFRST | | Transmit frame sync generator reset enable bit. A write to this bit affects the XFRST bit of GBLCTL. |
| | | 0 | Transmit frame sync generator is reset. |
| | | 1 | Transmit frame sync generator is active. |
| 11 | XSMRST | | Transmit state machine reset enable bit. A write to this bit affects the XSMRST bit of GBLCTL. |
| | | 0 | Transmit state machine is held in reset. |
| | | 1 | Transmit state machine is released from reset. |
| 10 | XSRCLR | | Transmit serializer clear enable bit. A write to this bit affects the XSRCLR bit of GBLCTL. |
| | | 0 | Transmit serializers are cleared. |
| | | 1 | Transmit serializers are active. |
| 9 | XHCLKRST | | Transmit high-frequency clock divider reset enable bit. A write to this bit affects the XHCLKRST bit of GBLCTL. |
| | | 0 | Transmit high-frequency clock divider is held in reset and passes through its input as divide-by-1. |
| | | 1 | Transmit high-frequency clock divider is running. |
| 8 | XCLKRST | | Transmit clock divider reset enable bit. A write to this bit affects the XCLKRST bit of GBLCTL. |
| | | 0 | Transmit clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input. |
| | | 1 | Transmit clock divider is running. |
| 7-5 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 4 | RFRST | x | Receive frame sync generator reset enable bit. A read of this bit returns the RFRST bit value of GBLCTL. Writes have no effect. |
| 3 | RSMRST | x | Receive state machine reset enable bit. A read of this bit returns the RSMRST bit value of GBLCTL. Writes have no effect. |
| 2 | RSRCLR | x | Receive serializer clear enable bit. A read of this bit returns the RSRSCLR bit value of GBLCTL. Writes have no effect. |
| 1 | RHCLKRST | x | Receive high-frequency clock divider reset enable bit. A read of this bit returns the RHCLKRST bit value of GBLCTL. Writes have no effect. |
| 0 | RCLKRST | x | Receive clock divider reset enable bit. A read of this bit returns the RCLKRST bit value of GBLCTL. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.25 Transmit Format Unit Bit Mask Register (XMASK)

The transmit format unit bit mask register (XMASK) determines which bits of the transmitted data are masked off and padded with a known value before being shifted out the McASP. The XMASK is shown in Figure 16-61 and described in Table 16-34.

**Figure 16-61. Transmit Format Unit Bit Mask Register (XMASK)**

| 31 | 0 |
|---|---|
| XMASK[*n*] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 16-34. Transmit Format Unit Bit Mask Register (XMASK) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | XMASK[31-0] | | Transmit data mask *n* enable bit. |
| | | 0 | Corresponding bit of transmit data (before passing through reverse and rotate units) is masked out and then padded with the selected bit pad value (XPAD and XPBIT bits in XFMT), which is transmitted out the McASP in place of the original bit. |
| | | 1 | Corresponding bit of transmit data (before passing through reverse and rotate units) is transmitted out the McASP. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.26 Transmit Bit Stream Format Register (XFMT)

The transmit bit stream format register (XFMT) configures the transmit data format. The XFMT is shown in Figure 16-62 and described in Table 16-35.

**Figure 16-62. Transmit Bit Stream Format Register (XFMT)**

| 31 | 18 | 17 | 16 |
|---|---|---|---|
| Reserved | | XDATDLY | |
| R-0 | | R/W-0 | |

| 15 | 14 | 13 | 12 | 8 | 7 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| XRVRS | XPAD | | XPBIT | | XSSZ | | XBUSEL | XROT | |
| R/W-0 | R/W-0 | | R/W-0 | | R/W-0 | | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-35. Transmit Bit Stream Format Register (XFMT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-18 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 17-16 | XDATDLY | 0-3h | Transmit frame sync delay of AXRn. |
| | | 0 | 0-bit delay. The first transmit data bit, AXRn, occurs in same ACLKX cycle as the transmit frame sync (AFSX). |
| | | 1h | 1-bit delay. The first transmit data bit, AXRn, occurs one ACLKX cycle after the transmit frame sync (AFSX). |
| | | 2h | 2-bit delay. The first transmit data bit, AXRn, occurs two ACLKX cycles after the transmit frame sync (AFSX). |
| | | 3h | Reserved. |
| 15 | XRVRS | | Transmit serial bitstream order. |
| | | 0 | Bitstream is LSB first. No bit reversal is performed in transmit format bit reverse unit. |
| | | 1 | Bitstream is MSB first. Bit reversal is performed in transmit format bit reverse unit. |
| 14-13 | XPAD | 0-3h | Pad value for extra bits in slot not belonging to word defined by XMASK. This field only applies to bits when XMASK[n] = 0. |
| | | 0 | Pad extra bits with 0. |
| | | 1h | Pad extra bits with 1. |
| | | 2h | Pad extra bits with one of the bits from the word as specified by XPBIT bits. |
| | | 3h | Reserved. |
| 12-8 | XPBIT | 0-1Fh | XPBIT value determines which bit (as written by the CPU or DMA to XBUF[n]) is used to pad the extra bits before shifting. This field only applies when XPAD = 2h. |
| | | 0 | Pad with bit 0 value. |
| | | 1-1Fh | Pad with bit 1 to bit 31 value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-35. Transmit Bit Stream Format Register (XFMT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7-4 | XSSZ | 0-Fh | Transmit slot size. |
| | | 0-2h | Reserved. |
| | | 3h | Slot size is 8 bits. |
| | | 4h | Reserved. |
| | | 5h | Slot size is 12 bits. |
| | | 6h | Reserved. |
| | | 7h | Slot size is 16 bits. |
| | | 8h | Reserved. |
| | | 9h | Slot size is 20 bits. |
| | | Ah | Reserved. |
| | | Bh | Slot size is 24 bits. |
| | | Ch | Reserved. |
| | | Dh | Slot size is 28 bits. |
| | | Eh | Reserved. |
| | | Fh | Slot size is 32 bits. |
| 3 | XBUSEL | | Selects whether writes to serializer buffer XRBUF[n] originate from the configuration bus (CFG) or the data (DAT) port. |
| | | 0 | Writes to XRBUF[n] originate from the data port. Writes to XRBUF[n] from the configuration bus are ignored with no effect to the McASP. |
| | | 1 | Writes to XRBUF[n] originate from the configuration bus. Writes to XRBUF[n] from the data port are ignored with no effect to the McASP. |
| 2-0 | XROT | 0-7h | Right-rotation value for transmit rotate right format unit. |
| | | 0 | Rotate right by 0 (no rotation). |
| | | 1h | Rotate right by 4 bit positions. |
| | | 2h | Rotate right by 8 bit positions. |
| | | 3h | Rotate right by 12 bit positions. |
| | | 4h | Rotate right by 16 bit positions. |
| | | 5h | Rotate right by 20 bit positions. |
| | | 6h | Rotate right by 24 bit positions. |
| | | 7h | Rotate right by 28 bit positions. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.27 Transmit Frame Sync Control Register (AFSXCTL)

The transmit frame sync control register (AFSXCTL) configures the transmit frame sync (AFSX). The AFSXCTL is shown in Figure 16-63 and described in Table 16-36.

**Figure 16-63. Transmit Frame Sync Control Register (AFSXCTL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| XMOD | | Reserved | | FXWID | Reserved | | FSXM | FSXP |
| R/W-0 | | R-0 | | R/W-0 | R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-36. Transmit Frame Sync Control Register (AFSXCTL) Field Descriptions**

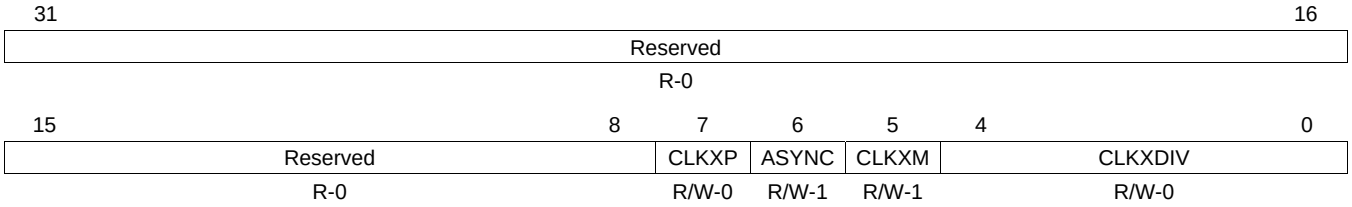
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 15-7 | XMOD | 0-1FFh | Transmit frame sync mode select bits. |
| | | 0 | Burst mode. |
| | | 1h | Reserved. |
| | | 2h-20h | 2-slot TDM (I2S mode) to 32-slot TDM. |
| | | 21h-17Fh | Reserved. |
| | | 180h | 384-slot DIT mode. |
| | | 181h-1FFh | Reserved. |
| 6-5 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 4 | FXWID | | Transmit frame sync width select bit indicates the width of the transmit frame sync (AFSX) during its active period. |
| | | 0 | Single bit. |
| | | 1 | Single word. Single word is not supported if XMOD is set to burst mode. |
| 3-2 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 1 | FSXM | | Transmit frame sync generation select bit. |
| | | 0 | Externally-generated transmit frame sync. |
| | | 1 | Internally-generated transmit frame sync. |
| 0 | FSXP | | Transmit frame sync polarity select bit. |
| | | 0 | A rising edge on transmit frame sync (AFSX) indicates the beginning of a frame. |
| | | 1 | A falling edge on transmit frame sync (AFSX) indicates the beginning of a frame. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.28 Transmit Clock Control Register (ACLKXCTL)

The transmit clock control register (ACLKXCTL) configures the transmit bit clock (ACLKX) and the transmit clock generator. The ACLKXCTL is shown in Figure 16-64 and described in Table 16-37.

**Figure 16-64. Transmit Clock Control Register (ACLKXCTL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-37. Transmit Clock Control Register (ACLKXCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 7 | CLKXP | | Transmit bitstream clock polarity select bit. |
| | | 0 | Rising edge. External receiver samples data on the falling edge of the serial clock, so the transmitter must shift data out on the rising edge of the serial clock. |
| | | 1 | Falling edge. External receiver samples data on the rising edge of the serial clock, so the transmitter must shift data out on the falling edge of the serial clock. |
| 6 | ASYNC | | Transmit/receive operation asynchronous enable bit. |
| | | 0 | Synchronous. Transmit clock and frame sync provides the source for both the transmit and receive sections. Note that in this mode, the receive bit clock is an inverted version of the transmit bit clock. |
| | | 1 | Asynchronous. Separate clock and frame sync used by transmit and receive sections. |
| 5 | CLKXM | | Transmit bit clock source bit. |
| | | 0 | External transmit clock source from ACLKX pin. |
| | | 1 | Internal transmit clock source from output of programmable bit clock divider. |
| 4-0 | CLKXDIV | 0-1Fh | Transmit bit clock divide ratio bits determine the divide-down ratio from AHCLKX to ACLKX. |
| | | 0 | Divide-by-1. |
| | | 1h | Divide-by-2. |
| | | 2h-1Fh | Divide-by-3 to divide-by-32. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.29 Transmit High-Frequency Clock Control Register (AHCLKXCTL)

The transmit high-frequency clock control register (AHCLKXCTL) configures the transmit high-frequency master clock (AHCLKX) and the transmit clock generator. The AHCLKXCTL is shown in Figure 16-65 and described in Table 16-38.

**Figure 16-65. Transmit High-Frequency Clock Control Register (AHCLKXCTL)**

| 31 | | | | | 16 |
|---|---|---|---|---|---|
| Reserved | | | | | |
| R-0 | | | | | |

| 15 | 14 | 13 | 12 | 11 | 0 |
|---|---|---|---|---|---|
| HCLKXM | HCLKXP | Reserved | | HCLKXDIV | |
| R/W-1 | R/W-0 | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-38. Transmit High-Frequency Clock Control Register (AHCLKXCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 15 | HCLKXM | | Transmit high-frequency clock source bit. |
| | | 0 | External transmit high-frequency clock source from AHCLKX pin. |
| | | 1 | Internal transmit high-frequency clock source from output of programmable high clock divider. |
| 14 | HCLKXP | | Transmit bitstream high-frequency clock polarity select bit. |
| | | 0 | AHCLKX is not inverted before programmable bit clock divider. In the special case where the transmit bit clock (ACLKX) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKXDIV = 0 in ACLKXCTL), AHCLKX is directly passed through to the ACLKX pin. |
| | | 1 | AHCLKX is inverted before programmable bit clock divider. In the special case where the transmit bit clock (ACLKX) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKXDIV = 0 in ACLKXCTL), AHCLKX is directly passed through to the ACLKX pin. |
| 13-12 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 11-0 | HCLKXDIV | 0-FFFh | Transmit high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKX. |
| | | 0 | Divide-by-1. |
| | | 1h | Divide-by-2. |
| | | 2h-FFFh | Divide-by-3 to divide-by-4096. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.30 Transmit TDM Time Slot Register (XTDM)

The transmit TDM time slot register (XTDM) specifies in which TDM time slot the transmitter is active. TDM time slot counter range is extended to 384 slots (to support SPDIF blocks of 384 subframes). XTDM operates modulo 32, that is, XTDMS specifies the TDM activity for time slots 0, 32, 64, 96, 128, etc. The XTDM is shown in Figure 16-66 and described in Table 16-39.

**Figure 16-66. Transmit TDM Time Slot Register (XTDM)**

| 31 | 0 |
|---|---|
| XTDMS[*n*] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 16-39. Transmit TDM Time Slot Register (XTDM) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | XTDMS[31-0] | | Transmitter mode during TDM time slot *n*. |
| | | 0 | Transmit TDM time slot *n* is inactive. The transmit serializer does not shift out data during this slot. |
| | | 1 | Transmit TDM time slot *n* is active. The transmit serializer shifts out data during this slot according to the serializer control register (SRCTL). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.31 Transmitter Interrupt Control Register (XINTCTL)

The transmitter interrupt control register (XINTCTL) controls generation of the McASP transmit interrupt (XINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates XINT. The XINTCTL is shown in Figure 16-67 and described in Table 16-40. See Section 16.3.32 for a description of the interrupt conditions.

**Figure 16-67. Transmitter Interrupt Control Register (XINTCTL)**

| 31 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| XSTAFRM | Reserved | XDATA | XLAST | XDMAERR | XCKFAIL | XSYNCERR | XUNDRN |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-40. Transmitter Interrupt Control Register (XINTCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 7 | XSTAFRM | | Transmit start of frame interrupt enable bit. |
| | | 0 | Interrupt is disabled. A transmit start of frame interrupt does not generate a McASP transmit interrupt (XINT). |
| | | 1 | Interrupt is enabled. A transmit start of frame interrupt generates a McASP transmit interrupt (XINT). |
| 6 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 5 | XDATA | | Transmit data ready interrupt enable bit. |
| | | 0 | Interrupt is disabled. A transmit data ready interrupt does not generate a McASP transmit interrupt (XINT). |
| | | 1 | Interrupt is enabled. A transmit data ready interrupt generates a McASP transmit interrupt (XINT). |
| 4 | XLAST | | Transmit last slot interrupt enable bit. |
| | | 0 | Interrupt is disabled. A transmit last slot interrupt does not generate a McASP transmit interrupt (XINT). |
| | | 1 | Interrupt is enabled. A transmit last slot interrupt generates a McASP transmit interrupt (XINT). |
| 3 | XDMAERR | | Transmit DMA error interrupt enable bit. |
| | | 0 | Interrupt is disabled. A transmit DMA error interrupt does not generate a McASP transmit interrupt (XINT). |
| | | 1 | Interrupt is enabled. A transmit DMA error interrupt generates a McASP transmit interrupt (XINT). |
| 2 | XCKFAIL | | Transmit clock failure interrupt enable bit. |
| | | 0 | Interrupt is disabled. A transmit clock failure interrupt does not generate a McASP transmit interrupt (XINT). |
| | | 1 | Interrupt is enabled. A transmit clock failure interrupt generates a McASP transmit interrupt (XINT). |
| 1 | XSYNCERR | | Unexpected transmit frame sync interrupt enable bit. |
| | | 0 | Interrupt is disabled. An unexpected transmit frame sync interrupt does not generate a McASP transmit interrupt (XINT). |
| | | 1 | Interrupt is enabled. An unexpected transmit frame sync interrupt generates a McASP transmit interrupt (XINT). |
| 0 | XUNDRN | | Transmitter underrun interrupt enable bit. |
| | | 0 | Interrupt is disabled. A transmitter underrun interrupt does not generate a McASP transmit interrupt (XINT). |
| | | 1 | Interrupt is enabled. A transmitter underrun interrupt generates a McASP transmit interrupt (XINT). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.32 Transmitter Status Register (XSTAT)

The transmitter status register (XSTAT) provides the transmitter status and transmit TDM time slot number. If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes a new interrupt request to be generated. The XSTAT is shown in Figure 16-68 and described in Table 16-41.

**Figure 16-68. Transmitter Status Register (XSTAT)**

| 31 | | | | | | | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | XERR |
| R-0 | | | | | | | | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | | 0 |
| XDMAERR | XSTAFRM | XDATA | XLAST | XTDMSLOT | XCKFAIL | XSYNCERR | | XUNDRN |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R-0 | R/W1C-0 | R/W1C-0 | | R/W1C-0 |

LEGEND: R/W = Read/Write; R = Read only; W1C = Writing a 1 clears this bit; writing a 0 has no effect; -*n* = value after reset

**Table 16-41. Transmitter Status Register (XSTAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 8 | XERR | | XERR bit always returns a logic-OR of:XUNDRN \| XSYNCERR \| XCKFAIL \| XDMAERR<br>Allows a single bit to be checked to determine if a transmitter error interrupt has occurred. |
| | | 0 | No errors have occurred. |
| | | 1 | An error has occurred. |
| 7 | XDMAERR | | Transmit DMA error flag. XDMAERR is set when the CPU or DMA writes more serializers through the data port in a given time slot than were programmed as transmitters. Causes a transmit interrupt (XINT), if this bit is set and XDMAERR in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. |
| | | 0 | Transmit DMA error did not occur. |
| | | 1 | Transmit DMA error did occur. |
| 6 | XSTAFRM | | Transmit start of frame flag. Causes a transmit interrupt (XINT), if this bit is set and XSTAFRM in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. |
| | | 0 | No new transmit frame sync (AFSX) is detected. |
| | | 1 | A new transmit frame sync (AFSX) is detected. |
| 5 | XDATA | | Transmit data ready flag. Causes a transmit interrupt (XINT), if this bit is set and XDATA in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. |
| | | 0 | XBUF is written and is full. |
| | | 1 | Data is copied from XBUF to XRSR. XBUF is empty and ready to be written. XDATA is also set when the transmit serializers are taken out of reset. When XDATA is set, it always causes a DMA event (AXEVT). |
| 4 | XLAST | | Transmit last slot flag. XLAST is set along with XDATA, if the current slot is the last slot in a frame. Causes a transmit interrupt (XINT), if this bit is set and XLAST in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. |
| | | 0 | Current slot is not the last slot in a frame. |
| | | 1 | Current slot is the last slot in a frame. XDATA is also set. |
| 3 | XTDMSLOT | | Returns the LSB of XSLOT. Allows a single read of XSTAT to determine whether the current TDM time slot is even or odd. |
| | | 0 | Current TDM time slot is odd. |
| | | 1 | Current TDM time slot is even. |
| 2 | XCKFAIL | | Transmit clock failure flag. XCKFAIL is set when the transmit clock failure detection circuit reports an error (see *Clock Failure Detection*). Causes a transmit interrupt (XINT), if this bit is set and XCKFAIL in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. |
| | | 0 | Transmit clock failure did not occur. |
| | | 1 | Transmit clock failure did occur. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 16-41. Transmitter Status Register (XSTAT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | XSYNCERR | | Unexpected transmit frame sync flag. XSYNCERR is set when a new transmit frame sync (AFSX) occurs before it is expected. Causes a transmit interrupt (XINT), if this bit is set and XSYNCERR in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. |
| | | 0 | Unexpected transmit frame sync did not occur. |
| | | 1 | Unexpected transmit frame sync did occur. |
| 0 | XUNDRN | | Transmitter underrun flag. XUNDRN is set when the transmit serializer is instructed to transfer data from XBUF to XRSR, but XBUF has not yet been serviced with new data since the last transfer. Causes a transmit interrupt (XINT), if this bit is set and XUNDRN in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect. |
| | | 0 | Transmitter underrun did not occur. |
| | | 1 | Transmitter underrun did occur. For details on McASP action upon underrun conditions, see *Buffer Underrun Error - Transmitter*. |

### 16.3.33 Current Transmit TDM Time Slot Register (XSLOT)

The current transmit TDM time slot register (XSLOT) indicates the current time slot for the transmit data frame. The XSLOT is shown in Figure 16-69 and described in Table 16-42.

**Figure 16-69. Current Transmit TDM Time Slot Register (XSLOT)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 0 |
|---|---|---|---|
| Reserved | | XSLOTCNT | |
| R-0 | | R-17Fh | |

LEGEND: R = Read only; -*n* = value after reset

**Table 16-42. Current Transmit TDM Time Slot Register (XSLOT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 9-0 | XSLOTCNT | 0-17Fh | Current transmit time slot count. Legal values: 0 to 383 (17Fh).<br>During reset, this counter value is 383 so the next count value, which is used to encode the first DIT group of data, will be 0 and encodes the B preamble.<br>TDM function is not supported for >32 time slots. However, TDM time slot counter may count to 383 when used to transmit a DIT block. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.34 Transmit Clock Check Control Register (XCLKCHK)

The transmit clock check control register (XCLKCHK) configures the transmit clock failure detection circuit. The XCLKCHK is shown in Figure 16-70 and described in Table 16-43.

**Figure 16-70. Transmit Clock Check Control Register (XCLKCHK)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| XCNT | | XMAX | |
| R-0 | | R/W-0 | |

| 15 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| XMIN | | Reserved | | XPS | |
| R/W-0 | | R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-43. Transmit Clock Check Control Register (XCLKCHK) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | XCNT | 0 | Transmit clock count value (from previous measurement). The clock circuit continually counts the number of system clocks for every 32 transmit high-frequency master clock (AHCLKX) signals, and stores the count in XCNT until the next measurement is taken. |
| 23-16 | XMAX | 0-FFh | Transmit clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If the current counter value is greater than XMAX after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic. |
| 15-8 | XMIN | 0-FFh | Transmit clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If XCNT is less than XMIN after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic. |
| 7-4 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 3-0 | XPS | 0-Fh | Transmit clock check prescaler value. |
| | | 0 | McASP system clock divided by 1. |
| | | 1h | McASP system clock divided by 2. |
| | | 2h | McASP system clock divided by 4. |
| | | 3h | McASP system clock divided by 8. |
| | | 4h | McASP system clock divided by 16. |
| | | 5h | McASP system clock divided by 32. |
| | | 6h | McASP system clock divided by 64. |
| | | 7h | McASP system clock divided by 128. |
| | | 8h | McASP system clock divided by 256. |
| | | 9h-Fh | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

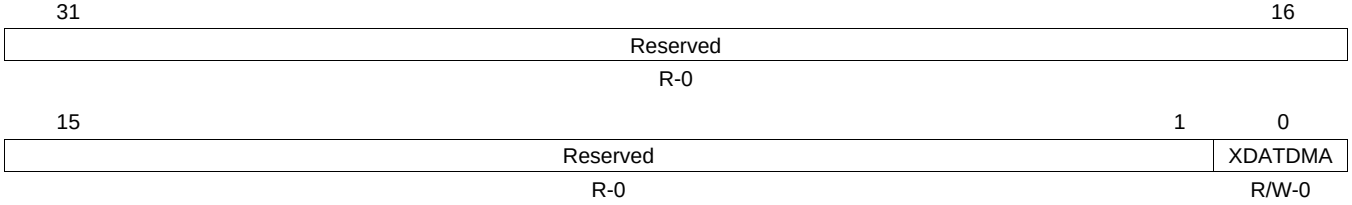
### 16.3.35 Transmitter DMA Event Control Register (XEVTCTL)

The transmitter DMA event control register (XEVTCTL) contains a disable bit for the transmit DMA event. The XEVTCTL is shown in Figure 16-71 and described in Table 16-44.

**NOTE: Device-specific registers**

Accessing REVTCTL not implemented on a specific device may cause improper device operation.

**Figure 16-71. Transmitter DMA Event Control Register (XEVTCTL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-44. Transmitter DMA Event Control Register (XEVTCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 0 | XDATDMA | | Transmit data DMA request enable bit. If writing to this bit, always write the default value of 0. |
| | | 0 | Transmit data DMA request is enabled. |
| | | 1 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.36 Serializer Control Registers (SRCTL*n*)

Each serializer on the McASP has a serializer control register (SRCTL). There are up to 16 serializers per McASP. The SRCTL is shown in Figure 16-72 and described in Table 16-45.

**NOTE: Device-specific registers**

Accessing SRCTL*n* not implemented on a specific device may cause improper device operation.

**Figure 16-72. Serializer Control Registers (SRCTL*n*)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | RRDY | XRDY | DISMOD | | SRMOD | |
| R-0 | | R-0 | R-0 | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-45. Serializer Control Registers (SRCTL*n*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility. |
| 5 | RRDY | | Receive buffer ready bit. RRDY indicates the current receive buffer state. Always reads 0 when programmed as a transmitter or as inactive. If SRMOD bit is set to receive (2h), RRDY switches from 0 to 1 whenever data is transferred from XRSR to RBUF. |
| | | 0 | Receive buffer (RBUF) is empty. |
| | | 1 | Receive buffer (RBUF) contains data and needs to be read before the start of the next time slot or a receiver overrun occurs. |
| 4 | XRDY | | Transmit buffer ready bit. XRDY indicates the current transmit buffer state. Always reads 0 when programmed as a receiver or as inactive. If SRMOD bit is set to transmit (1h), XRDY switches from 0 to 1 when XSRCLR in GBLCTL is switched from 0 to 1 to indicate an empty transmitter. XRDY remains set until XSRCLR is forced to 0, data is written to the corresponding transmit buffer, or SRMOD bit is changed to receive (2h) or inactive (0). |
| | | 0 | Transmit buffer (XBUF) contains data. |
| | | 1 | Transmit buffer (XBUF) is empty and needs to be written before the start of the next time slot or a transmit underrun occurs. |
| 3-2 | DISMOD | 0-3h | Serializer pin drive mode bit. Drive on pin when in inactive TDM slot of transmit mode or when serializer is inactive. This field only applies if the pin is configured as a McASP pin (PFUNC = 0). |
| | | 0 | Drive on pin is 3-state. |
| | | 1h | Reserved. |
| | | 2h | Drive on pin is logic low. |
| | | 3h | Drive on pin is logic high. |
| 1-0 | SRMOD | 0-3h | Serializer mode bit. |
| | | 0 | Serializer is inactive. |
| | | 1h | Serializer is transmitter. |
| | | 2h | Serializer is receiver. |
| | | 3h | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.37 DIT Left Channel Status Registers (DITCSRA0-DITCSRA5)

The DIT left channel status registers (DITCSRA) provide the status of each left channel (even TDM time slot). Each of the six 32-bit registers (Figure 16-73) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register file in time, if a different set of data need to be sent.

**Figure 16-73. DIT Left Channel Status Registers (DITCSRA0-DITCSRA5)**

| 31 | 0 |
|---|---|
| DITCSRA[*n*] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

### 16.3.38 DIT Right Channel Status Registers (DITCSRB0-DITCSRB5)

The DIT right channel status registers (DITCSRB) provide the status of each right channel (odd TDM time slot). Each of the six 32-bit registers (Figure 16-74) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register file in time, if a different set of data need to be sent.

**Figure 16-74. DIT Right Channel Status Registers (DITCSRB0-DITCSRB5)**

| 31 | 0 |
|---|---|
| DITCSRB[*n*] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

### 16.3.39 DIT Left Channel User Data Registers (DITUDRA0-DITUDRA5)

The DIT left channel user data registers (DITUDRA) provides the user data of each left channel (even TDM time slot). Each of the six 32-bit registers (Figure 16-75) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register in time, if a different set of data need to be sent.

**Figure 16-75. DIT Left Channel User Data Registers (DITUDRA0-DITUDRA5)**

| 31 | 0 |
|---|---|
| DITUDRA[*n*] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

### 16.3.40 DIT Right Channel User Data Registers (DITUDRB0-DITUDRB5)

The DIT right channel user data registers (DITUDRB) provides the user data of each right channel (odd TDM time slot). Each of the six 32-bit registers (Figure 16-76) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register in time, if a different set of data need to be sent.

**Figure 16-76. DIT Right Channel User Data Registers (DITUDRB0-DITUDRB5)**

| 31 | 0 |
|---|---|
| DITUDRB[*n*] | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

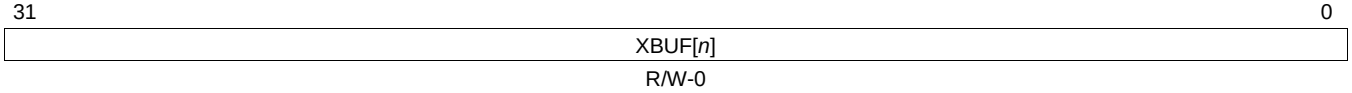
### 16.3.41 Transmit Buffer Registers (XBUF*n*)

The transmit buffers for the serializers (XBUF) hold data from the transmit format unit. For transmit operations, the XBUF (Figure 16-77) is an alias of the XRBUF in the serializer.

**NOTE: Device-specific registers**

Accessing XBUF registers not implemented on a specific device may cause improper device operation.

**Figure 16-77. Transmit Buffer Registers (XBUFn)**




LEGEND: R/W = Read/Write; -*n* = value after reset

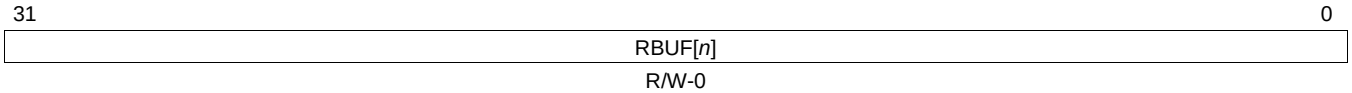
### 16.3.42 Receive Buffer Registers (RBUF*n*)

The receive buffers for the serializers (RBUF) hold data from the serializer before the data goes to the receive format unit. For receive operations, the RBUF (Figure 16-78) is an alias of the XRBUF in the serializer.

**NOTE: Device-specific registers**

Accessing RBUF registers not implemented on a specific device may cause improper device operation.

**Figure 16-78. Receive Buffer Registers (RBUF*n*)**




LEGEND: R/W = Read/Write; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

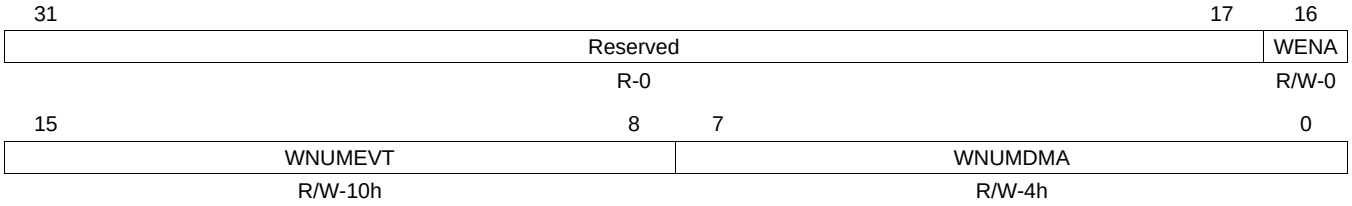
### 16.3.43 Write FIFO Control Register (WFIFOCTL)

Note that this register is only available if the Write FIFO is configured. The Write FIFO control register (WFIFOCTL) is shown in Figure 16-79 and described in Table 16-46.

**NOTE:** The WNUMEVT and WNUMDMA values must be set prior to enabling the Write FIFO.

If the Write FIFO is to be enabled, it must be enabled prior to taking the McASP out of reset.

**Figure 16-79. Write FIFO Control Register (WFIFOCTL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-46. Write FIFO Control Register (WFIFOCTL) Field Descriptions**

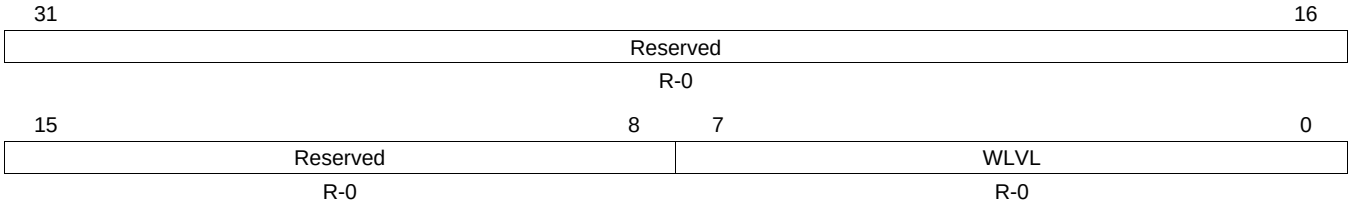
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved |
| 16 | WENA | | Write FIFO enable bit. |
| | | 0 | Write FIFO is disabled. The WLVL bit in the Write FIFO status register (WFIFOSTS) is reset to 0 and pointers are initialized, that is, the Write FIFO is "flushed." |
| | | 1 | Write FIFO is enabled. If Write FIFO is to be enabled, it must be enabled prior to taking McASP out of reset. |
| 15-8 | WNUMEVT | 0-FFh | Write word count per DMA event (32-bit). When the Write FIFO has space for at least WNUMEVT words of data, then an AXEVT (transmit DMA event) is generated to the host/DMA controller. This value should be set to a non-zero integer multiple of the number of serializers enabled as transmitters. This value must be set prior to enabling the Write FIFO. |
| | | 0 | 0 words |
| | | 1h | 1 word |
| | | 2h | 2 words |
| | | 3h-40h | 3 to 64 words |
| | | 41h-FFh | Reserved |
| 7-0 | WNUMDMA | 0-FFh | Write word count per transfer (32-bit words). Upon a transmit DMA event from the McASP, WNUMDMA words are transferred from the Write FIFO to the McASP. This value must equal the number of McASP serializers used as transmitters. This value must be set prior to enabling the Write FIFO. |
| | | 0 | 0 words |
| | | 1h | 1 word |
| | | 2h | 2 words |
| | | 3h-10h | 3-16 words |
| | | 11h-FFh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.44 Write FIFO Status Register (WFIFOSTS)

Note that this register is only available if the Write FIFO is configured. The Write FIFO status register (WFIFOSTS) is shown in Figure 16-80 and described in Table 16-47.

**Figure 16-80. Write FIFO Status Register (WFIFOSTS)**




LEGEND: R = Read only; -*n* = value after reset

**Table 16-47. Write FIFO Status Register (WFIFOSTS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | WLVL | 0-FFh | Write level (read-only). Number of 32-bit words currently in the Write FIFO. |
| | | 0 | 0 words currently in Write FIFO. |
| | | 1h | 1 word currently in Write FIFO. |
| | | 2h | 2 words currently in Write FIFO. |
| | | 3h-40h | 3 to 64 words currently in Write FIFO. |
| | | 41h-FFh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

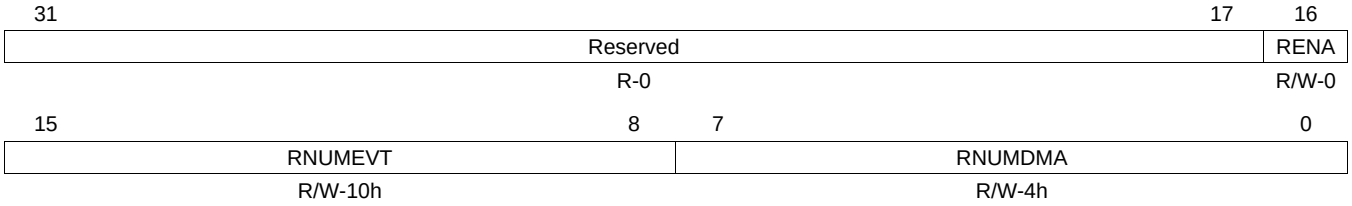
### 16.3.45 Read FIFO Control Register (RFIFOCTL)

Note that this register is only available if the Read FIFO is configured. The Read FIFO control register (RFIFOCTL) is shown in Figure 16-81 and described in Table 16-48.

**NOTE:** The RNUMEVT and RNUMDMA values must be set prior to enabling the Read FIFO.

If the Read FIFO is to be enabled, it must be enabled prior to taking the McASP out of reset.

**Figure 16-81. Read FIFO Control Register (RFIFOCTL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 16-48. Read FIFO Control Register (RFIFOCTL) Field Descriptions**

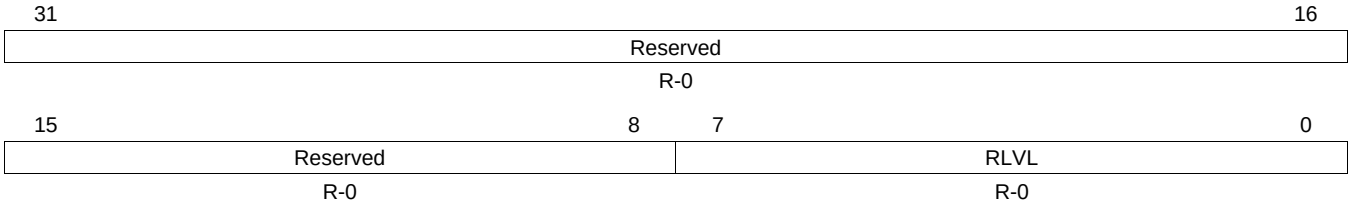
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved |
| 16 | RENA | | Read FIFO enable bit. |
| | | 0 | Read FIFO is disabled. The RLVL bit in the Read FIFO status register (RFIFOSTS) is reset to 0 and pointers are initialized, that is, the Read FIFO is “flushed.” |
| | | 1 | Read FIFO is enabled. If Read FIFO is to be enabled, it must be enabled prior to taking McASP out of reset. |
| 15-8 | RNUMEVT | 0-FFh | Read word count per DMA event (32-bit). When the Read FIFO contains at least RNUMEVT words of data, then an AREVT (receive DMA event) is generated to the host/DMA controller. This value should be set to a non-zero integer multiple of the number of serializers enabled as receivers. This value must be set prior to enabling the Read FIFO. |
| | | 0 | 0 words |
| | | 1h | 1 word |
| | | 2h | 2 words |
| | | 3h-40h | 3 to 64 words |
| | | 41h-FFh | Reserved |
| 7-0 | RNUMDMA | 0-FFh | Read word count per transfer (32-bit words). Upon a receive DMA event from the McASP, the Read FIFO reads RNUMDMA words from the McASP. This value must equal the number of McASP serializers used as receivers. This value must be set prior to enabling the Read FIFO. |
| | | 0 | 0 words |
| | | 1 | 1 word |
| | | 2 | 2 words |
| | | 3h-10h | 3-16 words |
| | | 11h-FFh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 16.3.46 Read FIFO Status Register (RFIFOSTS)

Note that this register is only available if the Read FIFO is configured. The Read FIFO status register (RFIFOSTS) is shown in Figure 16-82 and described in Table 16-49.

**Figure 16-82. Read FIFO Status Register (RFIFOSTS)**




LEGEND: R = Read only; -*n* = value after reset

**Table 16-49. Read FIFO Status Register (RFIFOSTS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | RLVL | 0-FFh | Read level (read-only). Number of 32-bit words currently in the Read FIFO. |
| | | 0 | 0 words currently in Read FIFO. |
| | | 1h | 1 word currently in Read FIFO. |
| | | 2h | 2 words currently in Read FIFO. |
| | | 3h-40h | 3 to 64 words currently in Read FIFO. |
| | | 41h-FFh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

# Multichannel Buffered Serial Port (McBSP)

This chapter describes the multichannel buffered serial port (McBSP).

**Topic** **Page**

17.1 Introduction ..... 3058
17.2 Architecture ..... 3060
17.3 McBSP Registers ..... 3110


Copyright © 2011–2016, Texas Instruments Incorporated

## 17.1 Introduction

### 17.1.1 Overview

The multichannel buffered serial port (McBSP) provides a full duplex direct serial interface between the host chip and other devices in a system such as application chips, codecs. It can accommodate to a wide range of peripherals and protocols thanks to its high level of versatility.

### 17.1.2 Features

The main features of the McBSP modules are:

- OCP-IP2.0 compliant interface (32-bit data bus)
- 512-byte internal buffer for transmit operation (fixed size)
- 512-byte internal buffer for receive operation (fixed size)
- Auto-gating clocks for power saving
- Power Management interface
- Full-duplex communication
- Buffered transmission and reception that allows a continuous data stream
- Independent clocking and framing for reception and for transmission
- Capability to generate interrupts or DMA requests on internal events
- 128 channels for transmission and for reception
- Multi-channel selection modes that allow to enable or block transfers in each one of the channels
- Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected A/D and D/A devices (I2S compliant devices)
- Support for external generation of clock signals and frame-synchronization (frame-synchronization) signals
- A programmable sample rate generator for internal generation and control of clock signals and frame-synchronization signals
- Programmable polarity for frame-synchronization pulses and for clock signals
- A wide selection of data sizes: 8, 12, 16, 20, 24, and 32 bits
- Bit reordering (send/receive LSB)
- Status bits for flagging exception/error conditions
- Full/Half cycle mode on both transmit and receive side

The McBSP supports dual phase frames in order to provide I2S fully compliant capabilities. The limitation on dual phase frame is that the number of words per phase must be set to 1 for both first and second phase.

Four operating modes are defined for the module:

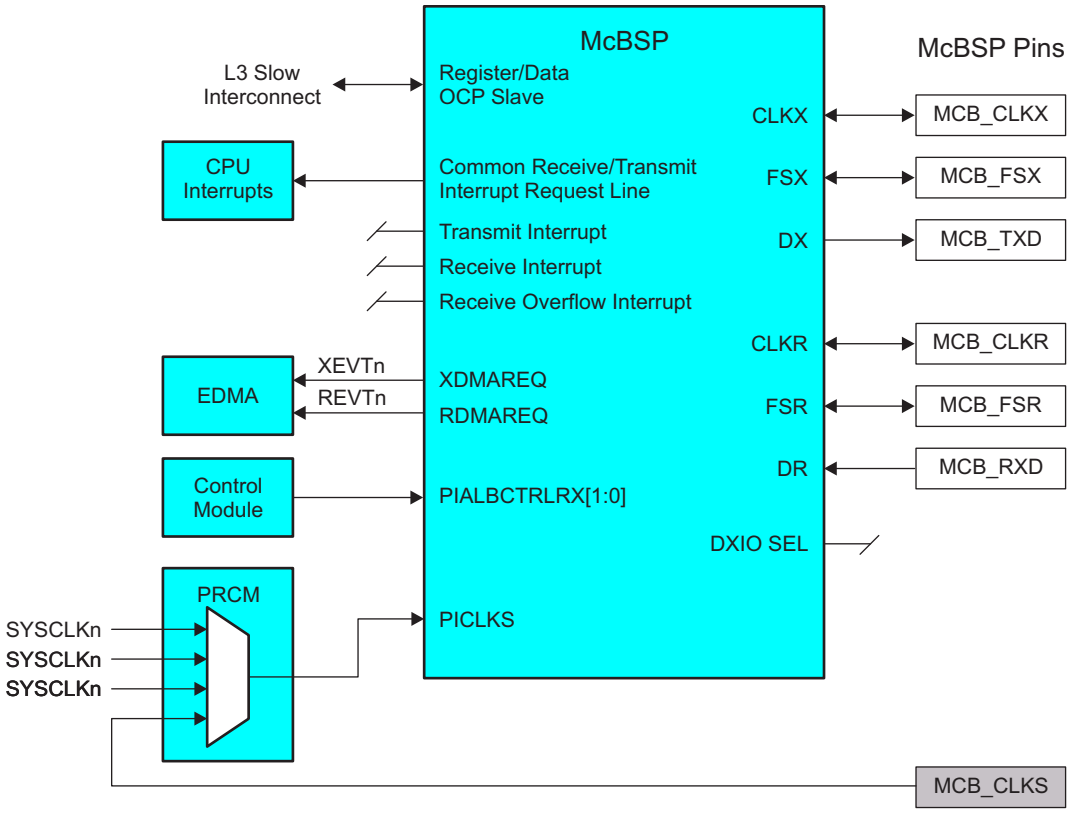
- **Active mode:** the module is running synchronously on the interface clock, interrupts and DMA requests can be generated according to the configuration and the external signals.
- **Smart Idle mode:** the module is in a waiting state, interface/functional clocks can be stopped, no interrupt can be generated, a wake-up signal can be generated according the configuration and external signals.
- **Force Idle mode:** the module has no activity, interface clock can be stopped, no interrupts and DMA requests can be generated, and the wake-up feature is inhibited. The software must disable the McBSP module (receive/transmit parts) prior to enter Force Idle mode.
- **No Idle mode:** the module is active but cannot respond to host interface requests.

The Smart/Force Idle modes are configured within the module and activated on request by the host processor through system interface sideband signals.


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.1.3 Block Diagram

**Figure 17-1. McBSP Block Diagram**




**Note:** The number of SYSCLKn varies per device implementation.
See Control Module Chapter for details.

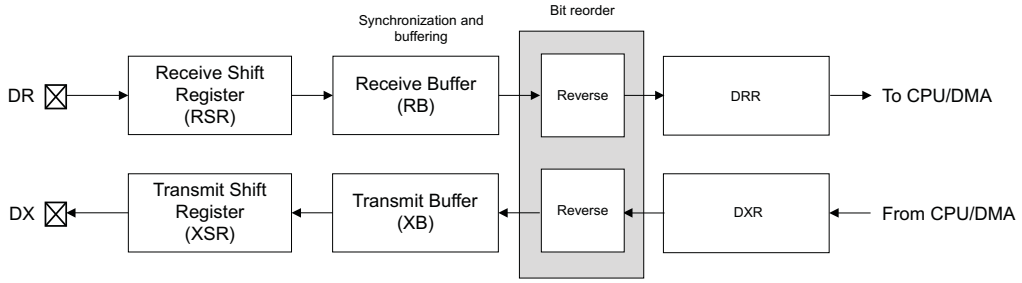

Copyright © 2011–2016, Texas Instruments Incorporated

## 17.2 Architecture

### 17.2.1 McBSP Data Transfer Process

Figure 17-2 shows the McBSP data transfer paths. McBSP receive operations and transmit operations are buffered (up to 512-byte buffers organized on 32-bit words are used). Input/output registers are 32-bit long words.

**Figure 17-2. McBSP Data Transfer Paths**




#### 17.2.1.1 Data Transfer Process for 8/12/16/20/24/32-bit Long Words

Receive data arrives on the McBSP.DR pin and is shifted into receive shift register. When a full word is received, the content of the shift register is copied into receive buffer under condition receive buffer is not full. When the receive buffer threshold is reached, the McBSP asserts DMA or interrupt request and the receive buffer content is then transferred to the host (the CPU or the DMA controller reads the data receive register DRR_REG).

Transmit data is written by the CPU or the DMA controller to data transmit register DXR_REG using the byte enable input (when a byte is not enabled the byte value in the memory will contain the previous written value). If there is no previous data in transmit shift register, the value from the transmit buffer is copied to transmit shift register; otherwise, the content is copied to transmit shift register when the last bit of the previous data is shifted out on the McBSP.DX pin.

Note that the byte enable is used for transmit data register DXR_REG only and the value is ignored when the CPU accesses are targeting other registers. When writing to DXR_REG, the bytes are independently enabled by the byte enable pattern (for example, byte enable is “0001” then only the less significant byte will be written).

#### 17.2.1.2 Bit Reordering (option to transfer LSB first)

Generally, the McBSP transmits or receives all data with the most significant bit (MSB) first. However, certain data protocols require the least significant bit (LSB) to be transferred first.

If you set XREVERSE = 01b in XCR2_REG[4:3], the bit ordering of the data words is reversed (LSB first) before being sent to the serial port.

If you set RREVERSE = 01b in RCR2_REG[4:3], the bit ordering of the data words is reversed during reception.

This feature is available for all the data formats from 8 up to 32-bit data length.

#### 17.2.1.3 Clocking and Framing Data

This section explains basic concepts and terminology important for understanding how McBSP data transfers are timed and delimited.


Copyright © 2011–2016, Texas Instruments Incorporated

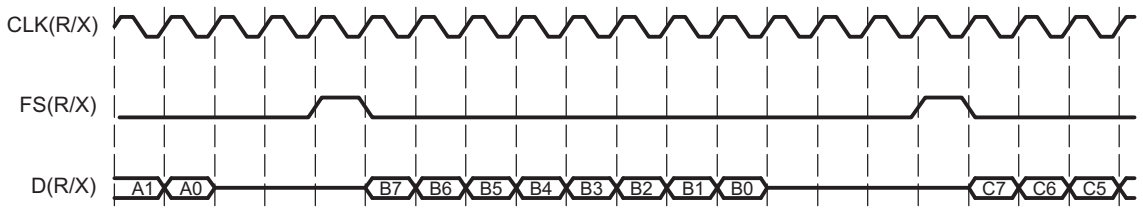
#### 17.2.1.3.1 Clocking

Data is shifted one bit at a time from the McBSP.DR pin to the receive shift registers or from the transmit shift registers to the McBSP.DX pin. The time for each bit transfer is controlled by the rising or falling edge of a clock signal.

The receive clock signal (CLKR) controls bit transfers from the McBSP.DR pin to the receive shift registers. The transmit clock signal (CLKX) controls bit transfers from the transmit shift registers to the McBSP.DX pin. CLKR or CLKX can be derived from a pin at the boundary of the McBSP (McBSP.CLKR and McBSP.CLKX respectively) or derived from inside the McBSP. The polarities of CLKR and CLKX are programmable.

Figure 17-3 gives an example in which the clock signal controls the timing of each bit transfer on the pin.

**Figure 17-3. Clock Signal Control of Bit Transfer Timing**




The McBSP is constrained to operate at an internal functional frequency up to 55 MHz while CPU interface (OCP compliant interface) can operate at a frequency up to 110 MHz. When driving CLKX or CLKR at the pin, choose an appropriate input clock frequency. When using the internal sample rate generator for CLKX/CLKR/CLKS, choose an appropriate input clock frequency (up to 110 MHz) and divide down value (CLKGDV). Please see the device datasheet for maximum McBSP CLKX cycle time supported on your device.

#### 17.2.1.3.2 Serial Words

Bits traveling between a shift register (RSR or XSR) and a data pin (McBSP.DR or McBSP.DX) are transferred in a group called a serial word. You can define how many bits are in a word.

Bits coming in on the McBSP.DR pin are held in RSR until RSR holds a full serial word. Only then is the word passed to the receive buffer RB (and ultimately to the DRR_REG register).

During transmission, XSR accept new data from transmit buffer XB after a full serial word has been passed from XSR to the McBSP.DX pin.

In the example in Figure 17-3, an 8–bit word size was defined (see bits 7 through 0 of word B being transferred).

#### 17.2.1.3.3 Frames and Frame Synchronization

One or more words are transferred in a group called a frame. You can define how many words are in a frame. All of the words in a frame are sent in a continuous stream. However, there can be pauses between frame transfers. The McBSP uses frame–synchronization signals to determine when each frame is received/transmitted. When a pulse occurs on a frame–synchronization signal, the McBSP begins receiving/transmitting a frame of data. When the next pulse occurs, the McBSP receives/transmits the next frame, and so on. Pulses on the receive frame–synchronization (FSR) signal initiate frame transfers on McBSP.DR. Pulses on the transmit frame–synchronization (FSX) signal initiate frame transfers on McBSP.DX. FSR or FSX can be derived from a pin at the boundary of the McBSP (McBSP.FSR and McBSP.FSX respectively) or derived from inside the McBSP. In the example in Figure 17-3, a one–word frame is transferred when a frame–synchronization pulse occurs. In McBSP operation, the inactive–to–active transition of the frame–synchronization signal indicates the start of the next frame. For this reason, the frame–synchronization signal may be high for an arbitrary number of clock cycles. Only after the signal is recognized to have gone inactive, and then active again, does the next frame synchronization occurs.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.1.3.4 Detecting Frame–Synchronization Pulses, Even in Reset State

The McBSP can generate receive and transmit interrupts to the CPU to indicate specific events in the McBSP. To facilitate detection of frame synchronization, these interrupts can be sent in response to frame–synchronization pulses. Unlike other serial port interrupt modes, this mode can operate while the associated portion of the serial port is in reset (such as activating RINT when the receiver is in reset). In this case, FSRM/FSXM and FSRP/FSXP still select the appropriate source and polarity of frame synchronization. Thus, even when the serial port is in the reset state, these signals are synchronized to the CPU clock and then sent to the CPU in the form of RINT and XINT at the point at which they feed the receiver and transmitter of the serial port. Consequently, a new frame–synchronization pulse can be detected, and after this occurs the CPU can take the serial port out of reset safely.

#### 17.2.1.3.5 Ignoring Frame–Synchronization Pulses

The McBSP ignores transmit and/or receive frame–synchronization pulses while the frame transfer has been already started by a previous frame–synchronization pulse (unexpected frame–synchronization pulses). The McBSP does not support features like re-transmit or re-receive of an erroneous frame or word. The receiver or transmitter ignores frame–synchronization pulses until the desired frame length or number of words is reached. For more details on unexpected frame–synchronization pulses, see Section 17.2.3.2 and Section 17.2.3.5.

#### 17.2.1.3.6 Frame Frequency

The frame frequency is determined by the period between frame–synchronization pulses and is defined as shown in the following equation:

Frame frequency = Clock frequency / (Number of clock cycles between frame – synchronization pulses)

The frame frequency can be increased by decreasing the time between frame–synchronization pulses (limited only by the number of bits per frame). As the frame transmit frequency increases, the inactivity period between the data packets for adjacent transfers decreases to zero.

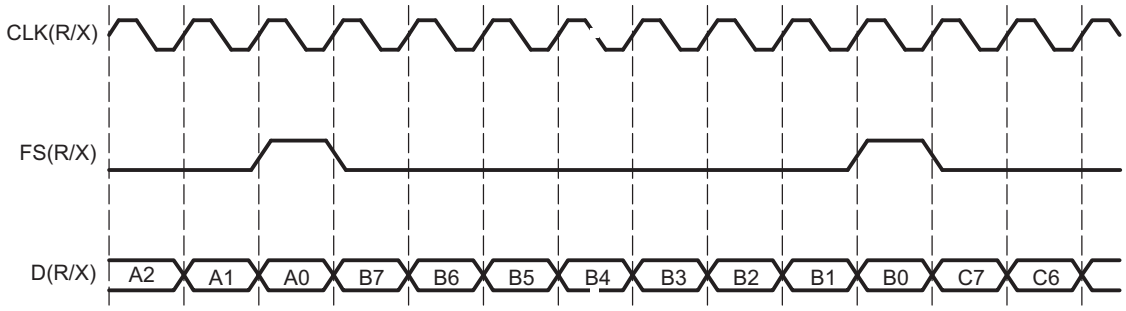
#### 17.2.1.3.7 Maximum Frame Frequency

The minimum number of clock cycles between frame synchronization pulses is equal to the number of bits transferred per frame. The maximum frame frequency is defined as shown in the following equation:

Maximum frame frequency = Clock frequency / Number of bits per frame

Figure 17-4 shows the McBSP operating at maximum packet frequency. At maximum packet frequency, the data bits in consecutive packets are transmitted contiguously with no inactivity between bits.

**Figure 17-4. McBSP Operating at Maximum Packet Frequency**




If there is a 1–bit data delay as shown in this figure, the frame–synchronization pulse overlaps the last bit transmitted in the previous frame. Effectively, this permits a continuous stream of data, back-to-back transfers.

For XDATDLY = 0 (0-bit data delay), the first bit of data is transmitted asynchronously to the internal transmit clock signal (CLKX).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.1.4 Frame Phases (dual phase frame I2S support)

The McBSP allows you to configure each frame to contain one or two phases. The support for dual phase frames is required in order to provide I2S fully compliant capabilities. The limitation on dual phase frame support is that the number of words per phase must be set to 1 for both first and second phase.

The number of bits per word can be specified differently for each of the two phases of a frame, allowing greater flexibility in structuring data transfers. For example, a user might define a frame as consisting of one phase containing one word of 16 bits, followed by a second phase consisting of one word of 32 bits. This configuration allows the user to compose frames for custom applications such as I2S protocol.

##### *17.2.1.4.1 Number of Phases, Words, and Bits per Frame*

Table 17-1 shows which bit fields in the receive control registers (RCR1_REG and RCR2_REG) and in the transmit control registers (XCR1_REG and XCR2_REG) determine the number of phases per frame, the number of words per frame, and the number of bits per word for each phase, for both receiver and transmitter. The maximum number of words per frame is limited to 2 when using dual-phase frames (one word for each phase), and to 128 for a single–phase frame. The number of bits per word can be 8, 12, 16, 20, 24, or 32 bits.

The following legend applies to the table:

- RPHASE = RCR2_REG[15]
- XPHASE = XCR2_REG[15]
- RFRLEN1 = RCR1_REG[14:8] (cleared to 0 if using dual-phase)
- RFRLEN2 = RCR2_REG[14:8] (cleared to 0 if using dual-phase)
- XFRLEN1 = XCR1_REG[14:8] (cleared to 0 if using dual-phase)
- XFRLEN2 = XCR2_REG[14:8] (cleared to 0 if using dual-phase)
- RWDLEN1 = RCR1_REG[7:5]
- RWDLEN2 = RCR2_REG[7:5]
- XWDLEN1 = XCR1_REG[7:5]
- XWDLEN2 = XCR2_REG[7:5]

**Table 17-1. Phases, Words and Bits Per Frame Control Bits**

| Operation | Number of Phases |
|---|---|
| Reception | 1 (RPHASE = 0) |
| Reception | 2 (RPHASE = 1) |
| Transmission | 1 (XPHASE = 0) |
| Transmission | 2 (XPHASE = 1) |

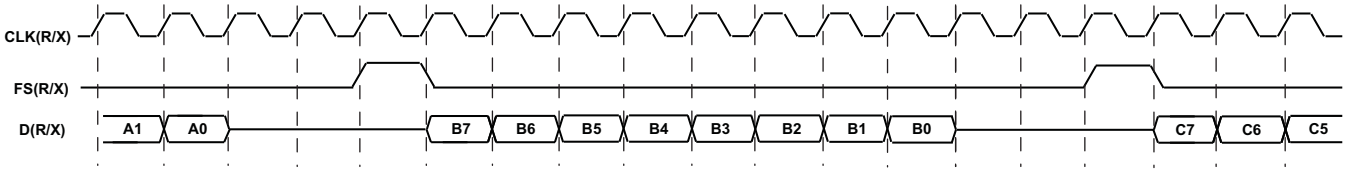

Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.1.4.2 Single-phase Frame Example

Figure 17-5 shows an example of a single-phase data frame containing one 8-bit word. Because the transfer is configured for one data bit delay, the data on the McBSP.DX and McBSP.DR pins are available one clock cycle after FS(R/X) goes active. The figure makes the following assumptions:

- RPHASE (RCR2_REG[15]) and XPHASE (XCR2_REG[15]) = 0: Single-phase frame
- RFRLEN1 (RCR1_REG[14:8]) and XFRLEN1 (XCR1_REG[14:8]) = 0: 1 word per frame
- RWDLEN1 (RCR1_REG[7:5]) and XWDLEN1 (XCR1_REG[7:5]) = 0: 8-bit word length
- RFRLEN2 (RCR2_REG[14:8]) and XWDLEN2 (XCR2_REG[14:8]) are ignored
- CLKRP (PCR_REG[0]) and CLKXP (PCR_REG[1]) = 0: Receive data clocked on falling edge; transmit data clocked on rising edge
- FSRP (PCR_REG[2]) and FSXP (PCR_REG[3]) = 0: Active–high frame–synchronization signals
- RDATDLY (RCR2_REG[1:0]) and XDARDLY (XCR2_REG[1:0]) = 01b: 1–bit data delay

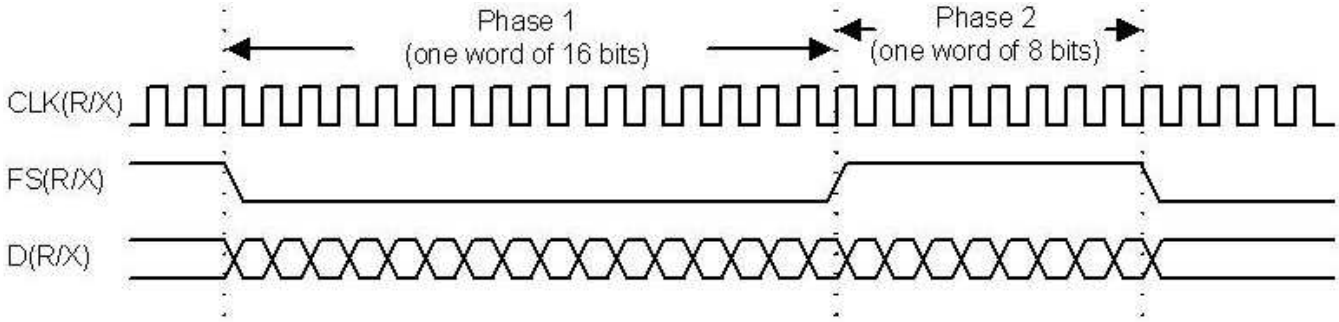
**Figure 17-5. Single-Phase Frame for a McBSP Data Transfer**




#### 17.2.1.4.3 Dual-Phase Frame Example

Figure 17-6 shows an example of a frame where the first phase consists of one word of 16 bits, followed by a second phase of one word of 8 bits. The entire bit stream in the frame is contiguous. There are no gaps between words/phases.

**Figure 17-6. Dual-Phase Frame for a McBSP Data Transfer**





Copyright © 2011–2016, Texas Instruments Incorporated

### 17.2.1.5 McBSP Reception

This section explains the fundamental process of reception in the McBSP.

Figure 17-7 and Figure 17-8 show how reception occurs in the McBSP. Figure 17-7 shows the physical path for the data. Figure 17-8 is a timing diagram showing signal activity for one possible reception scenario. A description of the process follows the figures.

**Figure 17-7. McBSP Reception Physical Data Path**

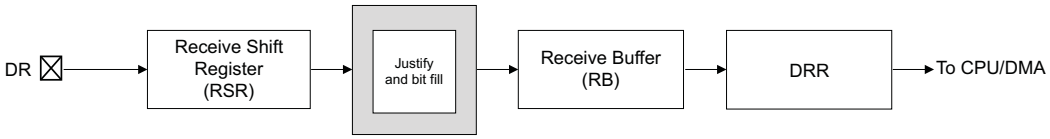
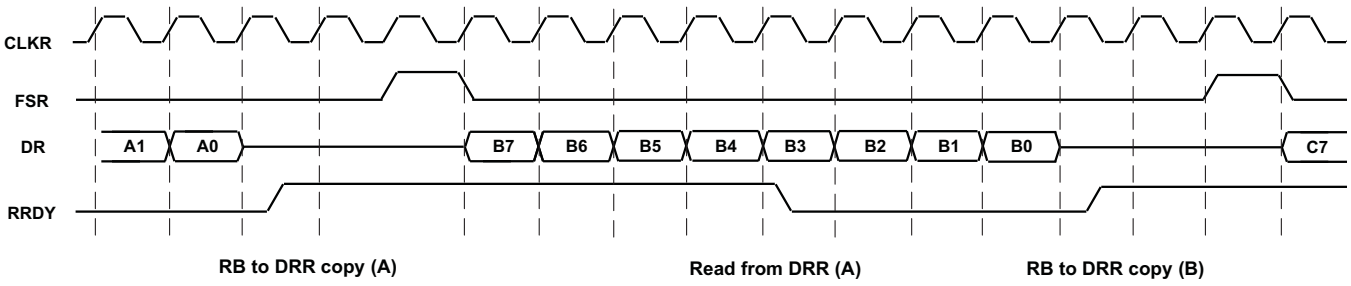



**Figure 17-8. McBSP Reception Signal Activity**




**RRDY: Status of receiver ready bit (high is 1)**

The following process describes how data travels from the McBSP.DR pin to the CPU or to the DMA controller:

1. The McBSP waits for a receive frame-synchronization pulse on internal FSR.
2. When the pulse arrives, the McBSP inserts the appropriate data delay that is selected with the RDATDLY bits of RCR2_REG[1:0] register. In the preceding timing diagram a 1-bit data delay is selected.
3. The McBSP accepts data bits on the McBSP.DR pin and shifts them into the receive shift register.
4. When a full word is received, the McBSP copies the contents of the receive shift register to the receive buffer, provided that RB is not full.
5. When the programmed receive threshold is reached the McBSP asserts the receiver ready bit (RRDY) in SPCR1_REG. This indicates that receive data is ready to be read by the CPU or the DMA controller by accessing DRR_REG register. The data copied from RB to DRR_REG is justified and bit filled according to the RJUST bits.
6. The CPU or the DMA controller reads the data from the data receive register. When the receive buffer is empty, RRDY bit is cleared.

When activity is not properly timed, errors can occur. See the following topics for more details:

- Overrun in the Receiver
- Underflow in the receiver
- Unexpected Receive Frame–Synchronization Pulse


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.2.1.6 McBSP Transmission

This section explains the fundamental process of transmission in the McBSP. Figure 17-9 and Figure 17-10 show how transmission occurs in the McBSP.

Figure 17-9 shows the physical path for the data. Figure 17-10 is a timing diagram showing signal activity for one possible transmission scenario. A description of the process follows the figures.

**Figure 17-9. McBSP Transmission Physical Data Path**

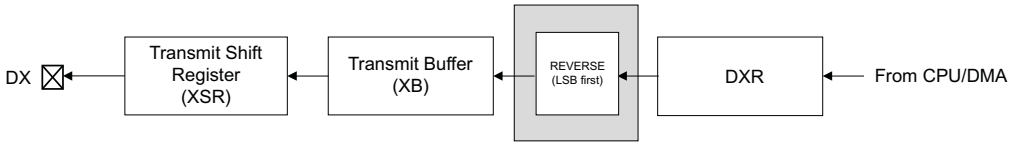
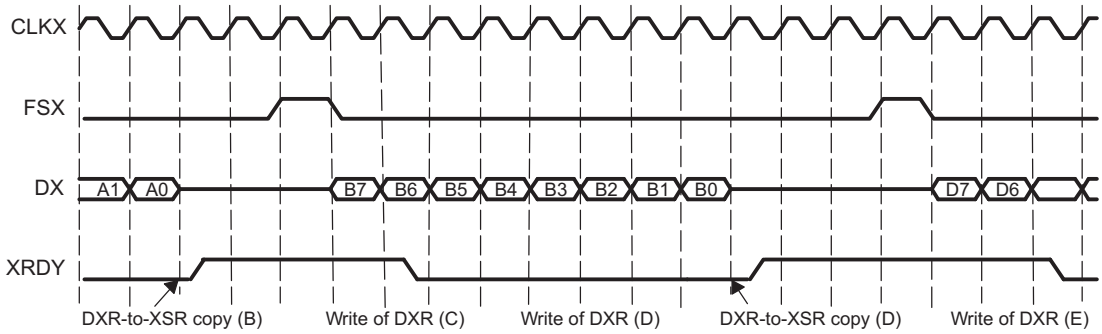



**Figure 17-10. McBSP Transmission Signal Activity**




The following process shows how.....

1. The CPU or the DMA controller writes data to the data transmit register. When the transmit buffer is reached the transmitter ready bit (XRDY) is cleared in SPCR2_REG[1] register to indicate that the transmitter is not ready for new data.
2. When new data arrives in DXR_REG register, the McBSP copies the content of the data transmit register to the transmit buffer. In addition, the Transmit Ready bit (XRDY) is set as long as the buffer contains at least the transmit threshold number of free locations. This indicates that the transmitter is ready to accept new data from the CPU or the DMA controller.
3. The McBSP waits for a transmit frame–synchronization pulse on internal FSX.
4. When the pulse arrives, the McBSP inserts the appropriate data delay that is selected with the XCR2_REG[1:0] register XDATDLY bits. In the preceding timing diagram, a 1-bit data delay is selected.
5. The McBSP shifts data bits from the transmit shift register to the McBSP.DX pin.

When activity is not properly timed, errors can occur. See the following topics for more details:

- Underflow in the transmitter
- Overflow in the transmitter
- Unexpected transmit frame-synchronization pulse


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.1.7 Enable/Disable the Transmit and Receive Processes

The McBSP has the option to stop-resume the transmit/receive process while the module is in functional mode (out of transmit/receive reset).

When the transmit/receive disable XDISABLE/RDISABLE bit in XCCR_REG/RCCR_REG register is set, the McBSP will stop the transmit/receive operation at the next frame boundary (frame corruption avoided).

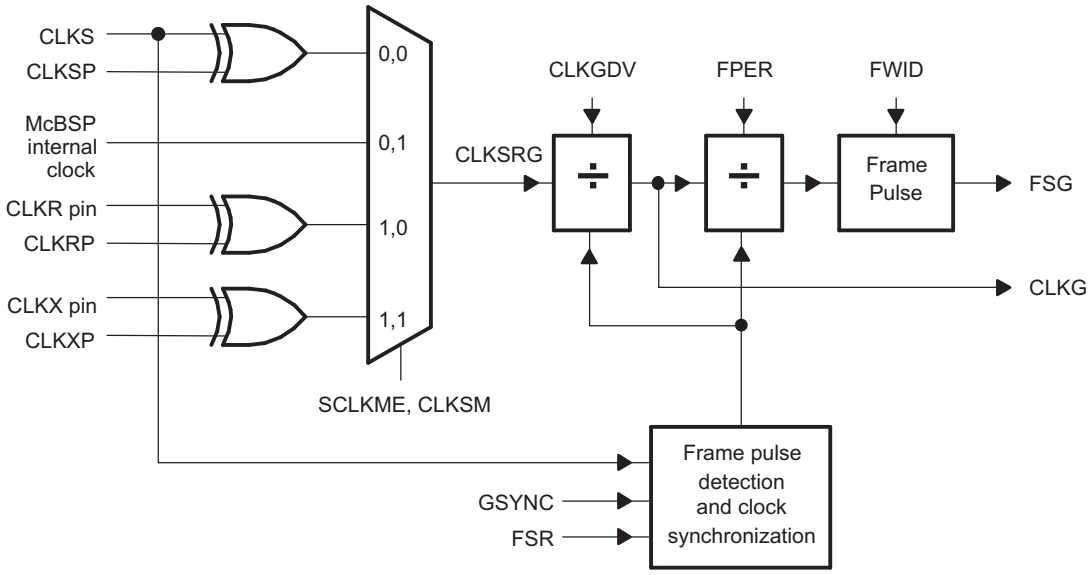
As soon as the XDISABLE/RDISABLE bit is cleared, the transmit/receive process will resume at the next frame boundary.

Note that it is not recommended to use this mechanism together with the possibility to interrogate the transmit/receive buffer status register (XBUFFSTAT/RBUFFSTAT indicating the occupied/available buffer locations), since this register is a CPU (OCP) clock synchronous register and does not reflect the exact number of occupied/free locations available on the functional clock domain.

### *17.2.2 McBSP Sample Rate Generator*

The McBSP contains a sample rate generator that can be used to generate an internal data clock (CLKG) and an internal frame–synchronization signal (FSG). CLKG can be used for bit shifting on the data receive (McBSP.DR) pin and/or the data transmit (McBSP.DX) pin. FSG can be used to initiate frame transfers on McBSP.DR and/or McBSP.DX. Figure 17-11 is a conceptual block diagram of the sample rate generator.

**Figure 17-11. Sample Rate Generator Block Diagram**




The source clock for the sample rate generator (labeled CLKSRG in the diagram) can be supplied by either the FCLK functional clock (OCP interface clock), or by an external pin (McBSP.CLKS, McBSP.CLKX, or McBSP.CLKR). The source is selected with the SCLKME bit of the PCR_REG[7] register and the CLKSM bit of the SRGR2_REG[13] register.

If a pin is used, the polarity of the incoming signal can be inverted with the appropriate polarity bit (CLKSP bit of SRGR2_REG[14] register, CLKXP bit of PCR_REG register, or CLKRP bit of PCR_REG[0] register).

The sample rate generator has a three-stage clock divider that gives CLKG and FSG programmability.

The three stages provide:

- Clock divide-down. The source clock is divided according to the SRGR1_REG[7:0] register CLKGDV bits to produce CLKG .
- Frame period divide-down. CLKG is divided according to the SRGR2_REG[11:0] register FPER bit field to control the period from the start of a frame–pulse to the start of the next pulse.
- Frame-synchronization pulse-width countdown. CLKG cycles are counted according to the SRGR1_REG[15:8] register FWID bits to control the width of each frame-synchronization pulse.


Copyright © 2011–2016, Texas Instruments Incorporated

Note: The McBSP cannot operate at an internal functional frequency faster than 55 MHz. Choose an input clock frequency and a CLKGDV value such that CLKG is less than or equal to 55 MHz. Please see the device datasheet for maximum McBSP CLKX cycle time supported on your device.

In addition to the three-stage clock divider, the sample rate generator has a frame-synchronization pulse detection and clock synchronization module that allows synchronization of the clock divide down with an incoming frame-synchronization pulse on the McBSP.FSR pin. This feature is enabled or disabled with the SRGR2_REG[15] register GSYNC bit.

CLKG is used as source in order to generate the output clocks CLKX/CLKR when the CLKXM/CLKRM indicates that the clock is an output. The output CLKX/CLKR is generated according to the clock polarity setting given by the CLKXP/CLKRP as follows: when falling edge, the CLKX/CLKR is CLKG inverted.

#### 17.2.2.1 Clock Generation in the Sample Rate Generator

The sample rate generator can produce a clock signal (CLKG) for use by the receiver, the transmitter, or both. Use of the sample rate generator to drive clocking is controlled by the clock mode bits (CLKRM and CLKXM) in the pin control register (PCR_REG[8] and PCR_REG[9] respectively) and polarity mode bits (CLKRP and CLKXP).

When a clock mode bit is set to 1 (CLKRM = 1 for reception, CLKXM = 1 for transmission), the corresponding data clock (CLKR for reception, CLKX for transmission) is driven by the internal sample rate generator output clock (CLKG) according with the polarity setting.

The effects of CLKRM = 1 and CLKXM = 1 on the McBSP are partially affected by the use of the digital loop back mode, the analog loop back mode and by the synchronous receive/transmit setting, respectively, as described in Table 17-2. The analog loop back mode is selected with the SPCR1_REG[15] register ALB bit. The digital loop back mode is selected with the DLB bit of the XCCR_REG register. The synchronous mode is controlled by the ALBRXCTRL input pin.

When using the sample rate generator as a clock source, make sure the sample rate generator is enabled (GRST = 1).

**Table 17-2. Effects of DLB and ALB Bits on Clock Modes**

| Mode Bit Settings | | Effect |
|---|---|---|
| CLKRM = 1 | DLB = 0 & ALB = 0<br>(Digital & analog loop back mode disabled) | CLKR is an output pin driven by the sample rate generator output clock (CLKG). |
| | DLB = 0 & ALB = 1<br>(Analog loop back mode enabled) | CLKR is an output pin driven by the sample rate generator output clock (CLKG).<br>The receive functional part internal clock is driven by CLKX input pin CLKXI. The source of CLKX depends on the CLKXM bit.<br>The receive frame synchronization is driven by FSX input pin (FSXI).<br>The receive data is driven by the DX loop back pin (DXI). |
| | DLB = 1 & ALB = 0<br>(Digital loop back mode enabled) | CLKR is not driven.<br>The sample rate generator and the frame synchronization generator need to be enabled.<br>The internal transmit and receive clocks are driven by SRG (CLKG having the appropriate CLKXP polarity).<br>The transmit and receive frame synchronization signals are driven by FSG (having the appropriate FSXP polarity).<br>The transmit data is connected to the receive input data.<br>Note that in digital loop back mode no serial link activity will be seen by the remote device because CLKREN, CLKXEN, FSREN, FSRXEN, DXEN are not active. |
| | DLB = 1 & ALB = 1 (reserved mode) | Undefined functionality. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-2. Effects of DLB and ALB Bits on Clock Modes (continued)**

| Mode Bit Settings | | Effect |
|---|---|---|
| CLKXM = 1 | DLB = 0 & ALB = 0<br>(Digital & analog loop back mode disabled) | CLKX is an output pin driven by the sample rate generator output clock (CLKG). |
| | DLB = 0 & ALB = 1<br>(Analog loop back mode enabled) | CLKX is an output pin driven by the sample rate generator output clock (CLKG). |
| | DLB = 1 & ALB = 0<br>(Digital loop back mode enabled) | CLKX is not driven.<br>The sample rate generator and the frame synchronization generator need to be enabled.<br>The internal transmit and receive clocks are driven by SRG (CLKG having the appropriate CLKXP polarity).<br>The transmit and receive frame synchronization signals are driven by FSG (having the appropriate FSXP polarity).<br>The transmit data is connected to the receive input data.<br>Note that in digital loop back mode no serial link activity will be seen by the remote device because CLKREN, CLKXEN, FSREN, FSRXEN, DXEN are not active. |
| | DLB = 1 & ALB = 1 (reserved mode) | Undefined functionality. |
| | ALBRXCTRL[0] = 1<br>(synchronous mode DLB = 0 & ALB = 0) | CLKX is an output pin driven by the sample rate generator output clock (CLKG). CLKR is connected to the CLKX. |

#### 17.2.2.2 Frame Synchronization Generation in the Sample Rate Generator

The sample rate generator can produce a frame–synchronization signal (FSG) for use by the receiver, the transmitter, or both.

If you want the receiver to use FSG for frame synchronization, make sure PCR_REG[10] register FSRM bit = 1. (When FSRM = 0, receive frame synchronization is supplied via the McBSP.FSR pin.)

If you want the transmitter to use FSG for frame synchronization, you must set:

- FSXM = 1 in PCR_REG[11] register: This indicates that transmit frame synchronization is supplied by the McBSP itself rather than from the McBSP.FSX pin
- FSGM = 1 in SRGR2_REG[12] register: This indicates that when FSXM = 1, transmit frame synchronization is supplied by the sample rate generator. (When FSGM = 0 and FSXM = 1, the transmit frame-synchronization signal (FSX) is generated when transmit buffer is not empty. When FSGM = 0, FPER and FWID are used to determine the frame synchronization period and width (external FSX is gated by the buffer empty condition).

In either case, the sample rate generator must be enabled (SPCR2_REG[6] register GRST bit = 1) and the frame–synchronization logic in the sample rate generator must be enabled (SPCR2_REG[7] register FRST bit = 0).

##### *17.2.2.2.1 Choosing the Width of the Frame-Synchronization Pulse*

Each pulse on FSG has a programmable width. You program the SRGR1_REG[15:8] register FWID bits, and the resulting pulse width is (FWID + 1) CLKG cycles, where CLKG is the output clock of the sample rate generator.

##### *17.2.2.2.2 Controlling the Period Between the Starting Edges of the Frame-Synchronization Pulses*

You can control the amount of time from the starting edge of one FSG pulse to the starting edge of the next FSG pulse. This period is controlled in one of two ways, depending on the configuration of the sample rate generator:

- If the sample rate generator is using an external input clock and SRGR2_REG[15] register GSYNC bit = 1, FSG pulses in response to an inactive–to–active transition on the McBSP.FSR pin. Thus, an external device controls the frame–synchronization period.
- Otherwise, you program the FPER bits of SRGR2_REG[11:0] register, and the resulting frame–synchronization period is (FPER + 1) CLKG cycles, where CLKG is the output clock of the sample rate generator.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.2.2.3 Keeping FSG Synchronized to an External Clock

When an external signal is selected to drive the sample rate generator, the GSYNC bit of SRGR2_REG15] register and the McBSP.FSR pin can be used to configure the timing of FSG pulses.

GSYNC = 1 ensures that the McBSP and an external device are dividing down the input clock with the same phase relationship. If GSYNC = 1, an inactive–to–active transition on the McBSP.FSR pin triggers a resynchronization of CLKG and generation of FSG.

### 17.2.2.3 Synchronizing Sample Rate Generator Outputs to an External Clock

The sample rate generator can produce a clock signal (CLKG) and a frame–synchronization signal (FSG) based on an input clock signal that is either the FCLK functional clock signal or a signal at the McBSP.CLKS, McBSP.CLKR, or McBSP.CLKX pin. When an external clock is selected to drive the sample rate generator, the GSYNC bit of SRGR2_REG[15] register and the McBSP.FSR pin can be used to control the timing of CLKG and the pulsing of FSG relative to the chosen input clock.

Make GSYNC = 1 when you want the McBSP and an external device to divide down the input clock with the same phase relationship. If GSYNC = 1:

- An inactive–to–active transition on the McBSP.FSR pin triggers a resynchronization of CLKG and a pulsing of FSG.
- CLKG always begins with a high state after synchronization.
- FSR is always detected at the same edge of the input clock signal that generates CLKG, no matter how long the FSR pulse is.
- The SRGR2_REG[11:0] register FPER bits are ignored because the frame–synchronization period on FSG is determined by the arrival of the next frame–synchronization pulse on the McBSP.FSR pin.

If GSYNC = 0, CLKG runs freely and is not resynchronized, and the frame–synchronization period on FSG is determined by FPER.

#### 17.2.2.3.1 Operating the Transmitter Synchronously with the Receiver

When GSYNC = 1, the transmitter can operate synchronously with the receiver, provided that:

- FSX is programmed to be driven by FSG (FSGM = 1 in SRGR2_REG[12] register and FSXM = 1 in PCR_REG[11] register). If the input FSR has appropriate timing so that it can be sampled by the falling edge of CLKG, it can be used, instead, by setting FSXM = 0 and connecting FSR to FSX externally.
- The sample rate generator clock drives the transmit and receive clocking (PCR_REG[8] CLKRM bit and PCR_REG[9] CLKXM bit are set to 1). Therefore, the CLK(R/X) pin must not be driven by any other driving source.

#### 17.2.2.3.2 Synchronization Examples

Figure 17-12 and Figure 17-13 show the clock and frame–synchronization operation with various polarities of CLKS (the chosen input clock) and FSR. These figures assume FWID = 0 in SRGR1_REG[15:8] register, for an FSG pulse that is one CLKG cycle wide. The FPER bits of SRGR2_REG[11:0] register are not programmed; the period from the start of a frame–synchronization pulse to the start of the next pulse is determined by the arrival of the next inactive–to–active transition on the McBSP.FSR pin. Each of the figures shows what happens to CLKG when it is initially synchronized and GSYNC = 1, and when it is not initially synchronized and GSYNC = 1. Figure 17-13 has a slower CLKG frequency (it has a larger divide–down value in the CLKGDV bits of SRGR1_REG[7:0] register).


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 17-12. CLKG Synchronization and FSG Generation (GSYNC = 1 and CLKGDV = 1)**

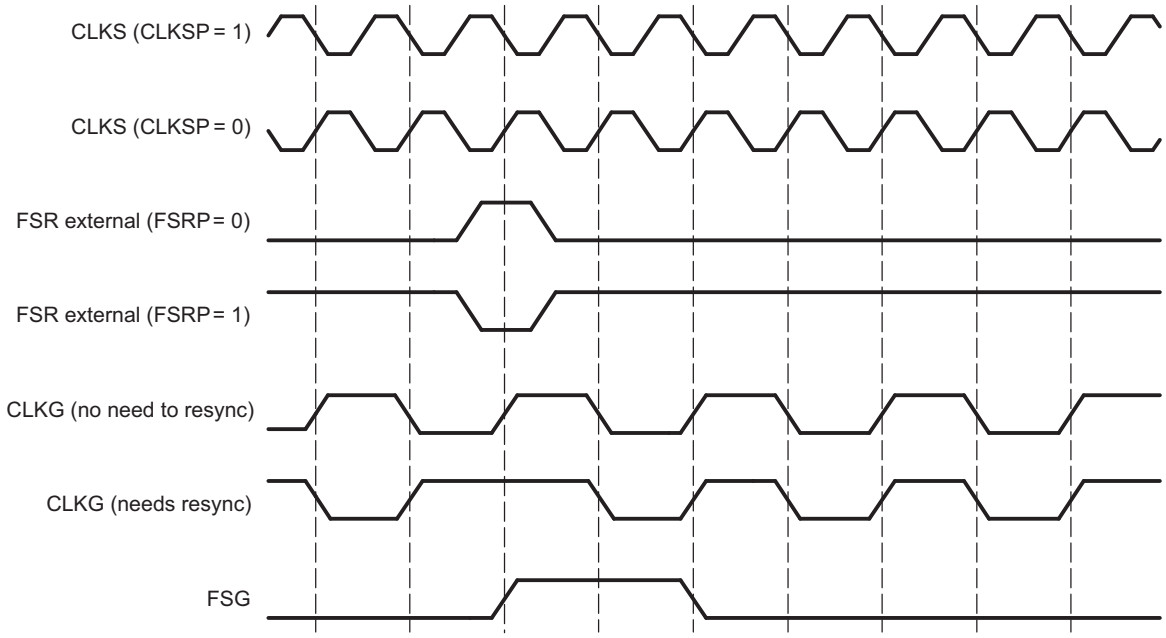
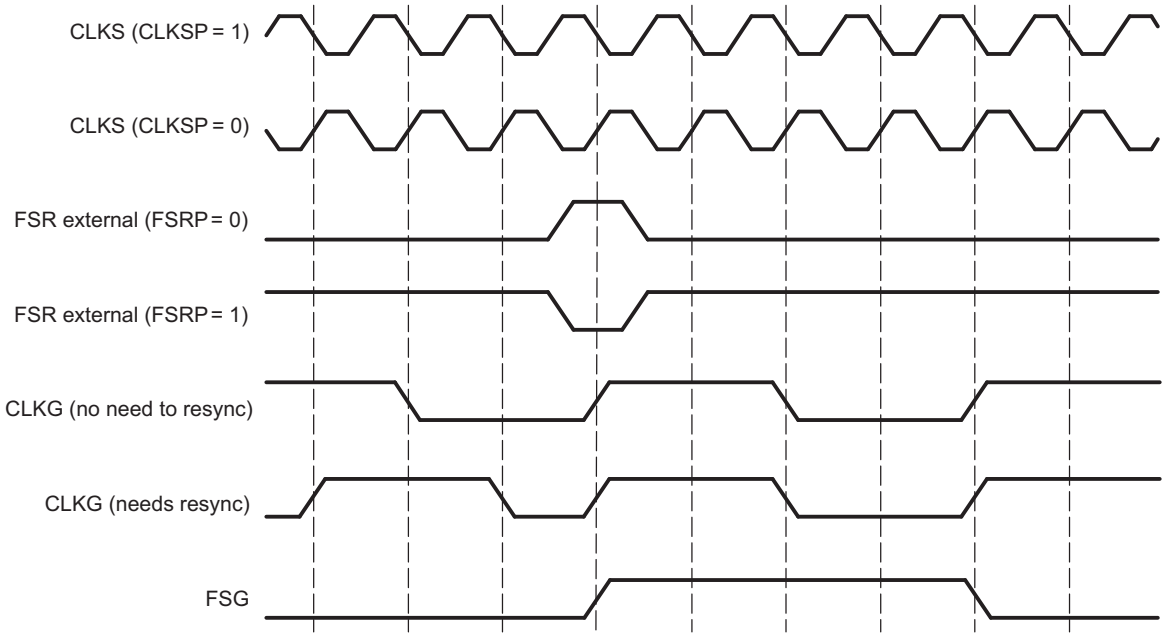



**Figure 17-13. CLKG Synchronization and FSG Generation (GSYNC = 1 and CLKGDV = 3h)**





Copyright © 2011–2016, Texas Instruments Incorporated

### 17.2.3 McBSP Exception/Error Conditions

There are several serial port events that can constitute a system error. Any of this error conditions may be sources of an interrupt:

- Receiver Overrun

(ROVFLSTAT bit is set to 1 in IRQSTATUS register, and legacy mode RFULL bit is set to 1 in SPCR1_REG register)

This occurs when receive buffer is full and RSR(s) are full with another new word shifted in from McBSP.DR. Therefore, ROVFLSTAT (RFULL) indicates an error condition wherein any new data that can arrive at this time on McBSP.DR replaces the contents of the RSR, and the previous word is lost. The RSR continue to be overwritten as long as new data arrives on McBSP.DR and DRR_REG register is not read.

- Unexpected Receive Frame-Synchronization Pulse

(RSYNCERR bit is set to 1 in IRQSTATUS register, and legacy mode RSYNCERR bit is set to 1 in SPCR1_REG register)

This occurs during reception when an unexpected frame–synchronization pulse arrives. An unexpected frame-synchronization pulse is one that is supposed to begin the next frame transfer before all the bits of the current frame have been received. Such a pulse is ignored by the receiver, but sets the RSYNCERR bit in SPCR1_REG.

- Receiver Underflow

(RUNDFLSTAT bit is set to 1 in IRQSTATUS register)

This occurs when DMA controller or CPU reads data from an empty receive buffer.

- Transmitter Underflow

(XUNDFLSTAT bit is set to 1 in IRQSTATUS register, and legacy mode XEMPTY bit is cleared to 0 in SPCR2_REG register)

If a new frame–synchronization signal arrives when transmit buffer is empty, the previous data in the XSR is sent again. This procedure continues for every new frame–synchronization pulse that arrives until DXR_REG register is loaded with new data (and the XB buffer will be no longer empty).

- Unexpected Transmit Frame-Synchronization Pulse

(XSYNCERR bit is set to 1 in IRQSTATUS register, and legacy mode XSYNCERR bit is set to 1 in SPCR2_REG register)

This occurs during transmission when an unexpected frame–synchronization pulse arrives. An unexpected frame–synchronization pulse is one that is supposed to begin the next frame transfer before all the bits of the current frame have been transferred. Such a pulse is ignored by the transmitter, but sets the XSYNCERR bit in SPCR2_REG.

- Transmitter Overflow

(XOVFLSTAT bit is set to 1 in IRQSTATUS register)

This occurs when DMA controller or CPU write data to a full transmit buffer.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.3.1 Overrun in the Receiver

(ROVFLSTAT bit set to 1 in IRQSTATUS register, and legacy mode RFULL bit set to 1 in SPCR1_REG register)

Indicates that the receiver has experienced overrun and is in an error condition. Receive overrun is set when all of the following conditions are met:

- DRR_REG is not read even if the RRDY bit in IRQSTATUS register is set (legacy mode: RRDY bit in SPCR1_REG register) and DMA or interrupt request has been asserted.
- RB is full
- RSR is full

As previously described, data arriving on McBSP.DR is continuously shifted into. Once a complete word is shifted into the RSR, an RSR–to–RB copy can occur only if the RB is not full.

Either of the following events clears the legacy mode RFULL bit and allows subsequent transfers to be read properly:

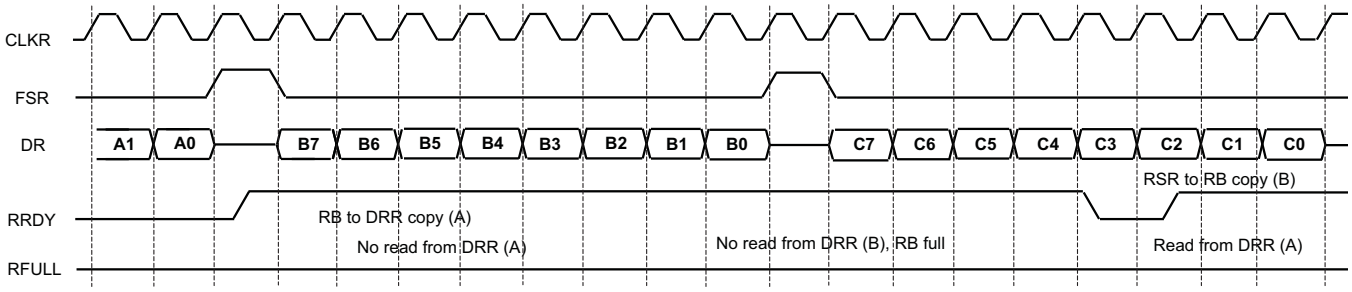
- The CPU or DMA controller reads DRR_REG.
- The receiver is reset individually (SPCR1_REG[0] register RRST bit = 0) or as part of an Global reset.

Another frame–synchronization pulse is required to restart the receiver.

According to the IRQENABLE register setting, this condition can generate the COMMONIRQ line to be asserted low. Writing 1 to the corresponding bit in status register will clear the interrupt.

Figure 17-14 shows the receive overrun condition.

**Figure 17-14. Overrun in the McBSP Receiver**





Copyright © 2011–2016, Texas Instruments Incorporated

### 17.2.3.2 Unexpected Receive Frame-Synchronization Pulse

#### *17.2.3.2.1 Possible Responses to Receive Frame-Synchronization Pulses*

If a frame–synchronization pulse starts the transfer of a new frame before the current frame is fully received, this pulse is treated as an unexpected frame–synchronization pulse, and the receiver sets the receive frame–synchronization error bit RSYNCERR in IRQSTATUS (and the legacy RSYNCERR bit SPCR1_REG[3]) register.

According to the IRQENABLE register settings this condition can generate the COMMONIRQ line to be asserted low. Writing 1 to the corresponding bit in status register will clear the interrupt.

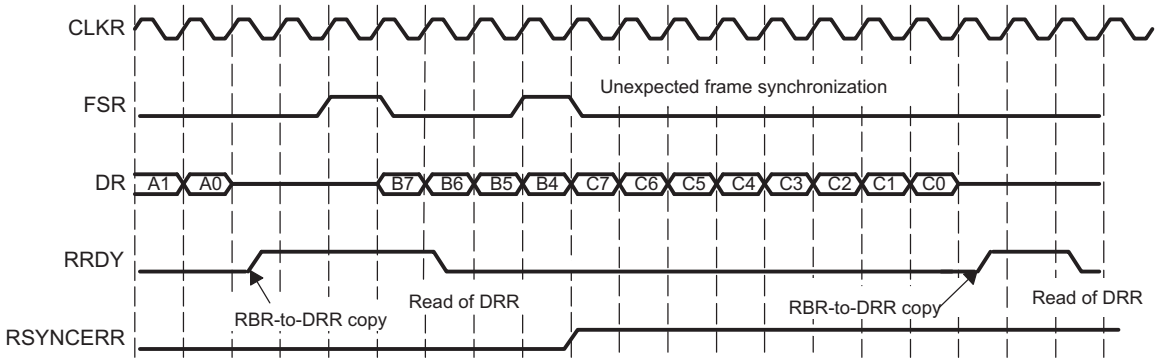
Using the legacy mode, RSYNCERR bit in SPCR1_REG can be cleared only by a receiver reset or by a write of 0 to this bit. If you want the McBSP to notify the CPU of receive frame–synchronization errors, you can set the legacy mode receive interrupt with the SPCR1_REG[5:4] register RINTM bits. When RINTM = 11b, the McBSP sends a receive interrupt (legacy mode RINT) request to the CPU each time that RSYNCERR is set.

#### *17.2.3.2.2 Example of an Unexpected Receive Frame-Synchronization Pulse*

Figure 17-15 shows an unexpected receive frame–synchronization pulse during normal operation of the serial port, with time intervals between data packets.

Note that the unexpected receive frame–synchronization pulse does not influence the data receive process, being ignored by the data receive state machine.

**Figure 17-15. Unexpected Frame-Synchronization Pulse During a McBSP Reception**



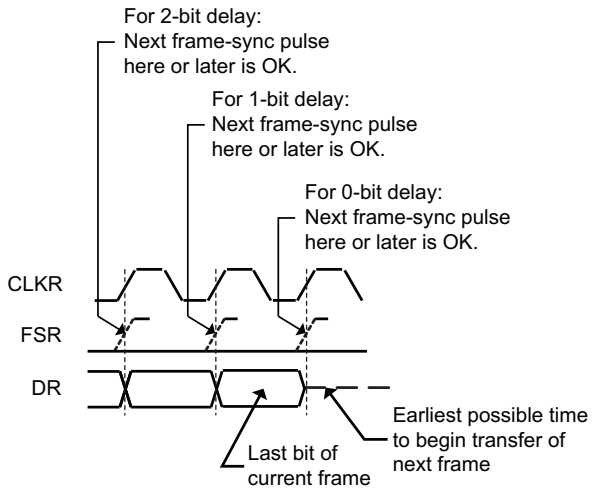



Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.3.2.3 Preventing Unexpected Receive Frame-Synchronization Pulses

Each frame transfer can be delayed by 0, 1, or 2 CLKR cycles, depending on the value of the RCR2_REG[1:0] RDATDLY bits. For each possible data delay, Figure 17-16 shows when a new frame–synchronization pulse on FSR can safely occur relative to the last bit of the current frame.

**Figure 17-16. Proper Positioning of Receive Frame-Synchronization Pulses**




### 17.2.3.3 Underflow in the Receiver

The McBSP indicates a receiver underflow condition by setting the RUNDFLSTAT bit in IRQSTATUS register. This error occurs when DMA controller or CPU reads data from an empty receive buffer (this may happen only if the CPU or DMA controller does not respect the DMA length, does not wait for DMA request or does not check the buffer status before reading data. According to the IRQENABLE register settings this condition can generate the COMMONIRQ line to be asserted low. Writing 1 to the corresponding bit in status register will clear the interrupt.

### 17.2.3.4 Underflow in the Transmitter

The McBSP indicates a transmitter empty (or underflow) condition by setting the XUNDFLSTAT bit in IRQSTATUS register. Also the legacy mode XEMPTY bit in SPCR2_REG[2] register is cleared. Either of the following events activates XEMPTY (XEMPTY = 0):

- DXR_REG has not been loaded and transmit buffer is empty, and all bits of the data word in the XSR have been shifted out on the McBSP.DX pin.
- The transmitter is reset (by forcing XRST = 0 in SPCR2_REG[0] register, or by an Global reset) and is then restarted.

XEMPTY is deactivated (XEMPTY = 1) when a new word in DXR_REG is transferred to XB. If FSXM = 1 in PCR_REG[11] and FSGM = 0 in SRGR2_REG[12], the transmit frame-synchronization signal (FSX) is generated when transmit buffer is not empty. When FSGM = 0, FPER and FWID are used to determine the frame synchronization period and width (external FSX is gated by the buffer empty condition). Otherwise, the transmitter waits for the next frame–synchronization pulse before sending out the next frame on McBSP.DX.

When the transmitter is taken out of reset (XRST = 1), it is in a transmitter ready state (SPCR2_REG[1] register XRDY bit = 1) and transmitter empty (XEMPTY = 0) state. If DXR_REG is loaded by the CPU or the DMA controller before internal FSX goes active high, a valid XB–to–XSR transfer occurs. This allows for the first word of the first frame to be valid even before the transmit frame–synchronization pulse is generated or detected. Alternatively, if a transmit frame–synchronization pulse is detected before DXR_REG is loaded, zeros are output on McBSP.DX.


Copyright © 2011–2016, Texas Instruments Incorporated

The XUNDFLSTAT in IRQSTATUS register indicates a real underflow condition, in which the frame is corrupted due to lack of data availability during transmit process. According to the IRQENABLE register settings this condition can generate the COMMONIRQ line to be asserted low. Writing 1 to the corresponding bit in status register will clear the interrupt.

### 17.2.3.5 Unexpected Transmit Frame-Synchronization Pulse

#### *17.2.3.5.1 Possible Responses to Transmit Frame-Synchronization Pulses*

If a frame–synchronization pulse starts the transfer of a new frame before the current frame is fully transmitted, this pulse is treated as an unexpected frame–synchronization pulse, and the transmitter sets the transmit frame–synchronization error bit XSYNCERR in IRQSTATUS (and the legacy XSYNCERR bit in SPCR2_REG[3]) register.

According to the IRQENABLE register settings this condition can generate the COMMONIRQ line to be asserted low. Writing 1 to the corresponding bit in status register will clear the interrupt.

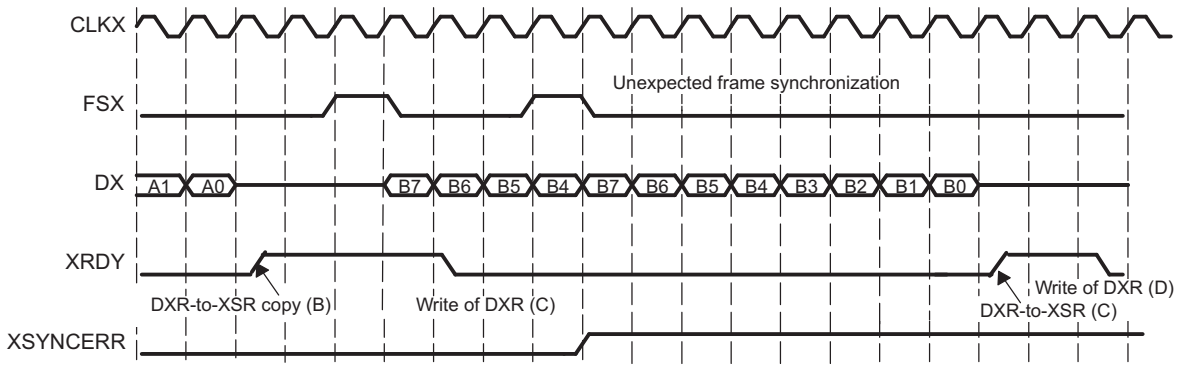
Using the legacy mode, XSYNCERR bit in SPCR2_REG can be cleared only by a transmitter reset or by a write of 0 to this bit. If you want the McBSP to notify the CPU of frame–synchronization errors, you can set a special transmit interrupt mode with the SPCR2_REG[5:4] register XINTM bits. When XINTM = 11b, the McBSP sends a transmit interrupt (XINT) request to the CPU each time that XSYNCERR is set.

#### *17.2.3.5.2 Example of Unexpected Transmit Frame-Synchronization Pulse*

Figure 17-17 shows an unexpected transmit frame–synchronization pulse during normal operation of the serial port with intervals between the data packets.

Note that the unexpected transmit frame–synchronization pulse does not influence the data transmit process, being ignored by the data transmit state machine.

**Figure 17-17. Unexpected Frame-Synchronization Pulse During a McBSP Transmission**



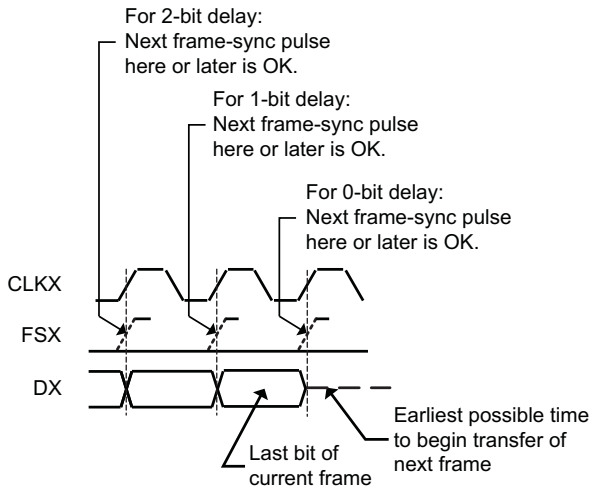



Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.3.5.3 Preventing Unexpected Transmit Frame-Synchronization Pulses

Each frame transfer can be delayed by 0, 1, or 2 CLKX cycles, depending on the value in the XCR2_REG[1:0] register XDATDLY bits. For each possible data delay, Figure 17-18 shows when a new frame–synchronization pulse on FSX can safely occur relative to the last bit of the current frame.

**Figure 17-18. Proper Positioning of Transmit Frame-Synchronization Pulses**




### 17.2.3.6 Overflow in the Transmitter

The McBSP indicates a transmitter overflow condition by setting the XOVFLSTAT bit in IRQSTATUS[11] register. This error occurs when DMA controller or CPU write data to a full transmit buffer (this may happen only if the CPU or DMA controller does not respect the DMA length, does not wait for DMA request or does not check the buffer status before writing data. According to the IRQENABLE register settings this condition can generate the COMMONIRQ line to be asserted low. Writing 1 to the corresponding bit in status register will clear the interrupt.

## 17.2.4 McBSP DMA Configuration

The McBSP receive and transmit data DMA requests are active after the receive RRST and transmit XRST are released. After reset the default DMA threshold (and length) is one.

The receive and transmit DMA requests can be individually disabled by setting zero the RDMAEN, XDMAEN bits in RCCR_REG respectively XCCR_REG. When disabling the DMA the DMA request line is deasserted even if a DMA transfer is pending and the DMA state machine is not reset.

The DMA threshold and length configuration is done through THRSH1_REG and THRSH2_REG registers as follows:

- (THRSH1_REG + 1) value represents the required receive DMA request length (the length of the transfer is the same as the threshold value plus one). As long as the RB occupied locations level is above or equal to the THRSH1_REG value plus one, the DMA request will be asserted. After transferring the configured (THRSH1_REG + 1) number of words, the receive DMA request (McBSP.REVNT) will be deasserted and reasserted as soon as the conditions are met again.
- (THRSH2_REG + 1) value represents the required transmit DMA request length (the length of the transfer is the same as the threshold value plus one). As long as the XB free locations level is above or equal to the THRSH2_REG value plus one the DMA request will be asserted. After transferring the configured (THRSH2_REG + 1) number of words, the transmit DMA request (McBSP.XEVNT) will be deasserted and reasserted as soon as the conditions are met again.


Copyright © 2011–2016, Texas Instruments Incorporated

Note that the CPU may decide not to use the DMA to transfer the data. In this case the DMA must be disabled (or the DMA request can be ignored by CPU) and the common interrupt line can be used. The RRDY bit for receive and XRDY bit for transmit will indicate when the threshold values are reached. Also, by reading the receive buffer status RBUFFSTAT_REG register and transmit buffer status XBUFFSTAT_REG register the CPU may decide to transfer data even if the threshold is not reached. This mechanism is useful on the last transfer on receive side when the threshold value is bigger than the occupied locations inside the receive buffer and the CPU needs to read this data. Since no interrupt or DMA request is asserted the only option in this case is to read the receive buffer status register value and to transfer the remaining data without using the DMA or interrupt indication.

## *17.2.5 Multichannel Selection Modes*

### 17.2.5.1 Channels, Blocks, and Partitions

A McBSP channel is a time slot for shifting in/out the bits of one serial word. The McBSP supports up to 128 channels for reception and 128 channels for transmission. In the receiver and in the transmitter, the 128 available channels are divided into 8 blocks that each contains 16 contiguous channels:

**Table 17-3. Channels, Block, Partitions**

| | |
|---|---|
| Block 0: Channels 0–15 | Block 4: Channels 64–79 |
| Block 1: Channels 16–31 | Block 5: Channels 80–95 |
| Block 2: Channels 32–47 | Block 6: Channels 96–111 |
| Block 3: Channels 48–63 | Block 7: Channels 112–127 |

The blocks are assigned to partitions according to the selected partition mode. In the 2–partition mode, you assign one even–numbered block (0, 2, 4, or 6) to partition A and one odd–numbered block (1, 3, 5, or 7) to partition B. In the 8–partition mode, blocks 0 through 7 are automatically assigned to partitions, A through H, respectively.

The number of partitions for reception and the number of partitions for transmission are independent. For example, it is possible to use two receive partitions (A and B) and eight transmit partitions (A–H).

### 17.2.5.2 Multichannel Selection

When a McBSP uses a time–division multiplexed (TDM) data stream while communicating with other McBSPs or serial devices, the McBSP may need to receive and/or transmit on only a few channels. To save memory and bus bandwidth, you can use a multichannel selection mode to prevent data flow in some of the channels.

Each channel partition has a dedicated channel enable register. If the appropriate multichannel selection mode is on, each bit in the register controls whether data flow is allowed or prevented in one of the channels that is assigned to that partition.

The McBSP has one receive multichannel selection mode and three transmit multichannel selection modes.

### 17.2.5.3 Configuring a Frame for Multichannel Selection

Before you enable a multichannel selection mode, make sure you properly configure the data frame:

- Select a single–phase frame (RCR2_REG[15] register RPHASE bit and XCR2_REG[15] register XPHASE bit = 0). Each frame represents a TDM data stream.
- Set a frame length (in RCR1_REG[14:8] register RFRLEN1 bit field and in XCR1_REG[14:8] register XFRLEN1 bit field) that includes the highest–numbered channel to be used. For example, if you plan to use channels 0, 15, and 39 for reception, the receive frame length must be at least 40 (RFRLEN1 = 39). If XFRLEN1 = 39 in this case, the receiver creates 40 time slots per frame but only receives data during time slots 0, 15, and 39 of each frame.


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.2.5.4 Using Eight Partitions

For multichannel selection operation in the receiver and/or the transmitter, you can use eight partitions or two partitions. If you choose the 8–partition mode (RMCME = 1 for reception, XMCME = 1 for transmission), McBSP channels are activated in the following order: A, B, C, D, E, F, G, H.

In response to a frame–synchronization pulse, the receiver or transmitter begins with the channels in partition A and then continues with the other partitions in order until the complete frame has been transferred. When the next frame–synchronization pulse occurs, the next frame is transferred, beginning with the channels in partition A.

In the 8–partition mode, the (R/X)PABLK and (R/X)PBBLK bits are ignored and the 16–channel blocks are assigned to the partitions as shown in the following tables. These assignments cannot be changed. The tables also show the registers used to control the channels in the partitions.

Figure 17-19 shows an example of the McBSP using the 8–partition mode. In response to a frame–synchronization pulse, the McBSP begins a frame transfer with partition A and then activates B, C, D, E, F, G, and H to complete a 128–word frame.

**Table 17-4. Eight Partitions – Receive Channel Assignment and Control**

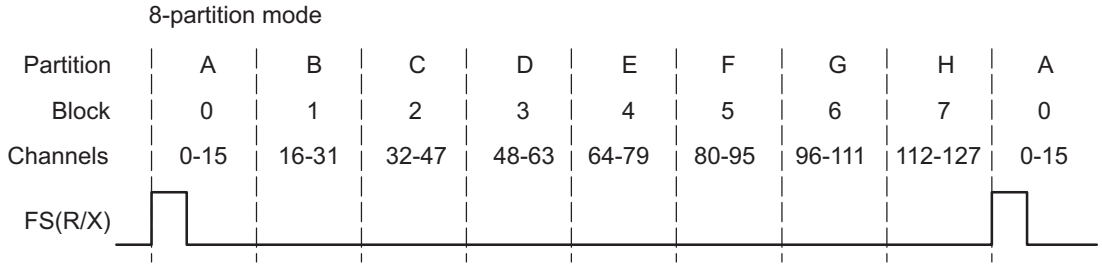
| Receive Partition | Assigned Block of Receive Channels | Register Used for Channel Control |
|---|---|---|
| A | Block 0: channels 0 through 15 | RCERA_REG |
| B | Block 1: channels 16 through 31 | RCERB_REG |
| C | Block 2: channels 32 through 47 | RCERC_REG |
| D | Block 3: channels 48 through 63 | RCERD_REG |
| E | Block 4: channels 64 through 79 | RCERE_REG |
| F | Block 5: channels 80 through 95 | RCERF_REG |
| G | Block 6: channels 96 through 111 | RCERG_REG |
| H | Block 7: channels 112 through 127 | RCERH_REG |

**Table 17-5. Eight Partitions – Transmit Channel Assignment and Control**

| Transmit Partition | Assigned Block of Receive Channels | Register Used for Channel Control |
|---|---|---|
| A | Block 0: channels 0 through 15 | XCERA_REG |
| B | Block 1: channels 16 through 31 | XCERB_REG |
| C | Block 2: channels 32 through 47 | XCERC_REG |
| D | Block 3: channels 48 through 63 | XCERD_REG |
| E | Block 4: channels 64 through 79 | XCERE_REG |
| F | Block 5: channels 80 through 95 | XCERF_REG |
| G | Block 6: channels 96 through 111 | XCERG_REG |
| H | Block 7: channels 112 through 127 | XCERH_REG |


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 17-19. McBSP Data Transfer in the 8-Partition Mode**




### 17.2.5.5 Receive Multichannel Selection Mode

The MCR1_REG[0] register RMCM bit determines whether all channels or only selected channels are enabled for reception. When RMCM = 0, all 128 receive channels are enabled and cannot be disabled. When RMCM = 1, the receive multichannel selection mode is enabled. In this mode:

- Channels can be individually enabled or disabled. The only channels enabled are those selected in the appropriate receive channel enable registers (RCER[A,H]_REG). The way channels are assigned to the RCER[A,H]_REG registers depends on the number of receive channel partitions (two or eight), as defined by the MCR1_REG[9] register RMCME bit.
- If a receive channel is disabled, any bits received in that channel are not transferred to the receive buffer (RB), and as a result, does not set the receiver ready bit (RRDY). Therefore, no DMA synchronization event (REVT) is generated and, if the receiver interrupt mode depends on RRDY (RINTM = 00b), no interrupt is generated.

As an example of how the McBSP behaves in the receive multichannel selection mode, suppose you enable only channels 0, 15, and 39 and that the frame length is 40. The McBSP:

- Accepts bits shifted in from the McBSP.DR pin in channel 0
- Ignores bits received in channels 1–14
- Accepts bits shifted in from the McBSP.DR pin in channel 15
- Ignores bits received in channels 16–38
- Accepts bits shifted in from the McBSP.DR pin in channel 39

### 17.2.5.6 Using Two Partitions (legacy only)

For multichannel selection operation in the receiver and/or the transmitter, you can use two partitions or eight partitions. If you choose the 2–partition mode (RMCME = 0 for reception, XMCME = 0 for transmission), McBSP channels are activated using an alternating scheme. In response to a frame–synchronization pulse, the receiver or transmitter begins with the channels in partition A and then alternates between partitions B and A until the complete frame has been transferred. When the next frame–synchronization pulse occurs, the next frame is transferred beginning with the channels in partition A.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.5.6.1 Assigning Blocks to Partitions A and B

For reception, any two of the eight receive–channel blocks can be assigned to receive partitions A and B, which means up to 32 receive channels can be enabled at any given point in time. Similarly, any two of the eight transmit–channel blocks (up 32 enabled transmit channels) can be assigned to transmit partitions A and B.

**For reception:**

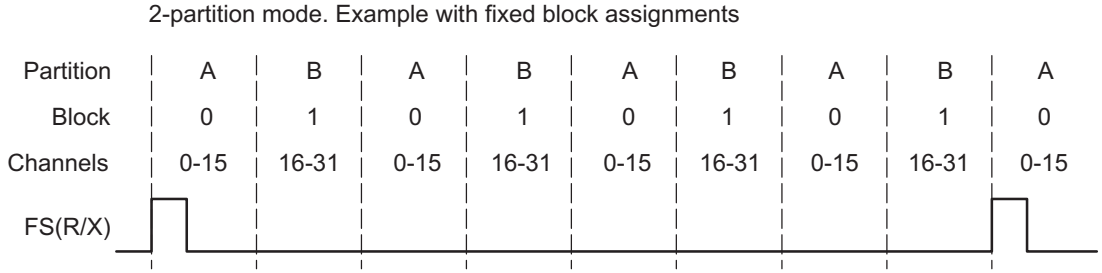
- Assign an even–numbered channel block (0, 2, 4, or 6) to receive partition A by writing to the RPABLK bits. In the receive multichannel selection mode, the channels in this partition are controlled by receive channel enable register A (RCERA_REG).
- Assign an odd–numbered block (1, 3, 5, or 7) to receive partition B with the RPBBLK bits. In the receive multichannel selection mode, the channels in this partition are controlled by receive channel enable register B (RCERB_REG).

**For transmission:**

- Assign an even–numbered channel block (0, 2, 4, or 6) to transmit partition A by writing to the XPABLK bits. In one of the transmit multichannel selection modes, the channels in this partition are controlled by transmit channel enable register A (XCERA_REG).
- Assign an odd–numbered block (1, 3, 5, or 7) to transmit partition B with the XPBBLK bits. In one of the transmit multichannel selection modes, the channels in this partition are controlled by transmit channel enable register B (XCERB_REG).

Figure 17-20 shows an example of alternating between the channels of partition A and the channels of partition B. Channels 0–15 have been assigned to partition A, and channels 16–31 have been assigned to partition B. In response to a frame–synchronization pulse, the McBSP begins a frame transfer with partition A and then alternates between partitions B and A until the complete frame is transferred.

**Figure 17-20. Alternating Between Partitions A and B Channels**




### 17.2.5.7 Transmit Multichannel Selection Modes

The MCR2_REG[1:0] register XMCM bits determine whether all channels or only selected channels are enabled and unmasked for transmission. The McBSP has three transmit multichannel selection modes (XMCM = 01b, XMCM = 10b, and XMCM = 11b), which are described in Table 17-6.

As an example of how the McBSP behaves in a transmit multichannel selection mode, suppose that XMCM = 01b (all channels disabled unless individually enabled) and that you have enabled only channels 0, 15, and 39. Suppose also that the frame length is 40. The McBSP:

- Shifts data to the McBSP.DX pin in channel 0
- Places the McBSP.DX pin in the high impedance state in channels 1–14
- Shifts data to the McBSP.DX pin in channel 15
- Places the McBSP.DX pin in the high impedance state in channels 16–38
- Shifts data to the McBSP.DX pin in channel 39



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-6. Selecting a Transmit Multichannel Selection Mode with the XMCM Bit Field**

| XMCM | Transmit Multichannel Selection Mode |
|---|---|
| 00b | No transmit multichannel selection mode is on. All channels are enabled and unmasked. No channels can be disabled or masked. |
| 01b | All channels are disabled unless they are selected in the appropriate transmit channel enable registers (XCER[A,H]_REG). If enabled, a channel in this mode is also unmasked. |
| | The MCR2_REG[9] register XMCME bit determines whether 32 channels or 128 channels are selectable in the XCER[A,H]_REG registers. |
| 10b | All channels are enabled, but they are masked unless they are selected in the appropriate transmit channel enable registers (XCER[A,H]_REG). |
| | The MCR2_REG[9] register XMCME bit determines whether 32 channels or 128 channels are selectable in the XCER[A,H]_REG registers. |
| 11b | This mode is used for symmetric transmission and reception. |
| | All channels are disabled for transmission unless they are enabled for reception in the appropriate receive channel enable registers (RCER[A,H]_REG). Once enabled, they are masked unless they are also selected in the appropriate transmit channel enable registers (XCER[A,H]_REG). |
| | The MCR2_REG[9] register XMCME bit determines whether 32 channels or 128 channels are selectable in RCER[A,H]_REG registers and XCER[A,H]_REG registers. |

#### *17.2.5.7.1 Disabling/Enabling Versus Masking/Unmasking*

For transmission, a channel can be:

- Enabled and unmasked (transmission can begin and can be completed
- Enabled but masked (transmission can begin but cannot be completed)
- Disabled (transmission cannot occur)

**Table 17-7. McBSP Channel Control Options**

| | |
|---|---|
| **Enabled channel** | A channel that can begin transmission by passing data from the data transmit register (DXR_REG) to the transmit shift register (XSR) through XB. |
| **Masked channel** | A channel that cannot complete transmission. The McBSP.DX pin is held in the high impedance state; data cannot be shifted out on the McBSP.DX pin. |
| | In systems where symmetric transmit and receive provide software benefits, this feature allows transmit channels to be disabled on a shared serial bus. A similar feature is not needed for reception because multiple receptions cannot cause serial bus contention. |
| **Disabled channel** | A channel that is not enabled. A disabled channel is also masked. |
| | Because no DXR–to–XB copy occurs, the SPCR2_REG[1] register XRDY bit is not set. Therefore, no DMA synchronization event (XEVT) is generated, and if the transmit interrupt mode depends on XRDY (XINTM = 00b in SPCR2_REG[5:4]), no interrupt is generated. |
| | The SPCR2_REG[2] register XEMPTY bit is not affected. |
| **Unmasked channel** | A channel that is not masked. Data in the XSR(s) is shifted out on the McBSP.DX pin. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.5.7.2 Activity on McBSP Pins for Different Values of XMCM Bit

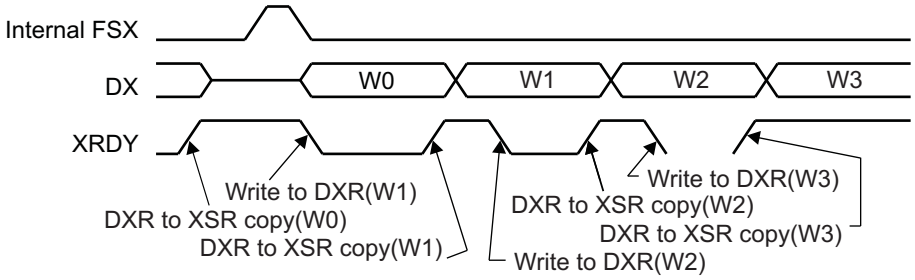
Figure 17-21 shows the activity on the McBSP pins for the various XMCM values. In all cases, the transmit frame is configured as follows:

- XPHASE = 0: Single–phase frame (required for multichannel selection modes)
- XFRLEN1 = 0000011b: 4 words per frame
- XWDLEN1 = 000b: 8 bits per word
- XMCME = 0: 2–partition mode (only partitions A and B used)

In the case where XMCM = 11b, transmission and reception are symmetric, which means the corresponding bits for the receiver (RPHASE, RFRLEN1, RWDLEN1, and RMCME) must have the same values as XPHASE, XFRLEN1, and XWDLEN1, respectively. In Figure 17-21, the arrows showing where the various events occur are only sample indications. Wherever possible, there is a time window in which these events can occur.

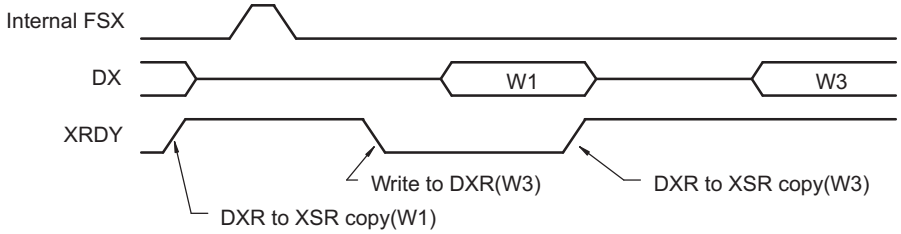
**Figure 17-21. Activity on McBSP Pins for the Possible Values of XMCM Bit**

**(a) XMCM = 00b: All channels enabled and unmasked.**




Words W0, W1, W2, W3 are written to the transmit buffer and W0, W1, W2, W3 from the transmit buffer are transferred by McBSP.DX

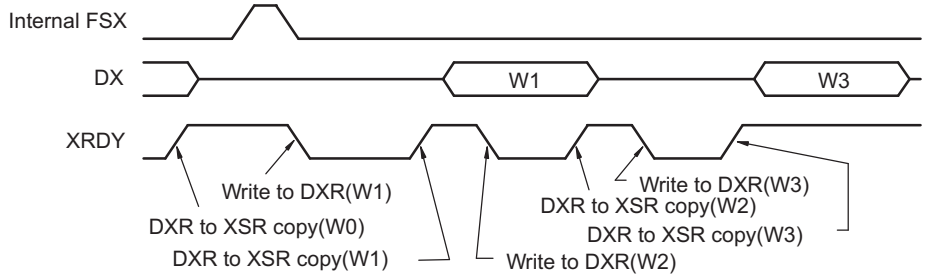
**(b) XMCM = 01b, XPABLK = 00b, XCERA = 1010b: Only channels 1 and 3 enabled and unmasked**




Words W1, W3 are written to the transmit buffer and W1, W3 from the transmit buffer are transferred by McBSP.DX


Copyright © 2011–2016, Texas Instruments Incorporated

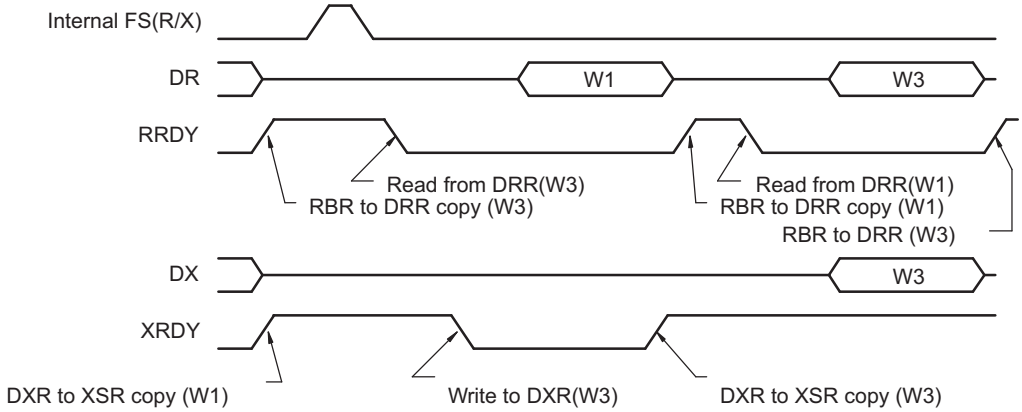
**(c) XMCM = 10b, XPABLK = 00b, XCERA = 1010b: All channels enabled, only 1 and 3 unmasked**




Words W0, W1, W2, W3 are written to the transmit buffer but only W1 and W3 from the transmit buffer are transferred by McBSP.DX

**(d) XMCM = 11b, RPABLK = 00b, XPABLK = X, RCERA = 1010b, XCERA = 1000b**

Receive channels: 1 and 3 enabled; Transmit channels: 1 and 3 enabled, but only 3 unmasked




Words W1, W3 are written to the transmit buffer but only W3 from the transmit buffer is transferred by McBSP.DX


Copyright © 2011–2016, Texas Instruments Incorporated

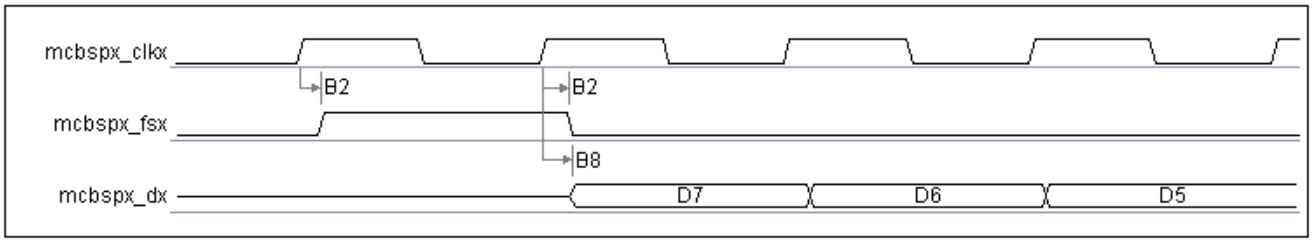
### 17.2.6 McBSP Full/Half Cycle Modes

The following figures are for data delay set to 1.

#### 17.2.6.1 Transmit Full Cycle Mode

When configured in full cycle mode (XCCR[11] register, XFULL_CYCLE bit field), the FSX is sampled on the configured CLKX edge and the data is driven on the same configured edge:

**Figure 17-22. Transmit Full Cycle Mode**

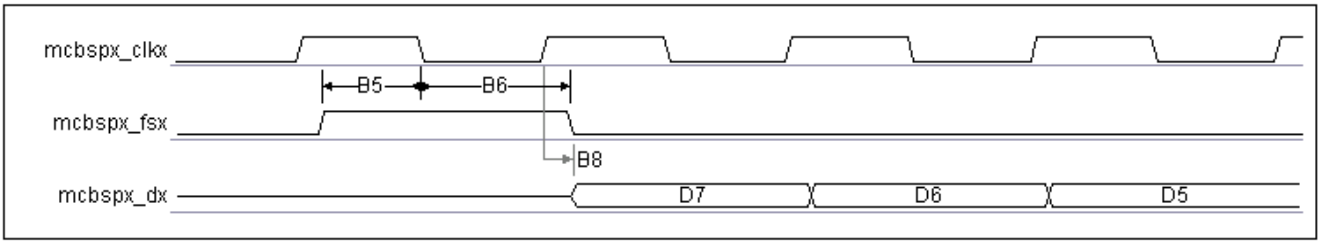



Data driven on positive CLKX edge, FSX sampled on positive CLKX edge.

#### 17.2.6.2 Transmit Half Cycle Mode

When configured in half cycle mode (XCCR[11] register, XFULL_CYCLE bit field), the FSX is sampled on the opposite configured CLKX edge and the data is driven on the next configured edge:

**Figure 17-23. Transmit Half Cycle Mode**



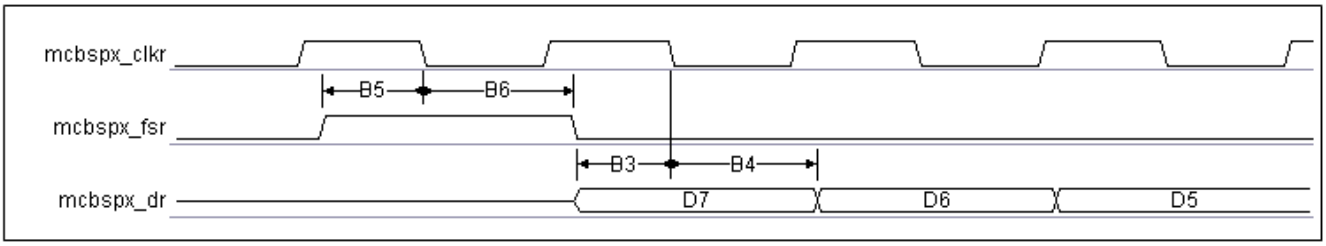

Data driven on positive CLKX edge, FSX sampled on negative CLKX edge.


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.2.6.3 Receive Full Cycle Mode

When configured in full cycle mode (RCCR[11] register, RFULL_CYCLE bit field), the FSR is sampled on the configured CLKR edge and the data is sampled on the same configured edge:

**Figure 17-24. Receive Full Cycle Mode**

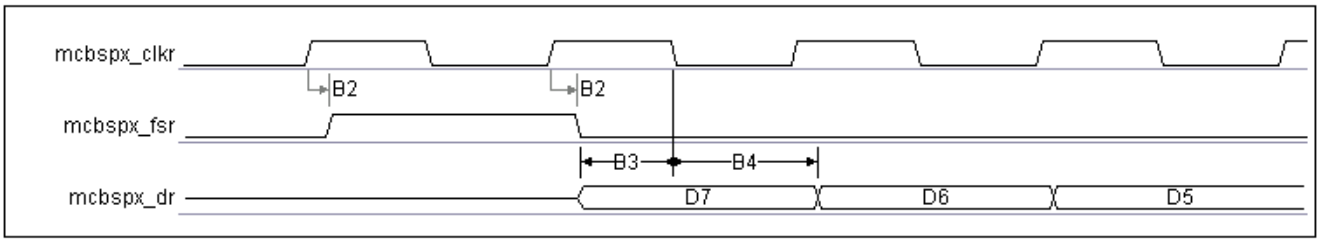



Data sampled on negative CLKR edge, FSR sampled on negative CLKR edge.

### 17.2.6.4 Receive Half Cycle Mode

When configured in half cycle mode (RCCR[11] register, RFULL_CYCLE bit field), the FSR is sampled on the opposite configured CLKR edge and the data is sampled on the next configured edge:

**Figure 17-25. Receive Half Cycle Mode**




Data sampled on negative CLKR edge, FSR sampled on positive CLKR edge.


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.2.7 Power Management

#### 17.2.7.1 ForceIdle Behavior

When configured in ForceIdle the module should respond immediately (according to the OCP guideline description) to a McBSP.MIDLEREQ, regardless of the internal state of the module. Entering in this mode, the module will freeze all the internal activity when the clocks are switched off by the power management external module. If the functional part, transmit or receive, is running within this period of time, the internal state of the module will not be idle (FSM states, processes, etc.), and when the module exits from the ForceIdle state unexpected behavior may happen in both receiver and transmitter. In order to avoid this both receive and transmit parts need to be disabled by software prior to MIdleReq assertion.

#### 17.2.7.2 SmartIdle Behavior

When configured in SmartIdle, the sources for wake-up generation are a subset of the interrupt sources. The wakeup sources are enabled by setting the corresponding bit in WAKEUPEN register.

For receive WAKEUP there are 4 possible configuration scenarios:

- **RRDYEN** - The McBSP asserts the McBSP.WAKEUP request when the RB reaches the high threshold (THRSH1_REG + 1) register value. If the corresponding bit is set in IRQENABLE register, McBSP sends an interrupt (McBSP.COMMONIRQ) request to the CPU when exiting from idle mode (interrupt will be asserted once the RRDY bit changes from 0 to 1, indicating that receive data is ready to be read).
- **REOFEN** - The McBSP asserts the McBSP.WAKEUP request at the end of the frame. If the corresponding bit is set in IRQENABLE register McBSP sends an interrupt (McBSP.COMMONIRQ) request to the CPU when exit from idle mode.
- **RSYNCERREN** - The McBSP asserts the McBSP.WAKEUP request when an unexpected receive frame-synchronization pulse is detected. If the corresponding bit is set in IRQENABLE register, McBSP sends an interrupt (McBSP.COMMONIRQ) request o the CPU when exiting from idle mode (interrupt will be asserted once the RSYNCERR bit changes from 0 to 1, indicating that a receive error occurred).

For transmit WAKEUP there are four possible configuration scenarios:

- **XRDYEN** - The McBSP asserts the McBSP.WAKEUP request when the XB reaches the high threshold (THRSH2_REG + 1) register value. If the corresponding bit is set in IRQENABLE register, McBSP sends an interrupt (McBSP.COMMONIRQ) request to the CPU when exiting from idle mode (interrupt will be asserted once the XRDY bit changes from 0 to 1, indicating that transmit buffer data is ready to accept new data).
- **XEOFEN** - The McBSP asserts the McBSP.WAKEUP request at the end of the frame. If the corresponding bit is set in IRQENABLE register, the McBSP sends an interrupt (McBSP.COMMONIRQ) request to the CPU when exiting from idle mode.
- **XFSXEN** - The McBSP sends a McBSP.WAKEUP request when a transmit frame-synchronization pulse is detected while the module is in idle mode. If the corresponding bit is set in IRQENABLE register, the McBSP sends an interrupt (McBSP.COMMONIRQ) request to the CPU when exiting from idle mode.
- **XSYNCERREN** - The McBSP asserts the McBSP.WAKEUP request when an unexpected transmit frame-synchronization pulse is detected. If the corresponding bit is set in IRQENABLE register, the McBSP sends an interrupt (McBSP.COMMONIRQ) request to the CPU when exiting from idle mode (interrupt will be asserted once the XSYNCERR bit changes from 0 to 1, indicating that a transmit error occurred).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.7.2.1 Analysis of the Receiver Smart Idle Behavior

The analysis of the power mode behavior is depicted in Table 17-8.

**Table 17-8. Analysis of the Receiver Smart Idle Behavior**

| CLKRM | CLKXM | Behavior |
|---|---|---|
| 0 | 0 | McBSP is configured as a slave, (functional clocks are provided from outside). The module will acknowledge the SmartIdle mode request as soon as there is no pending DMA, interrupt request or transmit buffer threshold synchronization (only when wake-up event is set on transmit threshold reached), regardless of the CLOCKACTIVITY settings or receive and transmit activity. |
| 0 | 1 | McBSP is configured as a transmit master (the source clock can be CLKS, CLKR or OCP).<br>When OCP clock is used as source clock and CLOCKACTIVITY is indicating that the OCP clock will be switched off, or CLKS clock is used as source clock and CLOCKACTIVITY is indicating that the functional clock is switched off, the McBSP will not acknowledge the SmartIdle request unless:<br>• the transmit part is disabled (XDISABLE) or under software reset (XRST) and the receive part is not using the transmit loop-back clock (ALBCTRLRX[0] pin is asserted).<br>• both transmit and receive parts are disabled (XDISABLE/RDISABLE) or under software reset (XRST/RRST)<br>The SmartIdle acknowledge is asserted as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached) and the pending transmit and/or receive frames where completed in case of transmit and/or receive disable.<br>When OCP or CLKS is used as source clock and the CLOCKACTIVITY indicates that the corresponding clock will not be switched off, the module will acknowledge the SmartIdle mode request as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached).<br>When CLKR is used as source (functional clock is provided from outside) then the module will acknowledge the SmartIdle mode request as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached), regardless of the CLOCKACTIVITY settings. |
| 1 | 0 | McBSP is configured as a receive master (the source clock can be CLKS, CLKX or OCP).<br>When OCP clock is used as source clock and CLOCKACTIVITY is indicating that the OCP clock will be switched off, or CLKS clock is used as source clock and CLOCKACTIVITY is indicating that the functional clock is switched off, the McBSP will not acknowledge the SmartIdle request unless:<br>• the receive part is disabled (RDISABLE) or under software reset (RRST).<br>The acknowledge is asserted as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached) and the pending transmit and/or receive frames where completed in case of transmit and/or receive disable.<br>When OCP or CLKS is used as source clock and the CLOCKACTIVITY indicates that the corresponding clock will not be switched off, the module will acknowledge the SmartIdle mode request as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached).<br>When CLKX is used as source (functional clock is provided from outside) then the module will acknowledge the SmartIdle mode request as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached), regardless of the CLOCKACTIVITY settings. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-8. Analysis of the Receiver Smart Idle Behavior (continued)**

| CLKRM | CLKXM | Behavior |
|---|---|---|
| 1 | 1 | McBSP is configured as transmit and receive master (the source clock can be CLKS or OCP).<br>When OCP clock is used as source clock and CLOCKACTIVITY is indicating that the OCP clock will be switched off, or CLKS clock is used as source clock and CLOCKACTIVITY is indicating that the functional clock is switched off, the McBSP will not acknowledge the SmartIdle request unless:<br>• both transmit and receive parts are disabled (XDISABLE/RDISABLE) or under software reset (XRST/RRST)<br>The SmartIdle acknowledge is asserted as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached) and the pending transmit and/or receive frames where completed in case of transmit and/or receive disable. Note that no wakeup event will be available in this mode since the entire McBSP and remote device activity will be frozen.<br>When OCP or CLKS is used as source clock and the CLOCKACTIVITY indicates that the corresponding clock will not be switched off, the module will acknowledge the SmartIdle mode request as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached). |

**Notes:**

- The RFSREN/XFSXEN mode is suitable for wake-up generation when both clocks (functional and OCP) are switched off and McBSP.FSR/McBSP.FSX is configured as input. The frame synchronization-pulse is asynchronously detected during idle.
- The RSYNCERREN/XSYNCERREN mode can be used to wake-up the McBSP only by a remote module implementing such a feature, to use this type of error in order to trigger a wake-up. This mode requires functional clock to be active.
- When McBSP.FSR/McBSP.FSX is configured as an output, the McBSP.FSR/ McBSP.FSX wake-up generation makes no sense (the module cannot be in SmartIdle mode).
- Detection of RSYNCERR/XSYNCERR during idle mode can be used only when McBSP.FSR/ McBSP.FSX is configured as an input and the remote system knows to assert such an error in order to trigger the wake-up of the McBSP.
- The module does not implement IRQ assertion when configured as GPIO; also a wake-up capability in this mode is not available.

## *17.2.8 Programming Model*

### 17.2.8.1 McBSP Initialization Procedure

The serial port initialization procedure is as follows:

1. Clear SPCR1_REG[0] register RRST bit, SPCR2_REG[7] register FRST bit, and SPCR2_REG[0] register XRST bit to 0. If coming out of a global reset, this step is not required.
2. While the serial port is in the reset state, program only the McBSP configuration registers (not the data registers) as required.
3. Wait for two clock cycles. This ensures proper internal synchronization
4. Set SPCR1_REG[0] register RRST bit and SPCR2_REG[0] register XRST bit to 1 to enable the serial port. Make sure that as you set these reset bits, you do not modify any of the other bits in SPCR1_REG and SPCR2_REG registers. Otherwise, you would change the configuration set in step 2.
5. Set up data acquisition as required (such as writing to DXR_REG register).
6. Set SPCR2_REG[7] register FRST bit to 1 if internally generated frame synchronization is required.
7. Wait for two clock cycles for the receiver and transmitter to become active.


Copyright © 2011–2016, Texas Instruments Incorporated

Alternatively, on write (step 1 or 5), the transmitter and receiver can be placed in or taken out of reset by modifying the desired bit.

The previous procedure for reset/initialization can be applied in general when the receiver or transmitter has to be reset during its normal operation, and also when the sample rate generator is not used for either operation.

**Notes:**

- The necessary duration of the active-low period of XRST or RRST is at least two CLKR/CLKX cycles.
- The appropriate bits in serial port configuration registers (SPCR1_REG, PCR_REG, RCR1_REG, RCR2_REG, XCR1_REG, XCR2_REG, SRGR1_REG and SRGR2_REG) should only be modified when the affected portion of the serial port is in its reset state.
- In most cases, the data transmit register (DXR_REG) should be loaded by the CPU or the DMA controller only when the transmitter is enabled (XRST = 1). An exception to this rule is when these registers are used for loop back internal data.
- The bits of the channel control registers (MCR1_REG, MCR2_REG, RCER{A-H}_REG and XCER{A-H}_REG) can be modified at any time as long as they are not being used by the current reception/transmission in a multichannel selection mode
- The sample rate generator is reset by clearing SPCR2_REG[6] register GRST bit to 0.

#### 17.2.8.2 Reset and Initialization Procedure for the Sample Rate Generator

To reset and initialize the sample rate generator:

1. Place the McBSP/sample rate generator in reset. During a Global reset, the sample rate generator, the receiver, and the transmitter reset bits (GRST, RRST, and XRST) are automatically forced to 0. Otherwise, during normal operation, the sample rate generator can be reset by making GRST = 0 in SPCR2_REG[6] register, provided that CLKG and/or FSG is not used by any portion of the McBSP. Depending on your system you may also want to reset the receiver (RRST = 0 in SPCR1_REG[0]) and reset the transmitter (SPCR2_REG[0] register XRST bit = 0).
2. Program the registers that affect the sample rate generator. Program the sample rate generator registers (SPCR1_REG and SPCR2_REG) as required for your application. If necessary, other control registers can be loaded with desired values provided the respective portion of the McBSP (the receiver or transmitter) is in reset. After the sample rate generator registers are programmed, wait 2 CLKSRG cycles. This ensures proper synchronization internally.
3. Enable the sample rate generator (take it out of reset). In SPCR2_REG[6], set GRST bit to 1 to enable the sample rate generator. After the sample rate generator is enabled, wait two CLKG cycles for the sample rate generator logic to stabilize. On the next rising edge of CLKSRG, CLKG transitions to 1 and starts clocking with a frequency equal to (input clock frequency)/(CLKGDV + 1) where the input clock is selected with the SCLKME bit of PCR_REG[7] register and the SRGR2_REG[13] register CLKSM bit in one of the following configurations:

**Table 17-9. Input Clock Selection for Sample Rate Generator**

| SCLKME bit | CLKSM bit | Input Clock for Sample Rate Generator |
|---|---|---|
| 0 | 0 | Signal on McBSP.CLKS pin |
| 0 | 1 | McBSP_FCLK clock |
| 1 | 0 | Signal on McBSP.CLKR pin |
| 1 | 1 | Signal on McBSP.CLKX pin |

4. If necessary, enable the receiver and/or the transmitter. If necessary, remove the receiver and/or transmitter from reset by setting RRST and/or XRST = 1.
5. Necessary, enable the frame–synchronization logic of the sample rate generator. After the required data acquisition setup is done (DXR_REG is loaded with data), set GRST = 1 in SPCR2_REG[6] if an internally generated frame–synchronization pulse is required. FSG is generated with an active–high edge after the programmed number of CLKG clocks (FPER + 1) has elapsed.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.3 Data Transfer DMA Request Configuration

To configure the McBSP receive/transmit data DMA requests, perform the following procedure:

1. Write the receive THRSH1_REG register with the required receive DMA request length (the length of the transfer is the same as the threshold value + 1). As long as the RB occupied locations level is above or equal to the THRSH1_REG value + 1, the DMA request will be asserted. After transferring the configured THRSH1_REG + 1 number of words, the receive DMA request (McBSP.REVNT) will be deasserted and reasserted as soon as the conditions are met again.
   Note that in case of a number of transfers that exceed the number of the programmed DMA length the McBSP will respond to the command, and will perform the transfer regardless of the receive buffer empty condition. When the receive buffer is empty a data transfer access will trigger a receive underflow interrupt, if enabled by RUNDFL_EN bit in IRQENSTAT_REG register.
2. Write the transmit THRSH2_REG register with the required transmit DMA request length (the length of the transfer is the same as the threshold value + 1). As long as the XB free locations level is above or equal to the THRSH2_REG value + 1, the DMA request will be asserted. After transferring the configured THRSH2_REG + 1 number of words, the transmit DMA request (McBSP.XEVNT) will be deasserted and reasserted as soon as the conditions are met again.
   Note that in case of a number of transfers that exceed the number of the programmed DMA length the McBSP will respond to the command, and will perform the transfer regardless of the transmit buffer full condition. When the transmit buffer is full a data transfer access will trigger a transmit overflow interrupt, if enabled by XOVFL_EN bit in IRQENSTAT_REG register

#### 17.2.8.4 Interrupt Configuration

The McBSP offers two interrupt schemes:

- OCP compliant interrupt request scheme using a common receive/transmit interrupt request line.
- The legacy interrupt compliant scheme using 3 interrupt lines: one for receive, one for transmit and the receive overflow interrupt line.

The OCP compliant interrupt line can be configured by using the IRQENABLE register. When the IRQSTATUS bit is set and the corresponding IRQENABLE bit is set to 1, the interrupt line is asserted. Writing one to a bit in IRQSTATUS register will clear the bit.

There are several conditions, which may be configured to generate an interrupt as follows:

- Transmit Buffer Overflow (XOVFLSTAT bit is set to 1 when transmit buffer overflow; the data which is written while overflow condition is discarded).
- Transmit Buffer Underflow (XUNDFLSTAT bit is set to 1 when the transmit data buffer is empty new data is required to be transmitted).
- Transmit Buffer Threshold Reached (XRDY bit is set to 1 when the transmit buffer free locations are equal or above the (THRSH2_REG + 1) value).
- Transmit End of Frame (XEOF is set to 1 when a complete frame was transmitted).
- Transmit Frame Synchronization (XFSX bit is set to 1 when a new transmit frame synchronization is asserted).
- Transmit Frame Synchronization Error (XSYNCERR is set to 1 when a transmit frame synchronization error is detected).
- Receive Buffer Overflow (ROVFLSTAT bit is set to 1 when receive buffer overflow; the data which is written while overflow condition is discarded).
- Receive Buffer Underflow (RUNDFLSTAT bit is set to 1 when read operation is performed to the receive data register while receive buffer is empty; data read while underflow condition is undefined).
- Receive Buffer Threshold Reached (RRDY bit is set to 1 when the receive buffer occupied locations are equal or above the (THRSH1_REG + 1) value).
- Receive End of Frame (REOF is set to 1 when a complete frame was received).
- Receive Frame Synchronization (RFSR bit is set to 1 when a new receive frame synchronization is asserted).
- Receive Frame Synchronization Error (RSYNCERR is set to 1 when a receive frame synchronization error is detected).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.5 Receiver Configuration

To configure the McBSP receiver, perform the following procedure:

1. Place the McBSP/receiver in reset.
2. Program the McBSP registers for the desired receiver operation.
3. Take the receiver out of reset.

##### *17.2.8.5.1 Programming the McBSP Registers for the Desired Receiver Configuration*

The following is a list of important tasks to be performed when you are configuring the McBSP receiver. Each task corresponds to one or more McBSP register bit fields. Note that SRG is Sample Rate Generator.

**Global Behavior**

1. Set the receiver pins to operate as McBSP pins.
2. Enable/disable the digital loop back mode.
3. Enable/disable the analog loop back mode.
4. Enable/disable the synchronous transmit-receive mode.
5. Enable/disable the receive multichannel selection mode.

**Data behavior:**

1. Choose 1 or 2 phases for the receive frame.
2. Set the receive word length(s).
3. Set the receive frame length (one word per phase if dual-phase selected).
4. Enable/disable the receive frame–synchronization ignore function.
5. Set the receive data delay.
6. Set the receive sign–extension and justification mode.
7. Set the receive interrupt mode.

**Frame–synchronization behavior:**

1. Set the receive frame–synchronization mode.
2. Set the receive frame–synchronization polarity.
3. Set the sample rate generator (SRG) frame–synchronization period and pulse width.

**Clock behavior:**

1. Set the receive clock mode.
2. Set the receive clock polarity.
3. Set the SRG clock divide–down value.
4. Set the SRG clock synchronization mode.
5. Set the SRG clock mode (choose an input clock).
6. Set the SRG input clock polarity.

##### *17.2.8.5.2 Resetting and Enabling the Receiver*

The first step of the receiver configuration procedure is to reset the receiver, and the last step is to enable the receiver (to take it out of reset).

The serial port can be reset in the following two ways:

- A global reset places the receiver, transmitter, and sample rate generator in reset. When the device reset is removed, GRST, FRST, RRST and XRST bits = 0, keeping the entire serial port in the reset state.
- The serial port transmitter and receiver can be reset directly using the RRST and XRST bits in the SPCR1_REG register. The sample rate generator can be reset directly using the GRST bit in SPCR2_REG[6] register.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.5.3 Set the Receiver Pins to Operate as McBSP Pins

The RIOEN bit (PCR_REG[12]) determines whether the receiver pins are McBSP pins or general–purpose I/O pins.

Refer to Section 17.2.8.7 for further information.

#### 17.2.8.5.4 Enable/Disable the Synchronous Transmit-Receive Mode

The ALBCTRLRX input pin is used to configure the synchronous transmit-receive mode. The ALBCTRLRX[0/1] input pin selection is determined by programming the Control Module AUD_CTRL Register.

The ALBCTRLRX [0] controls the multiplexer which connects the functional receive input clock (CLKR is connected to the CLKX input pin CLKXI when ALBCTRLRX[0] = 1).

The ALBCTRLRX[1] controls the multiplexer which connects the receive frame synchronization (FSR is connected to the FSX input pin FSXI when ALBCTRLRX[1] = 1).

#### 17.2.8.5.5 Enable/Disable the Analog Loop Back Mode

The ALB bit (SPCR1_REG[15]) determines whether the analog loop-back mode is on or off.

In the analog loop-back mode, the receive signals are connected internally through multiplexers to the corresponding transmit loop back signals (DR is connected to transmit loop back data on DXI pin, FSR is connected to FRX input pin FSXI, and CLKR is connected to the CLKX input pin CLKXI). This mode allows testing of serial port; the McBSP receives the data it transmits. This loop back mode is done through pads testing also the IO buffers.

#### 17.2.8.5.6 Enable/Disable the Digital Loop Back Mode

The DLB bit (XCCR_REG) determines whether the digital loop-back mode is on or off. In the digital loop-back mode, the receive signals are connected internally through multiplexers to the corresponding transmit signals (DR is connected to transmit data on DX output pin DXO, FSR is connected to FRX output pin FSXO, and CLKR is connected to the CLKX output pin CLKXO). This mode allows testing of serial port; the McBSP receives the data it transmits. This loop back mode is not done through pads, all output signals being disabled (CLKREN, CLKXEN, FSREN, FSRXEN, DXEN are not active).

Note that in digital loop back mode the sample rate generator and the frame synchronization generator need to be enabled in order to generate the CLKX and FSX signals.

#### 17.2.8.5.7 Enable/Disable the Receive Multichannel Selection Mode

The RMCM bit (MCR1_REG[0]) determines whether the receive multichannel selection mode is on or off.

#### 17.2.8.5.8 Choose 1 or 2 Phases for the Receive Frame

The RPHASE bit (RCR2_REG[15]) determines whether the receive data frame has one or two phases. When dual-phase is selected the number of words per phase must be set to 1.

#### 17.2.8.5.9 Set the Receive Word Length(s)

The RWDLEN1 (RCR1_REG[7:5]) and RWDLEN2 (RCR2_REG[7:5]) bit fields determine how many bits are in each serial word in phase 1 and in phase 2, respectively, of the receive data frame.

Each frame can have one or two phases, depending on the value that you load into the RPHASE bit. If a single–phase frame is selected, RWDLEN1 selects the length for every serial word received in the frame. If a dual–phase frame is selected, RWDLEN1 and RWDLEN2 must be cleared to 0 (one word per phase).

#### 17.2.8.5.10 Set the Receive Frame Length

The RFRLEN1 (RCR1_REG[14:8]) and RFRLEN2 (RCR2_REG[14:8]) bit fields determine how many serial words are in phase 1 and in phase 2, respectively, of the receive data frame.


Copyright © 2011–2016, Texas Instruments Incorporated

The receive frame length is the number of serial words in the receive frame. Each frame can have one or two phases, depending on value that you load into the RPHASE bit.

If a single–phase frame is selected (RPHASE = 0), the frame length is equal to the length of phase 1. If a dual–phase frame is selected (RPHASE = 1), the frame length is the length is two (one word for phase 1 plus the one word for phase 2).

The 7–bit RFRLEN1 field allow up to 128 words per phase when single-phase frame. See Table 17-10 for a summary of how to calculate the frame length. This length corresponds to the number of words or logical time slots or channels per frame–synchronization pulse.

Program the RFRLEN fields with [w minus 1], where w represents the number of words per phase. For the example, if you want a phase length of 128 words in phase 1, load 127 into RFRLEN1.

**Table 17-10. How to Calculate the Length of the Receive Frame**

| RPHASE | RFRLEN1 | RFRLEN2 | Frame Length |
|---|---|---|---|
| 0 | 0 <= RFRLEN1 <= 127 | Don't care | (RFRLEN1 + 1) words |
| 1 | RFRLEN1 = 0 | RFRLEN2 = 0 | 2 words |

#### *17.2.8.5.11 Set the Receive Reverse Mode*

The RREVERSE bit field (RCR2_REG[4:3]) determines whether reverse (LSB first) data transfer option is chosen for McBSP reception.

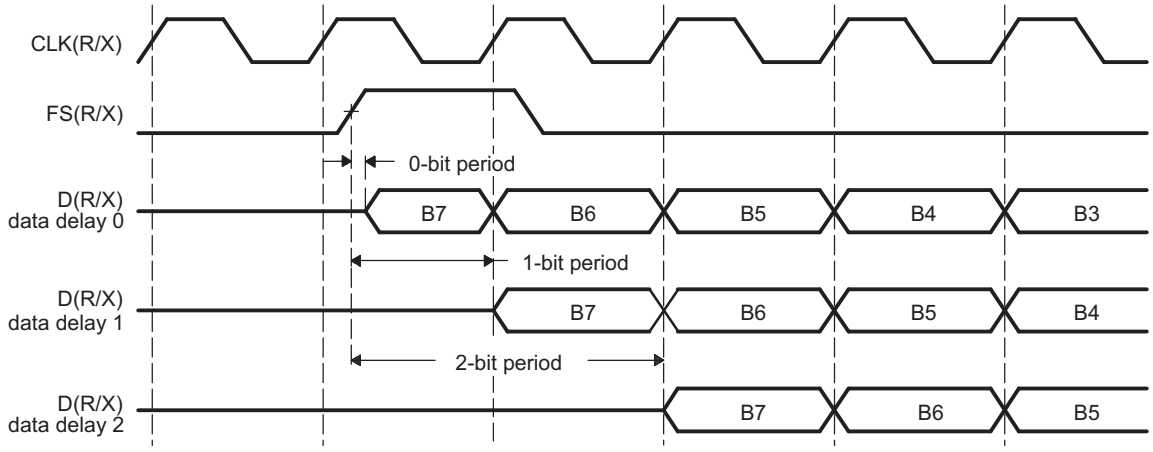
#### *17.2.8.5.12 Set the Receive Data Delay*

The RDATDLY bit field (RCR2_REG[1:0]) determines the length of the data delay for the receive frame.

The start of a frame is defined by the first clock cycle in which frame synchronization is found to be active. The beginning of actual data reception or transmission with respect to the start of the frame can be delayed if required. This delay is called data delay.

RDATDLY specifies the data delay for reception. The range of programmable data delay is 0 to 2 bit–clocks (RDATDLY = 00b–10b), as shown in Figure 17-26. In this figure, the data transferred is an 8–bit value with bits labeled B7, B6, B5, and so on. Typically a 1–bit delay is selected, because data often follows a 1–cycle active frame–synchronization pulse.

**Figure 17-26. Range of Programmable Data Delay**





Copyright © 2011–2016, Texas Instruments Incorporated

**0-bit data delay:**

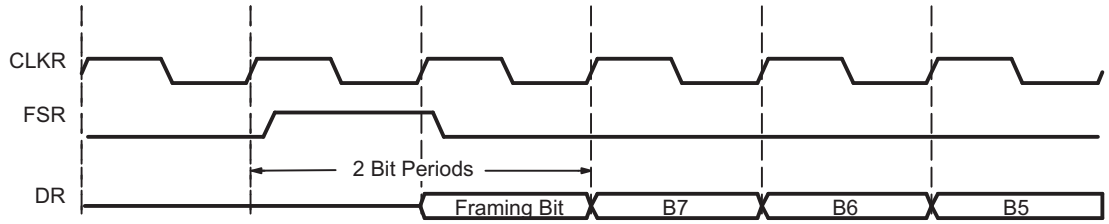
Normally, a frame–synchronization pulse is detected or sampled with respect to an edge of internal serial clock CLK(R/X). Thus, on the following cycle or later (depending on the data delay value), data may be received or transmitted. However, in the case of 0–bit data delay, the data must be ready for reception and/or transmission on the same serial clock cycle.

For reception, this problem is solved because receive data is sampled on the first falling edge of CLKR where an active–high internal FSR is detected. However, data transmission must begin on the rising edge of the internal CLKX clock that generated the frame synchronization. Therefore, the first data bit is assumed to be present in XSR1, and thus on McBSP.DX. The transmitter then asynchronously detects the frame–synchronization signal (FSX) going active high and immediately starts driving the first bit to be transmitted on the McBSP.DX pin.

**2-bit delay:**

A data delay of two bit periods allows the serial port to interface to different types of T1 framing devices where the data stream is preceded by a framing bit. During reception of such a stream with data delay of two bits (framing bit appears after a 1–bit delay and data appears after a 2–bit delay), the serial port essentially discards the framing bit from the data stream, as shown in Figure 17-27. In this figure, the data transferred is an 8–bit value with bits labeled B7, B6, B5, and so on.

**Figure 17-27. 2-Bit Data Delay Used to Skip a Framing Bit**




### *17.2.8.5.13 Set the Receive Sign-Extension and Justification Mode*

The RJUST bit field (SPCR1_REG[14:13]) determines whether data received by the McBSP is sign–extended or not and how it is justified.

RJUST bit field selects whether data in RB is right– or left–justified (with respect to the MSB) in DRR_REG register and whether unused bits in DRR_REG are filled with zeroes or with sign bits.

The following tables show the effects of various RJUST values. Table 17-11 shows the effect on an example 12–bit receive–data value ABCh, Table 17-12 shows the effect on an example 20–bit receive–data value ABCDEh.

**Table 17-11. Example: Use of RJUST Bit Field with 12-Bit Data Value ABCh**

| RJUST | Justification | Extension | Value in DRR_REG |
|---|---|---|---|
| 00b | Right | Zero fill MSBs | 0000 0ABCh |
| 01b | Right | Sign extend data into MSBs | FFFF FABCh |
| 10b | Left | Zero fill LSBs | ABC0 0000h |
| 11b | Reserved | Reserved | Reserved |

**Table 17-12. Example: Use of RJUST Bit Field with 20-Bit Data Value ABCDE**

| RJUST | Justification | Extension | Value in DRR_REG |
|---|---|---|---|
| 00b | Right | Zero fill MSBs | 000A BCDEh |
| 01b | Right | Sign extend data into MSBs | FFFA BCDEh |
| 10b | Left | Zero fill LSBs | ABCD E000h |
| 11b | Reserved | Reserved | Reserved |



Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.5.14 Set the Receive Interrupt Mode (legacy only)

The RINTM bit field (SPCR1_REG[5:4]) determines which event generates a receive interrupt request to the CPU. The receive interrupt (RINT) informs the CPU of changes to the serial port status. Four options exist for configuring this interrupt.

- RINTM = 00: RINT generated when the RRDY bit (SPCR1_REG[1]) changes from 0 to 1. Interrupt on every serial word by tracking the RRDY bit in the SPCR1_REG register. Regardless of the value of RINTM, RRDY can be read to detect the RRDY = 1 condition.
- RINTM = 01: RINT generated by an end–of–frame condition in the receive multichannel selection mode. In any other serial transfer case, this setting is not applicable and, therefore, no interrupts are generated.
- RINTM = 10: RINT generated by a new receive frame–synchronization pulse. Interrupt on detection of receive frame–synchronization pulses. This generates an interrupt even when the receiver is in its reset state. This is done by synchronizing the incoming frame–synchronization pulse to the CPU clock and sending it to the CPU via RINT.
- RINTM = 11: RINT generated when RSYNCERR is set. Interrupt on frame–synchronization error. Regardless of the value of RINTM, RSYNCERR can be read to detect this condition. For information on using RSYNCERR, see Section 17.2.3.2.

The McBSP provides also a common interrupt line McBSP.COMMONIRQ, which can be used by setting the IRQENABLE register. All the above settings have equivalent enable bits in the IRQENABLE register in order to enable the common interrupt line (RRDYEN is equivalent with RINTM = 00, REOFEN is equivalent with RINTM = 01 (the interrupt is generated by an end-of-frame condition regardless of the multichannel selection mode), RFSREN is equivalent with RINTM = 10 and RSYNCERREN is equivalent with RINTM = 11 setting). This interrupt line has its own status register.

#### 17.2.8.5.15 Set the Receive Frame-Synchronization Mode

FSRM bit (PCR_REG[10]), GSYNC bit (SRGR2_REG[15]), ALB bit (SPCR1_REG[15]) and DLB bit field (XCCR_REG) are used to determine the source for receive frame synchronization and the function of the McBSP.FSR pin.

Table 17-13 shows how you can select various sources to provide the receive frame–synchronization signal and the effect on the McBSP.FSR pin. The polarity of the signal on the McBSP.FSR pin is determined by the FSRP bit.

In digital loop-back mode (DLB = 1), the transmit frame–synchronization signal is used as the receive frame–synchronization signal.

Also in the analog loop back mode, the internal receive clock signal (CLKR), the internal receive frame–synchronization signal (FSR) are internally connected to their transmit counterparts, CLKX and FSX.

In the digital loop-back mode (DLB = 1) or analog loop-back mode (ALB = 1), the transmit frame synchronization signal is used as the receive frame synchronization signal.

**Table 17-13. FSRM and GSYNC Effects on Frame-Synchronization Signal and McBSP.FSR Pin**

| FSRM | GSYNC | Source of Receive Frame Synchronization | McBSP.FSR Pin Status |
|---|---|---|---|
| 0 | 0 or 1 | An external frame synchronization signal enters the McBSP through the McBSP.FSR pin. The signal is then inverted as determined by FSRP before being used as internal FSR. | Input |
| 1 | 0 | Internal FSR is driven by the sample rate generator frame–synchronization signal (FSG). | Output. FSG is inverted as determined by FSRP before being driven out on the McBSP.FSR pin. |
| 1 | 1 | Internal FSR is driven by the sample rate generator frame–synchronization signal (FSG). | Input. The external frame–synchronization input on the McBSP.FSR pin is used to synchronize CLKG and generate FSG pulses. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.5.16 Set the Receive Frame-Synchronization Polarity

The FSRP bit (PCR_REG[2]) determines whether frame–synchronization pulses are active high or active low on the McBSP.FSR pin.

Receive frame–synchronization pulses can be generated internally by the sample rate generator or driven by an external source. The source of frame synchronization is selected by programming the PCR_REG[10] register FSRM mode bit. FSR is also affected by the SRGR2_REG[15] register GSYNC bit. For information about the effects of FSRM and GSYNC, see the Frame Synchronization Generation in the Sample Rate Generator section. Similarly, receive clocks can be selected to be inputs or outputs by programming the PCR_REG[8] register CLKRM mode bit (see the Clock Generation in the Sample Rate Generator section).

When FSR and FSX are inputs (FSXM = FSRM= 0, external frame–synchronization pulses), the McBSP detects them on the internal falling edge of clock, internal CLKR, and internal CLKX, respectively. The receive data arriving at the McBSP.DR pin is also sampled on the falling edge of internal CLKR. These internal clock signals are either derived from an external source via CLK(R/X) pins or driven by the sample rate generator clock (CLKG) internal to the McBSP.

When FSR and FSX are outputs, implying that they are driven by the sample rate generator, they are generated (transition to their active state) on the rising edge of the internal clock, CLK(R/X). Similarly, data on the McBSP.DX pin is output on the rising edge of internal CLKX.

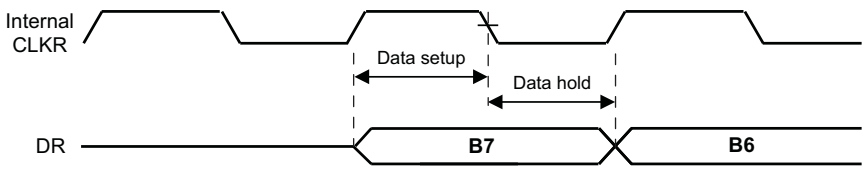
FSRP, FSXP, CLKRP, and CLKXP bits in the pin control register (PCR_REG) configure the polarities of the FSR, FSX, CLKR, and CLKX signals, respectively. All frame–synchronization signals (internal FSR, internal FSX) that are internal to the serial port are active high. If the serial port is configured for external frame synchronization (FSR/FSX are inputs to McBSP), and FSRP = FSXP = 1, the external active–low frame–synchronization signals are inverted before being sent to the receiver (internal FSR) and transmitter (internal FSX). Similarly, if internal synchronization (FSR/FSX are output pins and GSYNC = 0) is selected, the internal active–high frame–synchronization signals are inverted, if the polarity bit FS(R/X)P = 1, before being sent to the FS(R/X) pin.

On the transmit side, the transmit clock polarity bit, CLKXP, sets the edge used to shift and clock out transmit data. Data is always transmitted on the rising edge of internal CLKX. If CLKXP = 1 and external clocking is selected (CLKXM = 0 and CLKX is an input), the external falling–edge triggered input clock on CLKX is inverted to a rising–edge triggered clock before being sent to the transmitter. If CLKXP = 1, and internal clocking selected (CLKXM = 1 and CLKX is an output pin), the internal (rising–edge triggered) clock, internal CLKX, is inverted before being sent out on the McBSP.CLKX pin.

Similarly, the receiver can reliably sample data that is clocked with a rising edge clock (by the transmitter). The receive clock polarity bit, CLKRP, sets the edge used to sample received data. The receive data is always sampled on the falling edge of internal CLKR. Therefore, if CLKRP = 1 and external clocking is selected (CLKRM = 0 and CLKR is an input pin), the external rising–edge triggered input clock on CLKR is inverted to a falling–edge triggered clock before being sent to the receiver. If CLKRP = 1 and internal clocking is selected (CLKRM = 1), the internal falling–edge triggered clock is inverted to a rising–edge triggered clock before being sent out on the McBSP.CLKR pin.

Note that CLKRP = CLKXP in a system where the same clock (internal or external) is used to clock the receiver and transmitter. The receiver uses the opposite edge as the transmitter to ensure valid setup and hold of data around this edge. Figure 17-28 shows how data clocked by an external serial device using a rising edge can be sampled by the McBSP receiver on the falling edge of the same clock.

**Figure 17-28. Data Externally Clocked on a Rising Edge and Sampled on a Falling Edge**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.5.17 Set the SRG Frame-Synchronization Period and Pulse Width

FPER bit field (SRGR2_REG[11:0]) is used to set the SRG frame-synchronization period and FWID bit field (SRGR1_REG[15:8]) is used to set the SRG pulse width. The sample rate generator can produce a clock signal, CLKG, and a frame–synchronization signal, FSG. If the sample rate generator is supplying receive or transmit frame synchronization, you must program the bit fields FPER and FWID.

On FSG, the period from the start of a frame–synchronization pulse to the start of the next pulse is (FPER + 1) CLKG cycles. The 12 bits of FPER allow a frame–synchronization period of 1 to 4096 CLKG cycles, which allows up to 4096 data bits per frame. When GSYNC = 1, FPER is a don't care value.

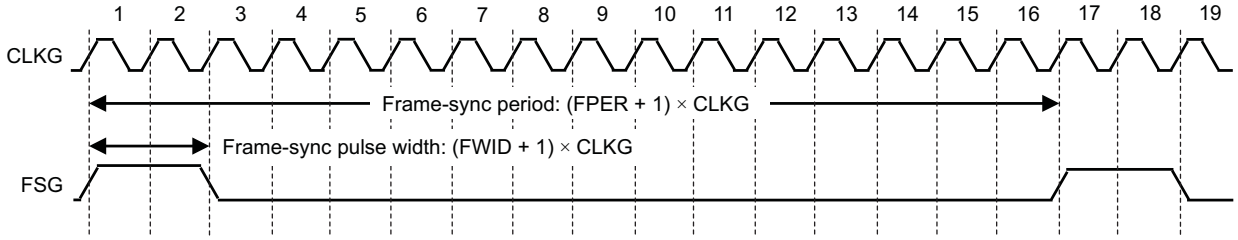
Each pulse on FSG has a width of (FWID + 1) CLKG cycles. The eight bits of FWID allow a pulse width of 1 to 256 CLKG cycles. It is recommended that FWID be programmed to a value less than the programmed word length.

The values in FPER and FWID are loaded into separate down–counters. The 12–bit FPER counter counts down the generated clock cycles from the programmed value (4095 maximum) to 0. The 8–bit FWID counter counts down from the programmed value (255 maximum) to 0.

Figure 17-29 shows a frame–synchronization period of 16 CLKG periods (FPER = 15 or 0000 1111b) and a frame–synchronization pulse with an active width of 2 CLKG periods (FWID = 1).

When the sample rate generator comes out of reset, FSG is in its inactive state. Then, when GRST = 1 and FSGM = 1, a frame–synchronization pulse is generated. The frame width value (FWID + 1) is counted down on every CLKG cycle until it reaches 0, at which time FSG goes low. At the same time, the frame period value (FPER + 1) is also counting down. When this value reaches 0, FSG goes high, indicating a new frame.

**Figure 17-29. Frame of 16 CLKG Periods and Active Width of 2 CLKG Periods**




#### 17.2.8.5.18 Set the Receive Clock Mode

CLKRM bit (PCR_REG[8]), ALB bit (SPCR1_REG[15]) and DLB bit field (XCCR_REG) are used to set the receive clock mode.

Table 17-14 shows how you can select various sources to provide the receive clock signal and affect the McBSP.CLKR pin. The CLKRP bit determines the polarity of the signal on the McBSP.CLKR pin.

In the digital loop-back mode (DLB = 1) or analog loop-back mode (ALB = 1), the transmit clock signal is used as the receive clock signal.

**Table 17-14. CLKRM Effect on Receive Clock Signal and McBSP.CLKR Pin**

| CLKRM | Source of Receive Clock | McBSP.CLKR Pin Status |
|---|---|---|
| 0 | The McBSP.CLKR pin is an input driven by an external clock. The external clock signal is inverted as determined by CLKRP before being used. | Input |
| 1 | The sample rate generator clock (CLKG) drives internal CLKR. | Output. CLKG, inverted as determined by CLKRP, is driven out on the McBSP.CLKR pin. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.5.19 Set the Receive Clock Polarity

CLKRP bit (PCR_REG[0]) is used to set the receive clock polarity.

Receive frame–synchronization pulses can be generated internally by the sample rate generator or driven by an external source. The source of frame synchronization is selected by programming the PCR_REG[10] register FSRM mode bit. FSR is also affected by the GSYNC bit in SRGR2_REG[15]. Similarly, receive clocks can be selected to be inputs or outputs by programming the PCR_REG[8] register CLKRM mode bit.

When FSR and FSX are inputs (FSXM = FSRM= 0, external frame–synchronization pulses), the McBSP detects them on the internal falling edge of clock, internal CLKR, and internal CLKX, respectively. The receive data arriving at the McBSP.DR pin is also sampled on the falling edge of internal CLKR. These internal clock signals are either derived from external source via CLK(R/X) pins or driven by the sample rate generator clock (CLKG) internal to the McBSP.

When FSR and FSX are outputs, implying that they are driven by the sample rate generator, they are generated (transition to their active state) on the rising edge of internal clock, CLK(R/X). Similarly, data on the McBSP.DX pin is output on the rising edge of internal CLKX. FSRP, FSXP, CLKRP, and CLKXP in the pin control register (PCR_REG) configure the polarities of the FSR, FSX, CLKR, and CLKX signals, respectively. All frame–synchronization signals (internal FSR, internal FSX) that are internal to the serial port are active high. If the serial port is configured for external frame synchronization (FSR/FSX are inputs to McBSP) and FSRP = FSXP = 1, the external active–low frame–synchronization signals are inverted before being sent to the receiver (internal FSR) and transmitter (internal FSX). Similarly, if internal synchronization (FSR/FSX are output pins and GSYNC = 0) is selected, the internal active–high frame–synchronization signals are inverted, if the polarity bit FS(R/X)P = 1, before being sent to the FS(R/X) pin.

On the transmit side, the transmit clock polarity bit, CLKXP, sets the edge used to shift and clock out transmit data. Data is always transmitted on the rising edge of internal CLKX. If CLKXP = 1 and external clocking is selected (CLKXM = 0 and CLKX is an input), the external falling–edge triggered input clock on CLKX is inverted to a rising–edge triggered clock before being sent to the transmitter. If CLKXP = 1 and internal clocking is selected (CLKXM = 1 and CLKX is an output pin), the internal (rising–edge triggered) clock, internal CLKX, is inverted before being sent out on the McBSP.CLKX pin.

Similarly, the receiver can reliably sample data that is clocked with a rising edge clock (by the transmitter). The receive clock polarity bit, CLKRP, sets the edge used to sample received data. The receive data is always sampled on the falling edge of internal CLKR. Therefore, if CLKRP = 1 and external clocking is selected (CLKRM = 0 and CLKR is an input pin), the external rising–edge triggered input clock on CLKR is inverted to a falling–edge triggered clock before being sent to the receiver. If CLKRP = 1 and internal clocking is selected (CLKRM = 1), the internal falling–edge triggered clock is inverted to a rising–edge triggered clock before being sent out on the McBSP.CLKR pin.

Note that CLKRP = CLKXP in a system where the same clock (internal or external) is used to clock the receiver and transmitter. The receiver uses the opposite edge as the transmitter to ensure valid setup and hold of data around this edge.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.5.20 Set the SRG Clock Divide-Down Value

CLKGDV bit field (SRGR1_REG[7:0]) is used to set the sample rate generator clock divide-down value.

The first divider stage generates the serial data bit clock from the input clock. This divider stage utilizes a counter, preloaded by CLKGDV, that contains the divide ratio value.

The output of the first divider stage is the data bit clock, which is output as CLKG and which serves as the input for the second and third stages of the divider.

CLKG has a frequency equal to 1/(CLKGDV + 1) of sample rate generator input clock. Thus, the sample generator input clock frequency is divided by a value between 1 and 256. The CLKG duty cycle is 50%.

#### 17.2.8.5.21 Set the SRG Clock Synchronization Mode

GSYNC bit (SRGR2_REG[15]) is used to set the SRG clock synchronization mode.

#### 17.2.8.5.22 Set the SRG Clock Mode (choose an input clock)

SCLKME bit (PCR_REG[7]) and CLKSM bit (SRGR2_REG[13]) are used to set the SRG clock mode.

The sample rate generator can produce a clock signal (CLKG) for use by the receiver, the transmitter, or both, but CLKG is derived from an input clock.

#### 17.2.8.5.23 Set the SRG Input Clock Polarity

CLKSP bit (SRGR2_REG[14]), CLKXP bit (PCR_REG[1]) and CLKRP bit (PCR_REG[0]) are used to set the SRG input clock polarity.

The sample rate generator can produce a clock signal (CLKG) and a frame–synchronization signal (FSG) for use by the receiver, the transmitter, or both. To produce CLKG and FSG, the sample rate generator must be driven by an input clock signal derived from the McBSP_FCLK clock or from an external clock on the McBSP.CLKS, McBSP.CLKX, or McBSP.CLKR pin. If you use a pin, choose a polarity for that pin by using the appropriate polarity bit (CLKSP for the McBSP.CLKS pin, CLKXP for the McBSP.CLKX pin, CLKRP for the McBSP.CLKR pin). The polarity determines whether the rising or falling edge of the input clock generates transitions on CLKG and FSG.

### 17.2.8.6 Transmitter Configuration

To configure the McBSP transmitter, perform the following procedure:

1. Place the McBSP transmitter in reset
2. Program the McBSP registers for the desired transmitter
3. Take the transmitter out of reset

These three steps are detailed in the following subsections.

#### 17.2.8.6.1 Programming the McBSP Registers for the Desired Transmitter Operation

The following is a list of important tasks to be performed when you are configuring the McBSP transmitter. Each task corresponds to one or more McBSP register bit fields.

**Global behavior:**

1. Set the transmitter pins to operate as McBSP pins.
2. Enable/disable the digital loop back mode.
3. Enable/disable the analog loop back mode.
4. Enable/disable the synchronous transmit-receive mode.
5. Enable/disable transmit multichannel selection.


Copyright © 2011–2016, Texas Instruments Incorporated

**Data behavior:**

1. Choose 1 or 2 phases for the transmit frame.
2. Set the transmit word length(s).
3. Set the transmit frame length (one word per phase if dual-phase selected).
4. Enable/disable the transmit frame–synchronization ignore function.
5. Set the transmit data delay.
6. Set the transmit DXENA mode.
7. Set the transmit interrupt mode.

**Frame-synchronization behavior:**

1. Set the transmit frame–synchronization mode.
2. Set the transmit frame–synchronization polarity.
3. Set the SRG frame–synchronization period and pulse width.

**Clock behavior:**

1. Set the transmit clock mode.
2. Set the transmit clock polarity.
3. Set the SRG clock divide–down value.
4. Set the SRG clock synchronization mode.
5. Set the SRG clock mode (choose an input clock).
6. Set the SRG input clock polarity.

#### *17.2.8.6.2 Resetting and Enabling the Transmitter*

The first step of the transmitter configuration procedure is to reset the transmitter, and the last step is to enable the transmitter (to take it out of reset). The serial port can be reset in the following two ways:

1. A global reset places the receiver, transmitter, and sample rate generator in reset. When the device reset is removed, GRST, FRST, RRST and XRST bits = 0, keeping the entire serial port in the reset state.
2. The serial port transmitter and receiver can be reset directly using the RRST and XRST bits in the SPCR1_REG register. The sample rate generator can be reset directly using the GRST bit in SPCR2_REG register.

#### *17.2.8.6.3 Set the Transmitter Pins to Operate as McBSP Pins*

The XIOEN bit (PCR_REG[13]) determines whether the transmitter pins are McBSP pins or general–purpose I/O pins.

Refer to Section 17.2.8.7 for further information.

#### *17.2.8.6.4 Enable/Disable the Digital Loop Back Mode*

The DLB bit (XCCR_REG) determines whether the digital loop-back mode is on or off. In the digital loop-back mode, the receive signals are connected internally through multiplexers to the corresponding transmit signals (DR is connected to transmit data on DX output pin DXO, FSR is connected to FRX output pin FSXO, and CLKR is connected to the CLKX output pin CLKXO). This mode allows testing of serial port; the McBSP receives the data it transmits. This loop back mode is not done through pads, all output signals being disabled (CLKREN, CLKXEN, FSREN, FSRXEN, DXEN are not active).

Note that in digital loop back mode the sample rate generator and the frame synchronization generator need to be enabled in order to generate the CLKX and FSX signals.

#### *17.2.8.6.5 Enable/Disable the Transmit Multichannel Selection*

The XMCM bit field (MCR2_REG[1:0]) determines whether the transmit multichannel selection mode is on or off.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.6.6 Choose 1 or 2 Phases for the Transmit Frame

The XPHASE bit (XCR2_REG[15]) determines whether the transmit data frame has one or two phases.

#### 17.2.8.6.7 Set the Transmit Word Length(s)

The XWDLEN1 (XCR1_REG[7:5]) and XWDLEN2 (XCR2_REG[7:5]) bit fields determine how many bits are in each serial word in phase 1 and in phase 2, respectively, of the transmit data frame.

Each frame can have one or two phases, depending on the value that you load into the XPHASE bit. If a single–phase frame is selected, XWDLEN1 selects the length for every serial word received in the frame. If a dual–phase frame is selected, XWDLEN1 determines the length of the serial words in phase 1 of the frame and XWDLEN2 determines the word length in phase 2 of the frame.

#### 17.2.8.6.8 Set the Transmit Frame Length

The XFRLEN1 (XCR1_REG[14:8]) and XFRLEN2 (XCR2_REG[14:8]) bit fields determine how many serial words are in phase 1 and in phase 2, respectively, of the transmit data frame.

The transmit frame length is the number of serial words in the transmit frame. Each frame can have one or two phases, depending on value that you load into the XPHASE bit. If a single–phase frame is selected (XPHASE = 0), the frame length is equal to the length of phase 1. If a dual–phase frame is selected (XPHASE = 1), the frame length is 2, the length of phase 1 (one word) plus the length of phase 2 (one word).

The 7–bit XFRLEN fields allow up to 128 words per phase. See Table 17-15 for a summary of how to calculate the frame length. This length corresponds to the number of words or logical time slots or channels per frame–synchronization pulse.

Program the XFRLEN fields with [w minus 1], where w represents the number of words per phase. For the example, if you want a phase length of 128 words in phase 1, load 127 into XFRLEN1.

**Table 17-15. How to Calculate the Length of the Transmit Frame**

| XPHASE | XFRLEN1 | XFRLEN2 | Frame Length |
|---|---|---|---|
| 0 | 0 <= XFRLEN1 <= 127 | Don't care | (XFRLEN1 + 1) words |
| 1 | XFRLEN1= 0 | XFRLEN2 = 0 | 2 words |

#### 17.2.8.6.9 Set the Transmit Reverse Mode

The XREVERSE bit field (XCR2_REG[4:3]) determines whether reverse (LSB first) data transfer option is chosen for McBSP reception.

#### 17.2.8.6.10 Set the Transmit Data Delay

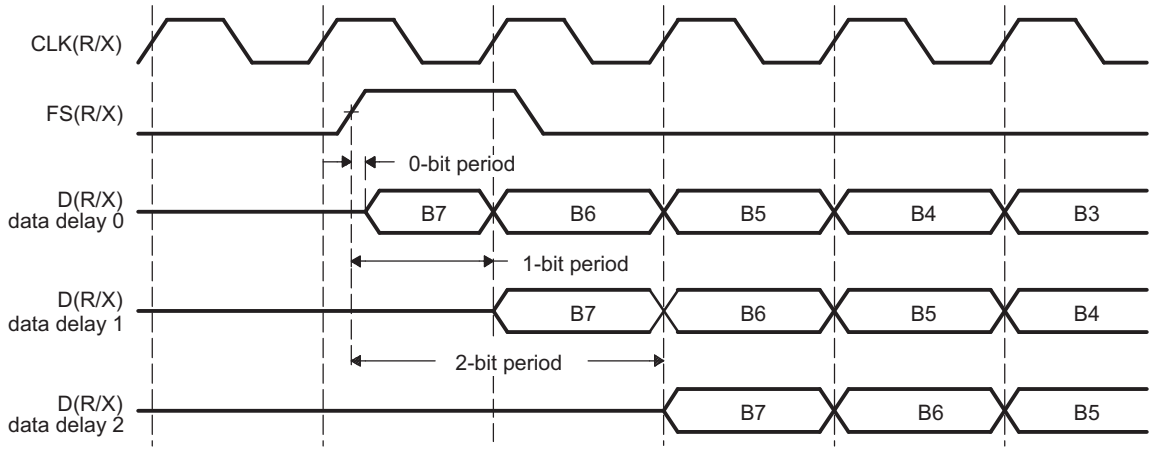
The XDATDLY bit field (XCR2_REG[1:0]) determines the length of the data delay for the transmit frame.

The start of a frame is defined by the first clock cycle in which frame synchronization is found to be active. The beginning of actual data reception or transmission with respect to the start of the frame can be delayed if required. This delay is called data delay.

XDATDLY specifies the data delay for reception. The range of programmable data delay is 0 to 2 bit–clocks (XDATDLY = 00b–10b), as shown in Figure 17-30. In this figure, the data transferred is an 8–bit value with bits labeled B7, B6, B5, and so on. Typically a 1–bit delay is selected, because data often follows a 1–cycle active frame–synchronization pulse.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 17-30. Range of Programmable Data Delay**

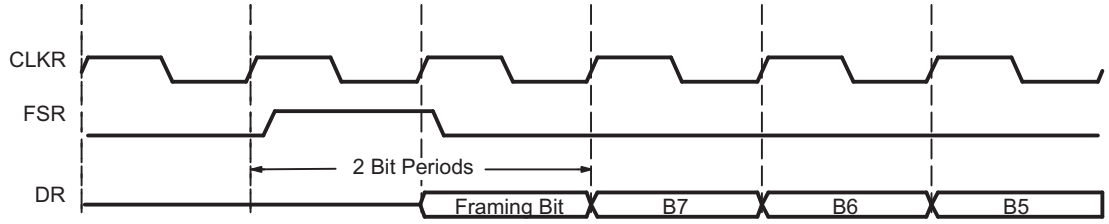



**0-bit data delay:**

Normally, a frame–synchronization pulse is detected or sampled with respect to an edge of internal serial clock CLK(R/X). Thus, on the following cycle or later (depending on the data delay value), data may be received or transmitted. However, in the case of 0–bit data delay, the data must be ready for reception and/or transmission on the same serial clock cycle.

For reception, this problem is solved because receive data is sampled on the first falling edge of CLKR where an active–high internal FSR is detected. However, data transmission must begin on the rising edge of the internal CLKX clock that generated the frame synchronization. Therefore, the first data bit is assumed to be present in XSR1, and thus on McBSP.DX. The transmitter then asynchronously detects the frame–synchronization signal (FSX) going active high and immediately starts driving the first bit to be transmitted on the McBSP.DX pin.

**Figure 17-31. 2-Bit Data Delay Used to Skip a Framing Bit**




### *17.2.8.6.11 Set the Transmit DXENA Mode*

DXENA bit (SPRC1_REG[7]) is used to set the transmit DXENA (DX delay enable) mode.

The DXENA bit controls the delay enabler on the McBSP.DX pin. Set DXENA to enable an extra delay for turn–on time. This bit does not control the data itself, so only the first bit is delayed (the delay is given by a combinatorial delay buffer). The inserted delay: 7 ns, 14 ns (default), 20 ns or 28 ns can be set using the DXENDLY bit field in XCCR_REG register. If you tie together the McBSP.DX pins of multiple McBSPs, make sure DXENA = 1 to avoid having more than one McBSP transmitting on the data line at one time.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.6.12 Set the Transmit Interrupt Mode (legacy only)

The XINTM bit field (SPCRX_REG[5:4]) determines which event generates a transmit interrupt request to the CPU.

The transmit interrupt (XINT) informs the CPU of changes to the serial port status. Four options exist for configuring this interrupt.

- XINTM = 00: XINT generated when the XRDY bit (SPCR2_REG[1]) changes from 0 to 1. Interrupt on every serial word by tracking the XRDY bit in the SPCR2_REG register. Regardless of the value of XINTM, XRDY can be read to detect the XRDY = 1 condition.
- XINTM = 01: XINT generated by an end–of–frame condition in the transmit multichannel selection mode. In any other serial transfer case, this setting is not applicable and, therefore, no interrupts are generated.
- XINTM = 10: XINT generated by a new transmit frame–synchronization pulse. Interrupt on detection of transmit frame–synchronization pulses. This generates an interrupt even when the transmitter is in its reset state. This is done by synchronizing the incoming frame–synchronization pulse to the CPU clock and sending it to the CPU via XINT.
- XINTM = 11: XINT generated when XSYNCERR is set. Interrupt on frame–synchronization error. Regardless of the value of XINTM, XSYNCERR can be read to detect this condition.

The McBSP provides also a common interrupt line McBSP.COMMONIRQ, which can be used by setting the IRQENABLE register. All the above settings have equivalent enable bits in the IRQENABLE register in order to enable the common interrupt line (XDYEN is equivalent with XINTM = 00, XEOFEN is equivalent with XINTM = 01 (the interrupt is generated by an end-of-frame condition regardless of the multichannel selection mode), XFSXEN is equivalent with XINTM = 10 and XSYNCERREN is equivalent with XINTM = 11 setting). This interrupt line has its own status register.

#### 17.2.8.6.13 Set the Transmit Frame-Synchronization Mode

FSXM bit (PCR_REG[11]) and FSGM bit (SRGR2_REG[12]) are used to set the transmit frame-synchronization mode.

Table 17-16 shows how FSXM and FSGM select the source of transmit frame–synchronization pulses. The three choices are:

1. External frame–synchronization input
2. Sample rate generator frame–synchronization signal (FSG)
3. Sample rate generator frame–synchronization signal (FSG) gated by the transmit buffer XB empty condition.

Table 17-16 also shows the effect of each bit setting on the McBSP.FSX pin. The FSXP bit determines the polarity of the signal on the McBSP.FSX pin.

**Table 17-16. How FSXM and FSGM Select the Source of Transmit Frame-Synchronization Pulses**

| FSXM | FSGM | Source of Transmit Frame Synchronization | McBSP.FSX Pin Status |
|---|---|---|---|
| 0 | 0 or 1 | An external frame–synchronization signal enters the McBSP through the McBSP.FSX pin. The signal is then inverted by FSXP before being used as internal FSX. | Input |
| 1 | 1 | Internal FSX is driven by the sample rate generator frame–synchronization signal (FSG). | Output. FSG is inverted by FSXP before being driven out on McBSP.FSX pin. |
| 1 | 0 | A XB empty condition causes the McBSP not to generate a transmit frame–synchronization pulse. The frame synchronization is generated taking into account the FWID and FPER as long as the transmit buffer is not empty. When the buffer is empty the generated frame synchronization signal is gated. | Output. The generated frame–synchronization pulse is inverted as determined by FSXP before being driven out on McBSP.FSX pin. |


Copyright © 2011–2016, Texas Instruments Incorporated

If the sample rate generator creates a frame–synchronization signal (FSG) that is derived from an external input clock, the GSYNC bit determines whether FSG is kept synchronized with pulses on the McBSP.FSR pin.

In the digital loop-back mode (DLB = 1) or analog loop-back mode (ALB = 1), the transmit frame synchronization signal is used as the receive frame synchronization signal.

#### *17.2.8.6.14 Set the Transmit Frame-Synchronization Polarity*

The FSXP bit (PCR_REG[3]) determines whether frame–synchronization pulses are active high or active low on the McBSP.FSX pin.

Transmit frame–synchronization pulses can be generated internally by the sample rate generator or driven by an external source. The source of frame synchronization is selected by programming the PCR_REG[11] register FSXM mode bit. FSX is also affected by the FSGM bit (SRGR2_REG[12]). Similarly, transmit clocks can be selected to be inputs or outputs by programming the PCR_REG[9] register CLKXM mode bit.

When FSR and FSX are inputs (FSXM = FSRM= 0, external frame–synchronization pulses), the McBSP detects them on the internal falling edge of clock, internal CLKR, and internal CLKX, respectively. The receive data arriving at the McBSP.DR pin is also sampled on the falling edge of internal CLKR. These internal clock signals are either derived from external source via CLK(R/X) pins or driven by the sample rate generator clock (CLKG) internal to the McBSP.

When FSR and FSX are outputs, implying that they are driven by the sample rate generator, they are generated (transition to their active state) on the rising edge of internal clock, CLK(R/X). Similarly, data on the McBSP.DX pin is output on the rising edge of internal CLKX.

FSRP, FSXP, CLKRP, and CLKXP in the pin control register (PCR_REG) configure the polarities of the FSR, FSX, CLKR, and CLKX signals, respectively. All frame–synchronization signals (internal FSR, internal FSX) that are internal to the serial port are active high. If the serial port is configured for external frame synchronization (FSR/FSX are inputs to McBSP) and FSRP = FSXP = 1, the external active–low frame–synchronization signals are inverted before being sent to the receiver (internal FSR) and transmitter (internal FSX). Similarly, if internal synchronization (FSR/FSX are output pins and GSYNC = 0) is selected and the polarity bit FS(R/X)P = 1, the internal active–high frame–synchronization signals are inverted before being sent to the FS(R/X) pin.

On the transmit side, the transmit clock polarity bit, CLKXP, sets the edge used to shift and clock out transmit data. Data is always transmitted on the rising edge of internal CLKX. If CLKXP = 1 and external clocking is selected (CLKXM = 0 and CLKX is an input), the external falling–edge triggered input clock on CLKX is inverted to a rising–edge triggered clock before being sent to the transmitter. If CLKXP = 1, and internal clocking selected (CLKXM = 1 and CLKX is an output pin), the internal (rising–edge triggered) clock, internal CLKX, is inverted before being sent out on the McBSP.CLKX pin.

Similarly, the receiver can reliably sample data that is clocked with a rising edge clock (by the transmitter). The receive clock polarity bit, CLKRP, sets the edge used to sample received data. The receive data is always sampled on the falling edge of internal CLKR. Therefore, if CLKRP = 1 and external clocking is selected (CLKRM = 0 and CLKR is an input pin), the external rising–edge triggered input clock on CLKR is inverted to a falling–edge triggered clock before being sent to the receiver. If CLKRP = 1 and internal clocking is selected (CLKRM = 1), the internal falling–edge triggered clock is inverted to a rising–edge triggered clock before being sent out on the McBSP.CLKR pin.

Note that CLKRP = CLKXP in a system where the same clock (internal or external) is used to clock the receiver and transmitter. The receiver uses the opposite edge as the transmitter to ensure valid setup and hold of data around this edge.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.6.15 Set the SRG Frame-Synchronization Period and Pulse Width

FPER bit field (SRGR2_REG[11:0]) is used to set the SRG frame-synchronization period and FWID bit field (SRGR1_REG[15:8]) is used to set the SRG pulse width. The sample rate generator can produce a clock signal, CLKG, and a frame–synchronization signal, FSG. If the sample rate generator is supplying receive or transmit frame synchronization, you must program the bit fields FPER and FWID.

On FSG, the period from the start of a frame–synchronization pulse to the start of the next pulse is (FPER + 1) CLKG cycles. The 12 bits of FPER allow a frame–synchronization period of 1 to 4096 CLKG cycles, which allows up to 4096 data bits per frame. When GSYNC = 1, FPER is a don't care value.

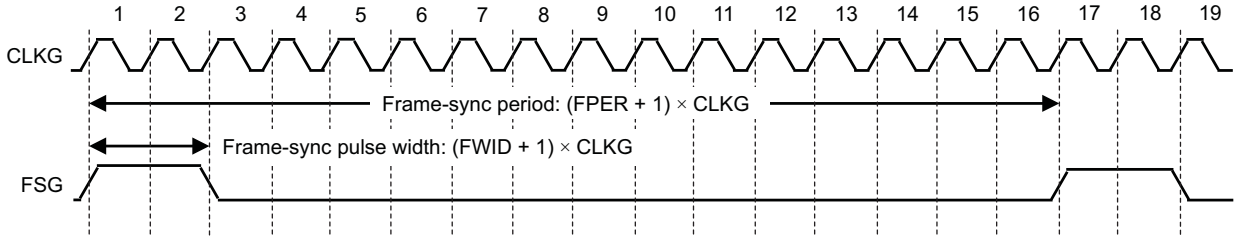
Each pulse on FSG has a width of (FWID + 1) CLKG cycles. The eight bits of FWID allow a pulse width of 1 to 256 CLKG cycles. It is recommended that FWID be programmed to a value less than the programmed word length.

The values in FPER and FWID are loaded into separate down–counters. The 12–bit FPER counter counts down the generated clock cycles from the programmed value (4095 maximum) to 0. The 8–bit FWID counter counts down from the programmed value (255 maximum) to 0.

Figure 17-32 shows a frame–synchronization period of 16 CLKG periods (FPER = 15 or 0000 1111b) and a frame–synchronization pulse with an active width of 2 CLKG periods (FWID = 1).

When the sample rate generator comes out of reset, FSG is in its inactive state. Then, when GRST = 1 and FSGM = 1, a frame–synchronization pulse is generated. The frame width value (FWID + 1) is counted down on every CLKG cycle until it reaches 0, at which time FSG goes low. At the same time, the frame period value (FPER + 1) is also counting down. When this value reaches 0, FSG goes high, indicating a new frame.

**Figure 17-32. Frame of 16 CLKG Periods and Active Width of 2 CLKG Periods**




#### 17.2.8.6.16 Set the Transmit Clock Mode

CLKXM bit (PCR_REG[9]) is used to set the transmit clock mode

Table 17-17 shows how the CLKXM bit selects the transmit clock and the corresponding status of the McBSP.CLKX pin. The CLKXP bit determines the polarity of the signal on the McBSP.CLKX pin.

If the sample rate generator creates a clock signal (CLKG) that is derived from an external input clock, the GSYNC bit determines whether CLKG is kept synchronized with pulses on the McBSP.FSR pin.

In the digital loop-back mode (DLB = 1) or analog loop-back mode (ALB = 1), the transmit frame synchronization signal is used as the receive frame synchronization signal.

**Table 17-17. CLKXM Bit Effect on Transmit Clock and McBSP.CLKX Pin**

| CLKXM | Source of Transmit Clock | McBSP.CLKX Pin Status |
|---|---|---|
| 0 | Internal CLKX is driven by an external clock on the McBSP.CLKX pin. CLKX is inverted as determined by CLKXP before being used. | Input |
| 1 | Internal CLKX is driven by the sample rate generator clock, CLKG. | Output. CLKG, inverted as determined by CLKXP, is driven out on McBSP.CLKX. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.6.17 Set the Transmit Clock Polarity

CLKXP bit (PCR_REG[1]) is used to set the transmit clock polarity.

Transmit frame–synchronization pulses can be either generated internally by the sample rate generator or driven by an external source. The source of frame synchronization is selected by programming the PCR_REG[11] register FSXM mode bit. FSX is also affected by the FSGM bit (SRGR2_REG[12]). Similarly, transmit clocks can be selected to be inputs or outputs by programming the PCR_REG[9] register CLKXM mode bit.

When FSR and FSX are inputs (FSXM = FSRM = 0, external frame–synchronization pulses), the McBSP detects them on the internal falling edge of clock, internal CLKR, and internal CLKX, respectively. The receive data arriving at the McBSP.DR pin is also sampled on the falling edge of internal CLKR. These internal clock signals are either derived from external source via CLK(R/X) pins or driven by the sample rate generator clock (CLKG) internal to the McBSP.

When FSR and FSX are outputs, implying that they are driven by the sample rate generator, they are generated (transition to their active state) on the rising edge of internal clock, CLK(R/X). Similarly, data on the McBSP.DX pin is output on the rising edge of internal CLKX. FSRP, FSXP, CLKRP, and CLKXP in the pin control register (PCR_REG) configure the polarities of the FSR, FSX, CLKR, and CLKX signals, respectively. All frame–synchronization signals (internal FSR, internal FSX) that are internal to the serial port are active high. If the serial port is configured for external frame synchronization (FSR/FSX are inputs to McBSP), and FSRP = FSXP = 1, the external active–low frame–synchronization signals are inverted before being sent to the receiver (internal FSR) and transmitter (internal FSX). Similarly, if internal synchronization (FSR/FSX are output pins and GSYNC = 0) is selected, the internal active–high frame–synchronization signals are inverted, if the polarity bit FS(R/X)P = 1, before being sent to the FS(R/X) pin.

On the transmit side, the transmit clock polarity bit, CLKXP, sets the edge used to shift and clock out transmit data. Data is always transmitted on the rising edge of internal CLKX. If CLKXP = 1 and external clocking is selected (CLKXM = 0 and CLKX is an input), the external falling–edge triggered input clock on CLKX is inverted to a rising–edge triggered clock before being sent to the transmitter. If CLKXP = 1 and internal clocking is selected (CLKXM = 1 and CLKX is an output pin), the internal (rising–edge triggered) clock, internal CLKX, is inverted before being sent out on the McBSP.CLKX pin.

Similarly, the receiver can reliably sample data that is clocked with a rising edge clock (by the transmitter). The receive clock polarity bit, CLKRP, sets the edge used to sample received data. The receive data is always sampled on the falling edge of internal CLKR. Therefore, if CLKRP = 1 and external clocking is selected (CLKRM = 0 and CLKR is an input pin), the external rising–edge triggered input clock on CLKR is inverted to a falling–edge triggered clock before being sent to the receiver. If CLKRP = 1 and internal clocking is selected (CLKRM = 1), the internal falling–edge triggered clock is inverted to a rising–edge triggered clock before being sent out on the McBSP.CLKR pin. Note that CLKRP = CLKXP in a system where the same clock (internal or external) is used to clock the receiver and transmitter. The receiver uses the opposite edge as the transmitter to ensure valid setup and hold of data around this edge.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 17.2.8.6.18 Set the SRG Clock Divide-Down Value

CLKGDV bit field (SRGR1_REG[7:0]) is used to set the sample rate generator clock divide-down value.

The first divider stage generates the serial data bit clock from the input clock. This divider stage utilizes a counter, preloaded by CLKGDV, that contains the divide ratio value.

The output of the first divider stage is the data bit clock, which is output as CLKG and which serves as the input for the second and third stages of the divider.

CLKG has a frequency equal to 1/(CLKGDV + 1) of sample rate generator input clock. Thus, the sample generator input clock frequency is divided by a value between 1 and 256. The CLKG duty cycle is 50%.

#### 17.2.8.6.19 Set the SRG Clock Synchronization Mode

GSYNC bit (SRGR2_REG[15]) is used to set the SRG clock synchronization mode.

#### 17.2.8.6.20 Set the SRG Clock Mode (choose an input clock)

SCLKME bit (PCR_REG[7]) and CLKSM bit (SRGR2_REG[13]) are used to set the SRG clock mode.

The sample rate generator can produce a clock signal (CLKG) for use by the receiver, the transmitter, or both, but CLKG is derived from an input clock.

#### 17.2.8.6.21 Set the SRG Input Clock Polarity

CLKSP bit (SRGR2_REG[14]), CLKXP bit (PCR_REG[1]) and CLKRP bit (PCR_REG[0]) are used to set the SRG input clock polarity.

The sample rate generator can produce a clock signal (CLKG) and a frame–synchronization signal (FSG) for use by the receiver, the transmitter, or both. To produce CLKG and FSG, the sample rate generator must be driven by an input clock signal derived from the McBSP_FCLK clock or from an external clock on the McBSP.CLKS, McBSP.CLKX, or McBSP.CLKR pin. If you use a pin, choose a polarity for that pin by using the appropriate polarity bit (CLKSP for the McBSP.CLKS pin, CLKXP for the McBSP.CLKX pin, CLKRP for the McBSP.CLKR pin). The polarity determines whether the rising or falling edge of the input clock generates transitions on CLKG and FSG.


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.2.8.7 General-Purpose I/O on the McBSP Pins (legacy only)

Table 17-18 summarizes how to use the McBSP pins as general–purpose I/O pins. All of the bits mentioned in the table except XRST and RRST bits are in the pin control register (PCR_REG). XRST and RRST are in the serial port control registers (SPCR2_REG[0] and SPCR1_REG[0], respectively.

To use receiver pins McBSP.CLKR, McBSP.FSR, and McBSP.DR as general–purpose I/O pins rather than as serial port pins, you must set two conditions:

1. The receiver of the serial port is in reset (SPCR1_REG[0] RRST bit = 0).
2. General–purpose I/O is enabled for the serial port receiver (RIOEN = 1 in PCR_REG[12]).

The McBSP.CLKR and McBSP.FSR pins can be individually configured as either input or output pins with the CLKRM and FSRM bits, respectively. The McBSP.DR pin can only be an input pin. The following table shows, which bits in PCR_REG are used to read from/write to these pins.

For the transmitter pins McBSP.CLKX, McBSP.FSX, and McBSP.DX, you must meet two conditions:

1. The transmitter of the serial port is in reset (XRST = 0 in SPCR2_REG[0]).
2. General–purpose I/O is enabled for the serial port transmitter (XIOEN = 1 in PCR_REG[12]).

The McBSP.CLKX and McBSP.FSX pins can be individually configured as input or output pins with the CLKXM and FSXM bits, respectively. The McBSP.DX pin can only be an output pin.

For the McBSP.CLKS pin, all of the reset and I/O enable conditions must be met:

1. Both the receiver and transmitter of the serial port are in reset (RRST = 0 and XRST = 0).
2. General–purpose I/O is enabled for both the receiver and the transmitter (RIOEN = 1 and XIOEN = 1).

The McBSP.CLKS pin can only be an input pin. To read the status of the signal on the McBSP.CLKS pin, read the CLKS_STAT bit in PCR_REG.

**Table 17-18. Using McBSP Pins for General-Purpose I/O**

| Pin | General–Purpose Use Enabled by This Bit Combination | Selected as Input When ... | Input Value Read From This Bit |
|---|---|---|---|
| McBSP.CLKX | XRST = 0<br>XIOEN = 1 | CLKXM = 0 | CLKXP |
| McBSP.FSX | XRST = 0<br>XIOEN = 1 | FSXM = 0 | FSXP |
| McBSP.DX | XRST = 0<br>XIOEN = 1 | Never | Does not apply |
| McBSP.CLKR | RRST = 0<br>RIOEN = 1 | CLKRM = 0 | CLKRP |
| McBSP.FSR | RRST = 0<br>RIOEN = 1 | FSRM = 0 | FSRP |
| McBSP.DR | RRST = 0<br>RIOEN = 1 | Always | DR_STAT |
| McBSP.CLKS | RRST = XRST = 0<br>RIOEN = XIOEN = 1 | Always | CLKS_STAT |


Copyright © 2011–2016, Texas Instruments Incorporated

## 17.3 McBSP Registers

Table 17-19 lists the registers for the McBSP.

**Table 17-19. McBSP Registers**

| Address Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | REVNB | Revision Number Register | Section 17.3.1 |
| 10h | SYSCONFIG_REG | System Configuration Register | Section 17.3.2 |
| 20h | EOI | End of Interrupt Register | Section 17.3.3 |
| 24h | IRQSTATUS_RAW | Interrupt Status Raw Register | Section 17.3.4 |
| 28h | IRQSTATUS | Interrupt Status Register | Section 17.3.5 |
| 2Ch | IRQENABLE_SET | Interrupt Enable Set Register | Section 17.3.6 |
| 30h | IRQENABLE_CLR | Interrupt Enable Clear Register | Section 17.3.7 |
| 34h | DMARXENABLE_SET | DMA Rx Enable Set Register | Section 17.3.8 |
| 38h | DMATXENABLE_SET | DMA Tx Enable Set Register | Section 17.3.9 |
| 3Ch | DMARXENABLE_CLR | DMA Rx Enable Clear Register | Section 17.3.10 |
| 40h | DMATXENABLE_CLR | DMA Tx Enable Clear Register | Section 17.3.11 |
| 48h | DMARXWAKE_EN | DMA Rx Wake Enable Register | Section 17.3.12 |
| 4Ch | DMATXWAKE_EN | DMA Tx Wake Enable Register | Section 17.3.13 |
| B4h | XBUFFSTAT_REG | McBSP Transmit Buffer Status Register | Section 17.3.50 |
| 100h | DRR_REG | McBSP Data Receive Register | Section 17.3.14 |
| 108h | DXR_REG | McBSP Data Transmit Register | Section 17.3.15 |
| 110h | SPCR2_REG | McBSP Serial Port Control Register 2 | Section 17.3.16 |
| 114h | SPCR1_REG | McBSP Serial Port Control Register 1 | Section 17.3.17 |
| 118h | RCR2_REG | McBSP Receive Control Register 2 | Section 17.3.18 |
| 11Ch | RCR1_REG | McBSP Receive Control Register 1 | Section 17.3.19 |
| 120h | XCR2_REG | McBSP Transmit Control Register 2 | Section 17.3.20 |
| 124h | XCR1_REG | McBSP Transmit Control Register 1 | Section 17.3.21 |
| 128h | SRGR2_REG | McBSP Sample Rate Generator Register 2 | Section 17.3.22 |
| 12Ch | SRGR1_REG | McBSP Sample Rate Generator Register 1 | Section 17.3.23 |
| 130h | MCR2_REG | McBSP Multichannel Register 2 | Section 17.3.24 |
| 134h | MCR1_REG | McBSP Multichannel Register 1 | Section 17.3.25 |
| 138h | RCERA_REG | McBSP Receive Channel Enable Register Partition A | Section 17.3.26 |
| 13Ch | RCERB_REG | McBSP Receive Channel Enable Register Partition B | Section 17.3.27 |
| 140h | XCERA_REG | McBSP Transmit Channel Enable Register Partition A | Section 17.3.28 |
| 144h | XCERB_REG | McBSP Transmit Channel Enable Register Partition B | Section 17.3.29 |
| 148h | PCR_REG | McBSP Pin Control Register | Section 17.3.30 |
| 14Ch | RCERC_REG | McBSP Receive Channel Enable Register Partition C | Section 17.3.31 |
| 150h | RCERD_REG | McBSP Receive Channel Enable Register Partition D | Section 17.3.32 |
| 154h | XCERC_REG | McBSP Transmit Channel Enable Register Partition C | Section 17.3.33 |
| 158h | XCERD_REG | McBSP Transmit Channel Enable Register Partition D | Section 17.3.34 |
| 15Ch | RCERE_REG | McBSP Receive Channel Enable Register Partition E | Section 17.3.35 |
| 160h | RCERF_REG | McBSP Receive Channel Enable Register Partition F | Section 17.3.36 |
| 164h | XCERE_REG | McBSP Transmit Channel Enable Register Partition E | Section 17.3.37 |
| 168h | XCERF_REG | McBSP Transmit Channel Enable Register Partition F | Section 17.3.38 |
| 16Ch | RCERG_REG | McBSP Receive Channel Enable Register Partition G | Section 17.3.39 |
| 170h | RCERH_REG | McBSP Receive Channel Enable Register Partition H | Section 17.3.40 |
| 174h | XCERG_REG | McBSP Transmit Channel Enable Register Partition G | Section 17.3.41 |
| 178h | XCERH_REG | McBSP Transmit Channel Enable Register Partition H | Section 17.3.42 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-19. McBSP Registers (continued)**

| Address Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 190h | THRSH2_REG | McBSP Transmit Buffer Threshold Register (DMA or IRQ Trigger) | Section 17.3.43 |
| 194h | THRSH1_REG | McBSP Receive Buffer Threshold Register (DMA or IRQ Trigger) | Section 17.3.44 |
| 1A0h | IRQSTATATUS | McBSP Interrupt Status Register (OCP Compliant IRQ Line) | Section 17.3.45 |
| 1A4h | IRQENABLE | McBSP Interrupt Enable Register (OCP Compliant IRQ Line) | Section 17.3.46 |
| 1A8h | WAKEUPEN | McBSP Wakeup Enable Register | Section 17.3.47 |
| 1ACh | XCCR_REG | McBSP Transmit Configuration Control Register | Section 17.3.48 |
| 1B0h | RCCR_REG | McBSP Receive Configuration Control Register | Section 17.3.49 |
| 1B8h | RBUFFSTAT_REG | McBSP Receive Buffer Status Register | Section 17.3.51 |
| 1C0h | STATUS_REG | McBSP Status Register | Section 17.3.52 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.1 Revision Number Register (REVNB)

The Revision Number (REVNB) register is shown in Figure 17-33 and described in Table 17-20.

This read-only register contains the hard-coded revision number of the module. A write to this register has no effect.

**Figure 17-33. Revision Number Register (REVNB)**

| 31 | 30 | 29 | 28 | 27 | 16 |
|---|---|---|---|---|---|
| SCHEME | | Reserved | | FUNC | |
| R-1 | | R-1 | | R-8h | |

| 15 | 11 | 10 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| RTL | | MAJOR | | CUSTOM | | MINOR | |
| R-0 | | R-0 | | R-0 | | R-37h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-20. Revision Number Register (REVNB) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-30 | SCHEME | 1h | Used to distinguish between old Scheme and current. Spare bit to encode future schemes. |
| 29-28 | Reserved | 0 | Reads return 0x1. |
| 27-16 | FUNC | 8h | Function: Indicates a software compatible module family. |
| 15-11 | RTL | 0h | RTL version. |
| 10-8 | MAJOR | 0h | Major Revision. |
| 7-6 | CUSTOM | 0h | Custom ID. |
| 5-0 | MINOR | 37h | Minor Revision. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.2 System Configuration Register (SYSCONFIG_REG)

The System Configuration (SYSCONFIG_REG) register is shown in Figure 17-34 and described in Table 17-21.

**Figure 17-34. System Configuration Register (SYSCONFIG_REG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 |
|---|---|---|---|
| Reserved | | CLOCKACTIVITY | |
| | | R/W-0 | |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | SIDLEMODE | | ENAWAKEUP | SOFTRESET | Reserved |
| R-0 | | R/W-0 | | R/W-0 | R/W-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-21. System Configuration Register (SYSCONFIG_REG) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved. |
| 9-8 | CLOCKACTIVITY | Bit 9 | OCP interface clock. |
| | | 0 | OCP clock can be switched-off. |
| | | 1 | OCP clock is maintained during wake up period. |
| | | Bit 8 | Functional clock. |
| | | 0 | Functional clock can be switched-off. |
| | | 1 | Functional clock is maintained during wake up period. |
| | | 0 | OCP clock can be switched-off. Functional clock can be switched-off. |
| | | 1h | OCP clock can be switched-off. Functional clock is maintained during wake up period. |
| | | 2h | OCP clock is maintained during wake up period. Functional clock can be switched-off. |
| | | 3h | OCP clock is maintained during wake up period. Functional clock is maintained during wake up period. |
| 7-5 | Reserved | 0 | Reserved. |
| 4-3 | SIDLEMODE | | Slave interface power management, req/ack control |
| | | 0 | Force-idle. An idle request is acknowledged unconditionally. |
| | | 1h | No-idle. An idle request is never acknowledged. |
| | | 2h | Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module. |
| | | 3h | Smart Idle Wakeup. |
| 2 | ENAWAKEUP | | WakeUp feature control. |
| | | 0 | WakeUp is disabled. |
| | | 1 | WakeUp capability is enabled. |
| 1 | SOFTRESET | | McBSPLP global software reset. |
| | | 0 | Read: Reset done, no pending action. |
| | | | Write: No action. |
| | | 1 | Read: Reset (software or other) ongoing. |
| | | | Write: Initiate software reset. |
| 0 | Reserved | 0 | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.3 End of Interrupt Register (EOI)

The End of Interrupt (EOI) register is shown in Figure 17-35 and described in Table 17-22.

**Figure 17-35. End of Interrupt Register (EOI)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | LINE_NUMBER |
| R-0 | | W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-22. End of Interrupt Register (EOI) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | LINE_NUMBER | | Software End Of Interrupt (EOI) control. Write number of interrupt output. Write a value of 0 to force the output interrupt to re-assert if any enabled interrupts are pending. |

### 17.3.4 Interrupt Status Raw Register (IRQSTATUS_RAW)

This register provides core status information for interrupt handling, showing all active events (enabled and not enabled ones). The fields are read-write. Writing a 1 to a bit will set it to 1, for example, trigger the IRQ when also enabled (mostly for debug). Writing a 0 will have no effect, for example, the register value will not be modified. Only enabled, active events will trigger an actual interrupt request on the IRQ output line.

Interrupts can either be serviced using the highlander standard IRQSTATUS, IRQSTATUS_RAW, IRQENABLE_SET, and IRQENABLE_CLR set of registers, or the legacy IRQSTATUS_REG and IRQENABLE_REG registers, but not both in the same system.

The Interrupt Status Raw (IRQSTATUS_RAW) register is shown in Figure 17-36 and described in Table 17-23.

**Figure 17-36. Interrupt Status Raw Register (IRQSTATUS_RAW)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XEMPTYEOF | Reserved | XOVFLSTAT | XUNDFLSTAT | XRDY | XEOF | XFSX |
| R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| XSYNCERR | Reserved | ROVFLSTAT | RUNDFLSTAT | RRDY | REOF | RFSR | RSYNCERR |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-23. Interrupt Status Raw Register (IRQSTATUS_RAW) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. |
| 14 | XEMPTYEOF | | Transmit Buffer Empty at end of frame (XEMPTYEOF is set to 1 when a complete frame was transmitted and the transmit buffer is empty). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | XEMPTYEOF is NOT set to when a complete frame was transmitted and the transmit buffer is empty. Writing a 0 will have no effect. |
| | | 1 | XEMPTYEOF is set to 1 when a complete frame was transmitted and the transmit buffer is empty. Writing 1 to this bit will set it to 1. |
| 13 | Reserved | 0 | Reserved. |
| 12 | XOVFLSTAT | | Transmit Buffer Overflow (XOVFLSTAT bit is set to 1 when transmit buffer overflow; the data which is written while overflow condition is discarded). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | XEMPTYEOF is NOT set to when a complete frame was transmitted and the transmit buffer is empty . Writing a 0 will have no effect. |
| | | 1 | XEMPTYEOF is set to 1 when a complete frame was transmitted and the transmit buffer is empty. Writing 1 to this bit will set it to 1. |
| 11 | XUNDFLSTAT | | Transmit Buffer Underflow (XUNDFLSTAT bit is set to 1 when the transmit data buffer is empty new data is required to be transmitted). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | The transmit data buffer is NOT empty new data is required to be transmitted. Writing a 0 will have no effect. |
| | | 1 | The transmit data buffer is empty new data is required to be transmitted. Writing 1 to this bit will set it to 1. |
| 10 | XRDY | | Transmit Buffer Threshold Reached (XRDY bit is set to 1 when the transmit buffer free locations are equal or above the THRSH2_REG value). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | Transmit buffer occupied locations are below the THRSH2_REG value). Writing a 0 will have no effect. |
| | | 1 | Transmit buffer occupied locations are equal or above the THRSH2_REG value). Writing 1 to this bit will set it to 1. |
| 9 | XEOF | | Transmit End Of Frame (XEOF is set to 1 when a complete frame was transmitted). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | Complete frame was NOT transmitted. Writing a 0 will have no effect. |
| | | 1 | Complete frame was transmitted. Writing 1 to this bit will set it to 1. |
| 8 | XFSX | | Transmit Frame Synchronization (XFSX bit is set to 1 when a new transmit frame synchronization is asserted). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | New transmit frame synchronization is NOT asserted. Writing a 0 will have no effect. |
| | | 1 | New transmit frame synchronization is asserted. Writing 1 to this bit will set it to 1. |
| 7 | XSYNCERR | | Transmit Frame Synchronization Error (XSYNCERR is set to 1 when a transmit frame synchronization error is detected). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | Transmit frame synchronization error is NOT detected. Writing a 0 will have no effect. |
| | | 1 | Transmit frame synchronization error is detected. Writing 1 to this bit will set it to 1. |
| 6 | Reserved | 0 | Reserved. |
| 5 | ROVFLSTAT | | Receive Buffer Overflow (ROVFLSTAT bit is set to 1 when receive buffer overflow; the data which is written while overflow condition is discarded). Writing 1 to this bit will set it to 1. Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | Receive buffer NOT overflow . Writing a 0 will have no effect. |
| | | 1 | Receive buffer overflow; Writing 1 to this bit will set it to 1. |
| 4 | RUNDFLSTAT | | Receive Buffer Underflow (RUNDFLSTAT bit is set to 1 when read operation is performed to the receive data register while receive buffer is empty; data read while underflow condition is undefined). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | Read operation is performed to the receive data register while receive buffer is NOT empty. Writing a 0 will have no effect. |
| | | 1 | Read operation is performed to the receive data register while receive buffer is empty; Writing 1 to this bit will set it to 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-23. Interrupt Status Raw Register (IRQSTATUS_RAW) Field Descriptions (continued)**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 3 | RRDY | | Receive Buffer Threshold Reached (RRDY bit is set to 1 when the receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | Receive buffer occupied locations are below the THRSH1_REG value). Writing a 0 will have no effect. |
| | | 1 | Receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit will set it to 1. |
| 2 | REOF | | Receive End Of Frame (REOF is set to 1 when a complete frame was received). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | Complete frame was NOT received. Writing a 0 will have no effect. |
| | | 1 | Complete frame was received; Writing 1 to this bit will set it to 1. |
| 1 | RFSR | | Receive Frame Synchronization (RFSR bit is set to 1 when a new receive frame synchronization is asserted). Wr Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | New receive frame synchronization is NOT asserted. Writing a 0 will have no effect. |
| | | 1 | New receive frame synchronization is asserted; Writing 1 to this bit will set it to 1. |
| 0 | RSYNCERR | | Receive Frame Synchronization Error (RSYNCERR is set to 1 when a receive frame synchronization error is detected). Writing 1 to this bit will set it to 1. Writing a 0 will have no effect. |
| | | 0 | Receive frame synchronization error is NOT detected. Writing a 0 will have no effect. |
| | | 1 | Receive frame synchronization error is detected. Writing 1 to this bit will set it to 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.5 Interrupt Status Register (IRQSTATUS)

This register provides core status information for interrupt handling, showing all active and enabled events and masking the others. The fields are read-write. Writing a 1 to a bit will clear it to 0, for example, clear the IRQ. Writing a 0 will have no effect, for example, the register value will not be modified. Only enabled, active events will trigger an actual interrupt request on the IRQ output line.

Interrupts can either be serviced using the highlander standard IRQSTATUS, IRQSTATUS_RAW, IRQENABLE_SET, and IRQENABLE_CLR set of registers, or the legacy IRQSTATUS_REG and IRQENABLE_REG registers, but not both in the same system.

The Interrupt Status (IRQSTATUS) register is shown in Figure 17-37 and described in Table 17-24.

**Figure 17-37. Interrupt Status Register (IRQSTATUS)**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XEMPTYEOF | Reserved | XOVFLSTAT | XUNDFLSTAT | XRDY | XEOF | XFSX |
| R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| XSYNCERR | Reserved | ROVFLSTAT | RUNDFLSTAT | RRDY | REOF | RFSR | RSYNCERR |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-24. Interrupt Status Register (IRQSTATUS) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. |
| 14 | XEMPTYEOF | | Transmit Buffer Empty at end of frame (XEMPTYEOF is set to 1 when a complete frame was transmitted and the transmit buffer is empty). |
| | | 0 | XEMPTYEOF is NOT set to when a complete frame was transmitted and the transmit buffer is empty. |
| | | 1 | XEMPTYEOF is set to 1 when a complete frame was transmitted and the transmit buffer is empty. Writing 1 to this bit clears the bit. |
| 13 | Reserved | 0 | Reserved. |
| 12 | XOVFLSTAT | | Transmit Buffer Overflow (XOVFLSTAT bit is set to 1 when transmit buffer overflow; the data which is written while overflow condition is discarded). Writing 1 to this bit clears the bit. |
| | | 0 | Transmit buffer NOT overflow. |
| | | 1 | Transmit buffer overflow; Writing 1 to this bit clears the bit. |
| 11 | XUNDFLSTAT | | Transmit Buffer Underflow (XUNDFLSTAT bit is set to 1 when the transmit data buffer is empty new data is required to be transmitted). Writing 1 to this bit clears the bit. |
| | | 0 | The transmit data buffer is NOT empty new data is required to be transmitted. |
| | | 1 | The transmit data buffer is empty new data is required to be transmitted. Writing 1 to this bit clears the bit. |
| 10 | XRDY | | Transmit Buffer Threshold Reached (XRDY bit is set to 1 when the transmit buffer free locations are equal or above the THRSH2_REG value). Writing 1 to this bit clears the bit. |
| | | 0 | Transmit buffer occupied locations are below the THRSH2_REG value). |
| | | 1 | Transmit buffer occupied locations are equal or above the THRSH2_REG value). Writing 1 to this bit clears the bit. |
| 9 | XEOF | | Transmit End Of Frame (XEOF is set to 1 when a complete frame was transmitted). Writing 1 to this bit clears the bit. |
| | | 0 | Complete frame was NOT transmitted. |
| | | 1 | Complete frame was transmitted; Writing 1 to this bit clears the bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-24. Interrupt Status Register (IRQSTATUS) Field Descriptions (continued)**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 8 | XFSX | | Transmit Frame Synchronization (XFSX bit is set to 1 when a new transmit frame synchronization is asserted). Writing 1 to this bit clears the bit. |
| | | 0 | New transmit frame synchronization is NOT asserted. |
| | | 1 | New transmit frame synchronization is asserted; Writing 1 to this bit clears the bit. |
| 7 | XSYNCERR | | Transmit Frame Synchronization Error (XSYNCERR is set to 1 when a transmit frame synchronization error is detected). Writing 1 to this bit clears the bit. |
| | | 0 | Transmit frame synchronization error is NOT detected. |
| | | 1 | Transmit frame synchronization error is detected. Writing 1 to this bit clears the bit. |
| 6 | Reserved | 0 | Reserved. |
| 5 | ROVFLSTAT | | Receive Buffer Overflow (ROVFLSTAT bit is set to 1 when receive buffer overflow; the data which is written while overflow condition is discarded). Writing 1 to this bit clears the bit. |
| | | 0 | Receive buffer NOT overflow. |
| | | 1 | Receive buffer overflow; Writing 1 to this bit clears the bit. |
| 4 | RUNDFLSTAT | | Receive Buffer Underflow (RUNDFLSTAT bit is set to 1 when read operation is performed to the receive data register while receive buffer is empty; data read while underflow condition is undefined). Writing 1 to this bit clears the bit. |
| | | 0 | Read operation is performed to the receive data register while receive buffer is NOT empty. |
| | | 1 | Read operation is performed to the receive data register while receive buffer is empty; Writing 1 to this bit clears the bit. |
| 3 | RRDY | | Receive Buffer Threshold Reached (RRDY bit is set to 1 when the receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit clears the bit. |
| | | 0 | Receive buffer occupied locations are below the THRSH1_REG value). |
| | | 1 | Receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit clears the bit. |
| 2 | REOF | | Receive End Of Frame (REOF is set to 1 when a complete frame was received). Writing 1 to this bit clears the bit. |
| | | 0 | Complete frame was NOT received. |
| | | 1 | Complete frame was received; Writing 1 to this bit clears the bit. |
| 1 | RFSR | | Receive Frame Synchronization (RFSR bit is set to 1 when a new receive frame synchronization is asserted). Writing 1 to this bit clears the bit. |
| | | 0 | New receive frame synchronization is NOT asserted. |
| | | 1 | New receive frame synchronization is asserted; Writing 1 to this bit clears the bit. |
| 0 | RSYNCERR | | Receive Frame Synchronization Error (RSYNCERR is set to 1 when a receive frame synchronization error is detected). Writing 1 to this bit clears the bit. |
| | | 0 | Receive frame synchronization error is NOT detected. |
| | | 1 | Receive frame synchronization error is detected. Writing 1 to this bit clears the bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.6 Interrupt Enable Set Register (IRQENABLE_SET)

All 1-bit fields enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit will enable the field. Writing a 0 will have no effect, for example, the register value will not be modified.

Interrupts can either be serviced using the highlander standard IRQSTATUS, IRQSTATUS_RAW, IRQENABLE_SET, and IRQENABLE_CLR set of registers, or the legacy IRQSTATUS_REG and IRQENABLE_REG registers, but not both in the same system.

The Interrupt Enable Set (IRQENABLE_SET) register is shown in Figure 17-38 and described in Table 17-25.

**Figure 17-38. Interrupt Enable Set Register (IRQENABLE_SET)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XEMPTYEOF | Reserved | XOVFLSTAT | XUNDFLSTAT | XRDY | XEOF | XFSX |
| R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| XSYNCERR | Reserved | ROVFLSTAT | RUNDFLSTAT | RRDY | REOF | RFSR | RSYNCERR |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-25. Interrupt Enable Set Register (IRQENABLE_SET) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. |
| 14 | XEMPTYEOF | | Transmit Buffer Empty at end of frame (XEMPTYEOF is set to 1 when a complete frame was transmitted and the transmit buffer is empty). |
| | | 0 | XEMPTYEOF is NOT set to when a complete frame was transmitted and the transmit buffer is empty. |
| | | 1 | XEMPTYEOF is set to 1 when a complete frame was transmitted and the transmit buffer is empty. Writing 1 to this bit clears the bit. |
| 13 | Reserved | 0 | Reserved. |
| 12 | XOVFLSTAT | | Transmit Buffer Overflow (XOVFLSTAT bit is set to 1 when transmit buffer overflow; the data which is written while overflow condition is discarded). Writing 1 to this bit clears the bit. |
| | | 0 | Transmit buffer NOT overflow. |
| | | 1 | Transmit buffer overflow; Writing 1 to this bit clears the bit. |
| 11 | XUNDFLSTAT | | Transmit Buffer Underflow (XUNDFLSTAT bit is set to 1 when the transmit data buffer is empty new data is required to be transmitted). Writing 1 to this bit clears the bit. |
| | | 0 | The transmit data buffer is NOT empty new data is required to be transmitted. |
| | | 1 | The transmit data buffer is empty new data is required to be transmitted. Writing 1 to this bit clears the bit. |
| 10 | XRDY | | Transmit Buffer Threshold Reached (XRDY bit is set to 1 when the transmit buffer free locations are equal or above the THRSH2_REG value). Writing 1 to this bit clears the bit. |
| | | 0 | Transmit buffer occupied locations are below the THRSH2_REG value). |
| | | 1 | Transmit buffer occupied locations are equal or above the THRSH2_REG value). Writing 1 to this bit clears the bit. |
| 9 | XEOF | | Transmit End Of Frame (XEOF is set to 1 when a complete frame was transmitted). Writing 1 to this bit clears the bit. |
| | | 0 | Complete frame was NOT transmitted. |
| | | 1 | Complete frame was transmitted; Writing 1 to this bit clears the bit. |
| 8 | XFSX | | Transmit Frame Synchronization (XFSX bit is set to 1 when a new transmit frame synchronization is asserted). Writing 1 to this bit clears the bit. |
| | | 0 | New transmit frame synchronization is NOT asserted. |
| | | 1 | New transmit frame synchronization is asserted; Writing 1 to this bit clears the bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-25. Interrupt Enable Set Register (IRQENABLE_SET) Field Descriptions (continued)**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 7 | XSYNCERR | | Transmit Frame Synchronization Error (XSYNCERR is set to 1 when a transmit frame synchronization error is detected). Writing 1 to this bit clears the bit. |
| | | 0 | Transmit frame synchronization error is NOT detected. |
| | | 1 | Transmit frame synchronization error is detected. Writing 1 to this bit clears the bit. |
| 6 | Reserved | 0 | Reserved. |
| 5 | ROVFLSTAT | | Receive Buffer Overflow (ROVFLSTAT bit is set to 1 when receive buffer overflow; the data which is written while overflow condition is discarded). Writing 1 to this bit clears the bit. |
| | | 0 | Receive buffer NOT overflow. |
| | | 1 | Receive buffer overflow; Writing 1 to this bit clears the bit. |
| 4 | RUNDFLSTAT | | Receive Buffer Underflow (RUNDFLSTAT bit is set to 1 when read operation is performed to the receive data register while receive buffer is empty; data read while underflow condition is undefined). Writing 1 to this bit clears the bit. |
| | | 0 | Read operation is performed to the receive data register while receive buffer is NOT empty. |
| | | 1 | Read operation is performed to the receive data register while receive buffer is empty; Writing 1 to this bit clears the bit. |
| 3 | RRDY | | Receive Buffer Threshold Reached (RRDY bit is set to 1 when the receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit clears the bit. |
| | | 0 | Receive buffer occupied locations are below the THRSH1_REG value). |
| | | 1 | Receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit clears the bit. |
| 2 | REOF | | Receive End Of Frame (REOF is set to 1 when a complete frame was received). Writing 1 to this bit clears the bit. |
| | | 0 | Complete frame was NOT received. |
| | | 1 | Complete frame was received; Writing 1 to this bit clears the bit. |
| 1 | RFSR | | Receive Frame Synchronization (RFSR bit is set to 1 when a new receive frame synchronization is asserted). Writing 1 to this bit clears the bit. |
| | | 0 | New receive frame synchronization is NOT asserted. |
| | | 1 | New receive frame synchronization is asserted; Writing 1 to this bit clears the bit. |
| 0 | RSYNCERR | | Receive Frame Synchronization Error (RSYNCERR is set to 1 when a receive frame synchronization error is detected). Writing 1 to this bit clears the bit. |
| | | 0 | Receive frame synchronization error is NOT detected. |
| | | 1 | Receive frame synchronization error is detected. Writing 1 to this bit clears the bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.7 Interrupt Enable Clear Register (IRQENABLE_CLR)

All 1-bit fields clear a specific interrupt event. Writing a 1 to a bit will disable the interrupt field. Writing a 0 will have no effect, for example, the register value will not be modified.

Interrupts can either be serviced using the highlander standard IRQSTATUS, IRQSTATUS_RAW, IRQENABLE_SET, and IRQENABLE_CLR set of registers, or the legacy IRQSTATUS_REG and IRQENABLE_REG registers, but not both in the same system.

The Interrupt Enable Clear (IRQENABLE_CLR) register is shown in Figure 17-39 and described in Table 17-26.

**Figure 17-39. Interrupt Enable Clear Register (IRQENABLE_CLR)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | XEMPTYEOF | Reserved | XOVFLSTAT | XUNDFLSTAT | XRDY | XEOF | XFSX |
| R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| XSYNCERR | Reserved | ROVFLSTAT | RUNDFLSTAT | RRDY | REOF | RFSR | RSYNCERR |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-26. Interrupt Enable Clear Register (IRQENABLE_CLR) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. |
| 14 | XEMPTYEOF | | Transmit Buffer Empty at end of frame (XEMPTYEOF is set to 1 when a complete frame was transmitted and the transmit buffer is empty). |
| | | 0 | XEMPTYEOF is NOT set to when a complete frame was transmitted and the transmit buffer is empty. |
| | | 1 | XEMPTYEOF is set to 1 when a complete frame was transmitted and the transmit buffer is empty. Writing 1 to this bit clears the bit. |
| 13 | Reserved | 0 | Reserved. |
| 12 | XOVFLSTAT | | Transmit Buffer Overflow (XOVFLSTAT bit is set to 1 when transmit buffer overflow; the data which is written while overflow condition is discarded). Writing 1 to this bit clears the bit. |
| | | 0 | Transmit buffer NOT overflow. |
| | | 1 | Transmit buffer overflow; Writing 1 to this bit clears the bit. |
| 11 | XUNDFLSTAT | | Transmit Buffer Underflow (XUNDFLSTAT bit is set to 1 when the transmit data buffer is empty new data is required to be transmitted). Writing 1 to this bit clears the bit. |
| | | 0 | The transmit data buffer is NOT empty new data is required to be transmitted. |
| | | 1 | The transmit data buffer is empty new data is required to be transmitted. Writing 1 to this bit clears the bit. |
| 10 | XRDY | | Transmit Buffer Threshold Reached (XRDY bit is set to 1 when the transmit buffer free locations are equal or above the THRSH2_REG value). Writing 1 to this bit clears the bit. |
| | | 0 | Transmit buffer occupied locations are below the THRSH2_REG value). |
| | | 1 | Transmit buffer occupied locations are equal or above the THRSH2_REG value). Writing 1 to this bit clears the bit. |
| 9 | XEOF | | Transmit End Of Frame (XEOF is set to 1 when a complete frame was transmitted). Writing 1 to this bit clears the bit. |
| | | 0 | Complete frame was NOT transmitted. |
| | | 1 | Complete frame was transmitted; Writing 1 to this bit clears the bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-26. Interrupt Enable Clear Register (IRQENABLE_CLR) Field Descriptions (continued)**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 8 | XFSX | | Transmit Frame Synchronization (XFSX bit is set to 1 when a new transmit frame synchronization is asserted). Writing 1 to this bit clears the bit. |
| | | 0 | New transmit frame synchronization is NOT asserted. |
| | | 1 | New transmit frame synchronization is asserted; Writing 1 to this bit clears the bit. |
| 7 | XSYNCERR | | Transmit Frame Synchronization Error (XSYNCERR is set to 1 when a transmit frame synchronization error is detected). Writing 1 to this bit clears the bit. |
| | | 0 | Transmit frame synchronization error is NOT detected. |
| | | 1 | Transmit frame synchronization error is detected. Writing 1 to this bit clears the bit. |
| 6 | Reserved | 0 | Reserved. |
| 5 | ROVFLSTAT | | Receive Buffer Overflow (ROVFLSTAT bit is set to 1 when receive buffer overflow; the data which is written while overflow condition is discarded). Writing 1 to this bit clears the bit. |
| | | 0 | Receive buffer NOT overflow. |
| | | 1 | Receive buffer overflow; Writing 1 to this bit clears the bit. |
| 4 | RUNDFLSTAT | | Receive Buffer Underflow (RUNDFLSTAT bit is set to 1 when read operation is performed to the receive data register while receive buffer is empty; data read while underflow condition is undefined). Writing 1 to this bit clears the bit. |
| | | 0 | Read operation is performed to the receive data register while receive buffer is NOT empty. |
| | | 1 | Read operation is performed to the receive data register while receive buffer is empty; Writing 1 to this bit clears the bit. |
| 3 | RRDY | | Receive Buffer Threshold Reached (RRDY bit is set to 1 when the receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit clears the bit. |
| | | 0 | Receive buffer occupied locations are below the THRSH1_REG value). |
| | | 1 | Receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit clears the bit. |
| 2 | REOF | | Receive End Of Frame (REOF is set to 1 when a complete frame was received). Writing 1 to this bit clears the bit. |
| | | 0 | Complete frame was NOT received. |
| | | 1 | Complete frame was received; Writing 1 to this bit clears the bit. |
| 1 | RFSR | | Receive Frame Synchronization (RFSR bit is set to 1 when a new receive frame synchronization is asserted). Writing 1 to this bit clears the bit. |
| | | 0 | New receive frame synchronization is NOT asserted. |
| | | 1 | New receive frame synchronization is asserted; Writing 1 to this bit clears the bit. |
| 0 | RSYNCERR | | Receive Frame Synchronization Error (RSYNCERR is set to 1 when a receive frame synchronization error is detected). Writing 1 to this bit clears the bit. |
| | | 0 | Receive frame synchronization error is NOT detected. |
| | | 1 | Receive frame synchronization error is detected. Writing 1 to this bit clears the bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.8 DMA Rx Enable Set Register (DMARXENABLE_SET)

The 1-bit field enables a Receive DMA request. Writing a 1 to this field will set it to 1. Writing a 0 will have no effect, for example, the register value is not modified. Note that the RCCR_REG.RDMAEN field is the global (slave) DMA enabler, and that it is disabled by default. The field RCCR_REG.RDMAEN should also be set to 1 to enable a Receive DMA Request.

The DMA Rx Enable Set (DMARXENABLE_SET) register is shown in Figure 17-40 and described in Table 17-27.

**Figure 17-40. DMA Rx Enable Set Register (DMARXENABLE_SET)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | DMARX_ENABLE_SET |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-27. DMA Rx Enable Set Register (DMARXENABLE_SET) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | DMARX_ENABLE_SET | | Receive DMA channel enable set. |

### 17.3.9 DMA Tx Enable Set Register (DMATXENABLE_SET)

The 1-bit field enables a Transmit DMA request. Writing a 1 to this field will set it to 1. Writing a 0 will have no effect, for example, the register value is not modified. Note that the XCCR_REG.XDMAEN field is the global (slave) DMA enabler, and that it is disabled by default. The field XCCR_REG.XDMAEN should also be set to 1 to enable a Transmit DMA Request.

The DMA Tx Enable Set (DMATXENABLE_SET) register is shown in Figure 17-41 and described in Table 17-28.

**Figure 17-41. DMA Tx Enable Set Register (DMATXENABLE_SET)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | DMATX_ENABLE_SET |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-28. DMA Tx Enable Set Register (DMATXENABLE_SET) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | DMATX_ENABLE_SET | | Transmit DMA channel enable set. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.10 DMA Rx Enable Clear Register (DMARXENABLE_CLR)

The 1-bit field disables a Receive DMA request. Writing a 1 to a bit will clear it to 0. Another result of setting to '1' the DMARX_ENABLE_CLEAR field, is the reset of the DMA RX request and wakeup lines. Writing a 0 will have no effect, for example, the register value is not modified.

The DMA Rx Enable Clear (DMARXENABLE_CLR) register is shown in Figure 17-42 and described in Table 17-29.

**Figure 17-42. DMA Rx Enable Clear Register (DMARXENABLE_CLR)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | DMARX_ENABLE_CLEAR |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-29. DMA Rx Enable Clear Register (DMARXENABLE_CLR) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | DMARX_ENABLE_CLEAR | | Receive DMA channel enable clear. |

### 17.3.11 DMA Tx Enable Clear Register (DMATXENABLE_CLR)

The 1-bit field disables a Transmit DMA request. Writing a 1 to a bit will clear it to 0. Another result of setting to '1' the DMATX_ENABLE_CLEAR field, is the reset of the DMA TX request and wakeup lines. Writing a 0 will have no effect, for example, the register value is not modified.

The DMA Tx Enable Clear (DMATXENABLE_CLR) register is shown in Figure 17-43 and described in Table 17-30.

**Figure 17-43. DMA Tx Enable Clear Register (DMATXENABLE_CLR)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | DMATX_ENABLE_CLEAR |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-30. DMA Tx Enable Clear Register (DMATXENABLE_CLR) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. |
| 0 | DMATX_ENABLE_CLEAR | | Transmit DMA channel enable clear. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.12 DMA Rx Wake Enable Register (DMARXWAKE_EN)

All 1-bit fields listed above enable a specific (synchronous) DMA request source to generate an asynchronous wakeup (on the appropriate wakeup line). Note that the SYSCONFIG_REG.ENAWAKEUP field is the global (slave) wakeup enabler, and that it is disabled by default.

The DMA Rx Wake Enable (DMARXWAKE_EN) register is shown in Figure 17-44 and described in Table 17-31.

**Figure 17-44. DMA Rx Wake Enable Register (DMARXWAKE_EN)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | XEMPTYEOFEN | Reserved | | | XRDYEN | XEOFEN | XFSXEN |
| R-0 | R/W-0 | R-0 | | | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| XSYNCERREN | Reserved | | | RRDYEN | REOFEN | RFSREN | RSYNCERREN |
| R/W-0 | R-0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-31. DMA Rx Wake Enable Register (DMARXWAKE_EN) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. |
| 14 | XEMPTYEOFEN | | Transmit Buffer Empty at End Of Frame enable bit. |
| | | 0 | Transmit Buffer Empty at End Of Frame WK enable is NOT active. |
| | | 1 | Transmit Buffer Empty at End Of Frame WK enable is active. |
| 13-11 | Reserved | 0 | Reserved. |
| 10 | XRDYEN | | Transmit Buffer Threshold Reached WK enable bit. |
| | | 0 | Transmit Buffer Threshold WK enable is NOT active. |
| | | 1 | Transmit Buffer Threshold WK enable is active. |
| 9 | XEOFEN | | Transmit End Of Frame WK enable bit. |
| | | 0 | Transmit End Of Frame WK enable is NOT active. |
| | | 1 | Transmit End Of Frame WK enable is active. |
| 8 | XFSXEN | | Transmit Frame Synchronization WK enable bit. |
| | | 0 | Transmit Frame Synchronization WK enable is NOT active. |
| | | 1 | Transmit Frame Synchronization WK enable is active. |
| 7 | XSYNCERREN | | Transmit Frame Synchronization Error WK enable bit. |
| | | 0 | Transmit Frame Synchronization Error WK enable is NOT active. |
| | | 1 | Transmit Frame Synchronization Error WK enable is active. |
| 6-4 | Reserved | 0 | Reserved. |
| 3 | RRDYEN | | Receive Buffer Threshold wakeup enable bit. |
| | | 0 | Receive Buffer Threshold WK enable is NOT active. |
| | | 1 | Receive Buffer Threshold WK enable is active. |
| 2 | REOFEN | | Receive End Of Frame WK enable bit. |
| | | 0 | Receive End Of Frame WK enable is NOT active. |
| | | 1 | Receive End Of Frame WK enable is active |
| 1 | RFSREN | | Receive Frame Synchronization WK enable bit. |
| | | 0 | Receive Frame Synchronization WK enable is NOT active. |
| | | 1 | Receive Frame Synchronization WK enable is active. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-31. DMA Rx Wake Enable Register (DMARXWAKE_EN) Field Descriptions (continued)**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 0 | RSYNCERREN | | Receive Frame Synchronization Error WK enable bit. |
| | | 0 | Receive Frame Synchronization Error WK enable is NOT active. |
| | | 1 | Receive Frame Synchronization Error WK enable is active. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.13 DMA Tx Wake Enable Register (DMATXWAKE_EN)

All 1-bit fields listed above enable a specific (synchronous) DMA request source to generate an asynchronous wakeup (on the appropriate wakeup line). Note that the SYSCONFIG_REG.ENAWAKEUP field is the global (slave) wakeup enabler, and that it is disabled by default.

The DMA Tx Wake Enable (DMATXWAKE_EN) register is shown in Figure 17-45 and described in Table 17-32.

**Figure 17-45. DMA Tx Wake Enable Register (DMATXWAKE_EN)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | XEMPTYEOFEN | Reserved | | | XRDYEN | XEOFEN | XFSXEN |
| R-0 | R/W-0 | R-0 | | | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| XSYNCERREN | Reserved | | | RRDYEN | REOFEN | RFSREN | RSYNCERREN |
| R/W-0 | R-0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-32. DMA Tx Wake Enable Register (DMATXWAKE_EN) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. |
| 14 | XEMPTYEOFEN | | Transmit Buffer Empty at End Of Frame enable bit. |
| | | 0 | Transmit Buffer Empty at End Of Frame WK enable is NOT active. |
| | | 1 | Transmit Buffer Empty at End Of Frame WK enable is active. |
| 13-11 | Reserved | 0 | Reserved. |
| 10 | XRDYEN | | Transmit Buffer Threshold Reached WK enable bit. |
| | | 0 | Transmit Buffer Threshold WK enable is NOT active. |
| | | 1 | Transmit Buffer Threshold WK enable is active. |
| 9 | XEOFEN | | Transmit End Of Frame WK enable bit. |
| | | 0 | Transmit End Of Frame WK enable is NOT active. |
| | | 1 | Transmit End Of Frame WK enable is active. |
| 8 | XFSXEN | | Transmit Frame Synchronization WK enable bit. |
| | | 0 | Transmit Frame Synchronization WK enable is NOT active. |
| | | 1 | Transmit Frame Synchronization WK enable is active. |
| 7 | XSYNCERREN | | Transmit Frame Synchronization Error WK enable bit. |
| | | 0 | Transmit Frame Synchronization Error WK enable is NOT active. |
| | | 1 | Transmit Frame Synchronization Error WK enable is active. |
| 6-4 | Reserved | 0 | Reserved. |
| 3 | RRDYEN | | Receive Buffer Threshold wakeup enable bit. |
| | | 0 | Receive Buffer Threshold WK enable is NOT active. |
| | | 1 | Receive Buffer Threshold WK enable is active. |
| 2 | REOFEN | | Receive End Of Frame WK enable bit. |
| | | 0 | Receive End Of Frame WK enable is NOT active. |
| | | 1 | Receive End Of Frame WK enable is active. |
| 1 | RFSREN | | Receive Frame Synchronization WK enable bit. |
| | | 0 | Receive Frame Synchronization WK enable is NOT active. |
| | | 1 | Receive Frame Synchronization WK enable is active. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-32. DMA Tx Wake Enable Register (DMATXWAKE_EN) Field Descriptions (continued)**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 0 | RSYNCERREN | | Receive Frame Synchronization Error WK enable bit. |
| | | 0 | Receive Frame Synchronization Error WK enable is NOT active. |
| | | 1 | Receive Frame Synchronization Error WK enable is active. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.14 McBSP Data Receive Register (DRR_REG)

The McBSP_DRR_REG register is shown in Figure 17-46 and described in Table 17-33.

**Figure 17-46. McBSP_DRR_REG**

| 31 | 0 |
|---|---|
| DRR | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 17-33. McBSP_DRR_REG Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-0 | DRR | 0 | Data receive register. |

### 17.3.15 McBSP Data Transmit Register (DXR_REG)

The McBSP_DXR_REG register is shown in Figure 17-47 and described in Table 17-34.

**Figure 17-47. McBSP_DRR_REG**

| 31 | 0 |
|---|---|
| DXR | |
| W-0 | |

LEGEND: W = Write only; -*n* = value after reset

**Table 17-34. McBSP_DXR_REG Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31-0 | DXR | 0 | Data transmit register. |

### 17.3.16 McBSP Serial Port Control Register 2 (SPCR2_REG)

The McBSP_SPCR2_REG register is shown in Figure 17-48 and described in Table 17-35.

**Figure 17-48. McBSP_SPCR2_REG**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 |
|---|---|---|---|
| Reserved | | FREE | SOFT |
| R-0 | | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FRST | GRST | XINTM | | XSYNCERR | XEMPTY | XRDY | XRST |
| R/W-0 | R/W-0 | R/W-0 | | R/W-0 | R-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-35. McBSP_SPCR2_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9 | FREE | | Free Running Mode<br>(When this bit is set, the module ignores the Msuspend input) |
| | | 0 | Free running mode is disabled |
| | | 1 | Free running mode is enabled |
| 8 | SOFT | | Soft bit. |
| | | 0 | SOFT Mode is disabled: the McBSP module stops its activity immediately following MSuspend assertion. |
| | | 1 | SOFT Mode is enabled: the McBSP module freezes its state after completion of the current operation when MSuspend is asserted. |
| 7 | FRST | | Frame-Synchronization Generator Reset. |
| | | 0 | Frame-Synchronization logic is reset. Frame-Synchronization signal FSG is not generated by the sample-rate generator. |
| | | 1 | Frame-Synchronization signal FSG is generated after (FPER+1) number of CLKG clocks; all frame counters are loaded with their programmed values. |
| 6 | GRST | | Sample-Rate Generator Reset. |
| | | 0 | SRG is reset. |
| | | 1 | SRG is pulled out of reset. CLKG is driven as per programmed value in SRG registers (SRGR[1,2]) |
| 5-4 | XINTM | | Transmit Interrupt Mode (legacy). |
| | | 0 | Transmit interrupt is driven by XRDY. |
| | | 1h | Transmit interrupt generated by end-of-frame. |
| | | 2h | Transmit interrupt generated by a new frame synchronization. |
| | | 3h | Transmit interrupt generated by XSYNCERR. |
| 3 | XSYNCERR | | Transmit Synchronization Error.<br>(Writing 0 to this bit clear the legacy transmit interrupt if asserted due to XSYNCERROR condition). |
| | | 0 | No synchronization error. |
| | | 1 | Synchronization error detected by McBSP. |
| 2 | XEMPTY | | Transmit Shift Register XSR Empty. |
| | | 0 | XSR is empty. |
| | | 1 | XSR is not empty. |
| 1 | XRDY | | Transmitter ready. |
| | | 0 | Transmitter is not ready. |
| | | 1 | Transmitter is ready for new data in DXR. |
| 0 | XRST | | Transmitter reset. This resets and enables the transmitter. |
| | | 0 | The serial port transmitter is disabled and in reset state. |
| | | 1 | The serial port transmitter is enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.17 McBSP Serial Port Control Register 1 (SPCR1_REG)

The McBSP_SPCR1_REG register is shown in Figure 17-49 and described in Table 17-36.

**Figure 17-49. McBSP_SPCR1_REG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| ALB | RJUST | | Reserved | | | | |
| R/W-0 | R/W-0 | | R-0 | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DXENA | Reserved | RINTM | | RSYNCERR | RFULL | RRDY | RRST |
| R/W-0 | R-0 | R/W-0 | | R/W-0 | R-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-36. McBSP_SPCR1_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15 | ALB | | Analog loopback Mode. |
| | | 0 | Analog loopback mode disabled. |
| | | 1 | Analog loopback mode enabled. |
| 14-13 | RJUST | | Receive Sign-Extension and Justification Mode. |
| | | 0 | Right-justify and zero-fill MSBs in DRR. |
| | | 1h | Right-justify and sign-extend MSBs in DRR. |
| | | 2h | Left-justify and zero-fill LSBs in DRR. |
| | | 3h | Reserved. |
| 12-8 | Reserved | 0 | Reserved |
| 7 | DXENA | | DX Enabler. |
| | | 0 | DX enabler is off. |
| | | 1 | DX enabler is on. |
| 6 | Reserved | 0 | Reserved |
| 5-4 | RINTM | | Receive Interrupt Mode (legacy). |
| | | 0 | Receive Interrupt driven by RRDY (end of word) and end of frame in A-bis mode. |
| | | 1h | Receive Interrupt generated by end-of-block or end-of-frame in multichannel operation. |
| | | 2h | Receive Interrupt generated by a new frame synchronization. |
| | | 3h | Receive Interrupt generated by RSYNCERR. |
| 3 | RSYNCERR | | Receive Synchronization Error.<br>(Writing 0 to this bit clear the legacy receive interrupt if asserted due to RSYNCERROR condition). |
| | | 0 | No synchronization error. |
| | | 1 | Synchronization error detected by McBSP. |
| 2 | RFULL | | Receive Shift Register (RSR [1,2]) Full. |
| | | 0 | RB is not in overrun condition. |
| | | 1 | DRR is not read, RB is full and RSR is also full with new word. |
| 1 | RRDY | | Receiver Ready. |
| | | 0 | Receiver is not ready. |
| | | 1 | Receiver is ready with data to be read from DRR. |
| 0 | RRST | | Receiver reset. This resets and enables the receiver. |
| | | 0 | The serial port receiver is disabled and in reset state. |
| | | 1 | The serial port receiver is enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.18 McBSP Receive Control Register 2 (RCR2_REG)

The McBSP_RCR2_REG register is shown in Figure 17-50 and described in Table 17-37.

**Figure 17-50. McBSP_RCR2_REG**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 8 | 7 5 | 4 3 | 2 | 1 0 |
|---|---|---|---|---|---|
| RPHASE | RFRLEN2 | RWDLEN2 | RREVERSE | Reserved | RDATDLY |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-37. McBSP_RCR2_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15 | RPHASE | | Receive Phases. |
| | | 0 | Single-phase frame. |
| | | 1 | Dual-phase frame. |
| 14-8 | RFRLEN2 | | Receive Frame Length 2 |
| | | | Single-phase frame selected: RFRLEN2 = don't care. |
| | | | Dual-phase frame selected: RFRLEN2 = 000 0000 - 1 word per second phase (other values are reserved). |
| 7-5 | RWDLEN2 | | Receive Word Length 2. |
| | | 0 | 8 bits |
| | | 1h | 12 bits |
| | | 2h | 16 bits |
| | | 3h | 20 bits |
| | | 4h | 24 bits |
| | | 5h | 32 bits |
| | | 6h-7h | Reserved (do not use) |
| 4-3 | RREVERSE | | Receive reverse mode. |
| | | 0 | Data transfer starts with MSB first. |
| | | 1h | Data transfer starts with LSB first. |
| | | 2h-3h | Reserved (do not use) |
| 2 | Reserved | 0 | Reserved |
| 1-0 | RDATDLY | | Receive Data Delay |
| | | 0 | 0-bit data delay |
| | | 1h | 1-bit data delay |
| | | 2h | 2-bit data delay |
| | | 3h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.19 McBSP Receive Control Register 1 (RCR1_REG)

The McBSP_RCR1_REG register is shown in Figure 17-51 and described in Table 17-38.

**Figure 17-51. McBSP_RCR1_REG**

| 31 15 | 14 8 | 7 5 | 4 0 |
|---|---|---|---|
| Reserved | RFRLEN1 | RWDLEN1 | Reserved |
| R-0 | R/W-0 | R/W-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 17-38. McBSP_RCR1_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved |
| 14-8 | RFRLEN1 | | Receive Frame Length 1.<br>Single-phase frame selected:<br>RFRLEN1 = 000 0000 - 1 word per frame<br>RFRLEN1 = 000 0001 - 2 words per frame.<br>RFRLEN1 = 111 1111 - 128 words per frame.<br>Dual-phase frame selected: RFRLEN1 = 000 0000 - 1 word per phase (other values are reserved). |
| 7-5 | RWDLEN1 | | Receive Word Length 1 |
| | | 0 | 8 bits |
| | | 1h | 12 bits |
| | | 2h | 16 bits |
| | | 3h | 20 bits |
| | | 4h | 24 bits |
| | | 5h | 32 bits |
| | | 6h-7h | Reserved (do not use). |
| 4-0 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.20 McBSP Transmit Control Register 2 (XCR2_REG)

The McBSP_XCR2_REG register is shown in Figure 17-52 and described in Table 17-39.

**Figure 17-52. McBSP_XCR2_REG**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 8 | 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| XPHASE | XFRLEN2 | | XWDLEN2 | | XREVERSE | | Reserved | XDATDLY | |
| R/W-0 | R/W-0 | | R/W-0 | | R/W-0 | | R-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-39. McBSP_XCR2_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15 | XPHASE | | Transmit Phases. |
| | | 0 | Single-phase frame. |
| | | 1 | Dual-phase frame. |
| 14-8 | XFRLEN2 | | Transmit Frame Length 2. |
| | | | Single-phase frame selected: XFRLEN2 = don't care. |
| | | | Dual-phase frame selected: XFRLEN2 = 000 0000 - 1 word per second phase (other values are reserved). |
| 7-5 | XWDLEN2 | | Transmit Word Length 2 |
| | | 0 | 8 bits |
| | | 1h | 12 bits |
| | | 2h | 16 bits |
| | | 3h | 20 bits |
| | | 4h | 24 bits |
| | | 5h | 32 bits |
| | | 6h-7h | Reserved (do not use) |
| 4-3 | XREVERSE | | Transmit reverse mode. |
| | | 0 | Data transfer starts with MSB first. |
| | | 1h | Data transfer starts with LSB first. |
| | | 2h-3h | Reserved (do not use). |
| 2 | Reserved | 0 | Reserved |
| 1-0 | XDATDLY | | Transmit Data Delay |
| | | 0 | 0-bit data delay |
| | | 1h | 1-bit data delay |
| | | 2h | 2-bit data delay |
| | | 3h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.21 McBSP Transmit Control Register 1 (XCR1_REG)

The McBSP_XCR1_REG register is shown in Figure 17-53 and described in Table 17-40.

**Figure 17-53. McBSP_XCR1_REG**

| 31 15 | 14 8 | 7 5 | 4 0 |
|---|---|---|---|
| Reserved | XFRLEN1 | XWDLEN1 | Reserved |
| R-0 | R/W-0 | R/W-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 17-40. McBSP_XCR1_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved |
| 14-8 | XFRLEN1 | | Transmit Frame Length 1<br>Single-phase frame selected:<br>XFRLEN1 = 000 0000 - 1 word per frame.<br>XFRLEN1 = 000 0001 - 2 words per frame.<br>XFRLEN1 = 111 1111 - 128 words per frame.<br>Dual-phase frame selected: XFRLEN1 = 000 0000 - 1 word per phase (other values are reserved). |
| 7-5 | XWDLEN1 | | Transmit Word Length 1 |
| | | 0 | 8 bits |
| | | 1h | 12 bits |
| | | 2h | 16 bits |
| | | 3h | 20 bits |
| | | 4h | 24 bits |
| | | 5h | 32 bits |
| | | 6h-7h | Reserved (do not use) |
| 4-0 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.22 McBSP Sample Rate Generator Register 2 (SRGR2_REG)

The McBSP_SRGR2_REG register is shown in Figure 17-54 and described in Table 17-41.

**Figure 17-54. McBSP_SRGR2_REG**

| 31 | | | | | 16 |
|---|---|---|---|---|---|
| Reserved | | | | | |
| R-0 | | | | | |
| 15 | 14 | 13 | 12 | 11 | 0 |
| GSYNC | CLKSP | CLKSM | FSGM | FPER | |
| R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 17-41. McBSP_SRGR2_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15 | GSYNC | | Sample Rate Generator (SRG) Synchronization |
| | | | Only used when the external clock (CLKS) drives the SRG clock (CLKSM=0). |
| | | 0 | The SRG clock (CLKG) is free running. |
| | | 1 | The SRG clock (CLKG) is running, but CLKG is resynchronized and frame-synchronization signal (FSG) is generated only after detecting the receive frame-synchronization signal (FSR). |
| | | | Also, frame period, FPER, is a don't care because the period is dictated by the external frame-synchronization pulse. |
| 14 | CLKSP | | CLKS Polarity Clock Edge Select |
| | | | Only used when the external clock CLKS drives the SRG clock (CLKSM=0). |
| | | 0 | Rising edge of CLKG and FSG. |
| | | 1 | Falling edge of CLKG and FSG. |
| 13 | CLKSM | | McBSP SRG Clock Mode |
| | | 0 | SCLKME = 0: SRG clock derived from the CLKS pin. |
| | | | SCLKME = 1: SRG clock derived from the the CLKRI pin. |
| | | 1 | SCLKME = 0: SRG clock derived from the OCP clock. |
| | | | SCLKME = 1: SRG clock derived from the CLKXI pin. |
| 12 | FSGM | | Sample Rate Generator Transmit Frame-Synchronization Mode. Used when FSXM=1 in the PCR. |
| | | 0 | Transmit frame-synchronization signal (FSX) is generated when transmit buffer is not empty. |
| | | | When FSGM = 0, FPER and FWID are used to determine the frame synchronization period and width (external FSX is gated by the buffer empty condition). |
| | | 1 | Transmit frame-synchronization signal driven by the SRG frame-synchronization signal, FSG. |
| 11-0 | FPER | 0-FFFh | Frame Period. This value + 1 determines when the next frame-synchronization signal becomes active. |
| | | | Range: 1 to 4096 CLKG periods. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.23 McBSP Sample Rate Generator Register 1 (SRGR1_REG)

The McBSP_SRGR1_REG register is shown in Figure 17-55 and described in Table 17-42.

**Figure 17-55. McBSP_SRGR1_REG**

| 31 16 | 15 8 | 7 0 |
|---|---|---|
| Reserved | FWID | CLKGDV |
| R-0 | R/W-0 | R/W-1 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 17-42. McBSP_SRGR1_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15-8 | FWID | 0-FFh | Frame Width. This value + 1 determines the width of the frame-synchronization pulse, FSG, during its active period.<br>Range: 1 to 256 CLKG periods. |
| 7-0 | CLKGDV | 1-FFh | Sample Rate Generator Clock Divider.<br>This value is used as the divide-down number to generate the required SRG clock frequency.<br>Default value is 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.24 McBSP Multichannel Register 2 (MCR2_REG)

The McBSP_MCR2_REG register is shown in Figure 17-56 and described in Table 17-43.

**Figure 17-56. McBSP_MCR2_REG**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | XMCME | XPBBLK | | XPABLK | | Reserved | | XMCM | |
| R-0 | | R/W-0 | R/W-0 | | R/W-0 | | R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-43. McBSP_MCR2_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9 | XMCME | | XMCME Transmit multichannel partition mode bit 0.<br>XMCME is only applicable if channels can be individually disabled/enabled or masked/unmasked for transmission (XMCM is non-zero).<br>XMCME determines whether only 32 channels or all 128 channels are to be individually selectable. |
| | | 0 | **2-partition mode (legacy only):**<br>Only partitions A and B are used. You can control up to 32 channels in the transmit multichannel selection mode selected with the XMCM bits.<br>If XMCM = 01b or 10b, assign 16 channels to partition A with the XPABLK bits. Assign 16 channels to partition B with the XPBBLK bits.<br>If XMCM = 11b (for symmetric transmission and reception), assign 16 channels to receive partition A with the RPABLK bits. Assign 16 channels to receive partition B with the RPBBLK bits<br>You control the channels with the appropriate transmit channel enable registers:<br>XCERA: Channels in partition A.<br>XCERB: Channels in partition B. |
| | | 1 | **8-partition mode:**<br>All partitions (A through H) are used. You can control up to 128 channels in the transmit multichannel selection mode selected with the XMCM bits.<br>You control the channels with the appropriate transmit channel enable registers:<br>XCERA: Channels 0 through 15<br>XCERB: Channels 16 through 31<br>XCERC: Channels 32 through 47<br>XCERD: Channels 48 through 63<br>XCERE: Channels 64 through 79<br>XCERF: Channels 80 through 95<br>XCERG: Channels 96 through 111<br>XCERH: Channels 112 through 127 |
| 8-7 | XPBBLK | | Transmit Partition B Block (legacy only) |
| | | 0 | Block 1. Channel 16 to channel 31 |
| | | 1h | Block 3. Channel 48 to channel 63 |
| | | 2h | Block 5. Channel 80 to channel 95 |
| | | 3h | Block 7. Channel 112 to channel 127 |
| 6-5 | XPABLK | | Transmit Partition A Block (legacy only) |
| | | 0 | Block 0. Channel 0 to channel 15 |
| | | 1h | Block 2. Channel 32 to channel 47 |
| | | 2h | Block 4. Channel 64 to channel 79 |
| | | 3h | Block 6. Channel 96 to channel 111 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-43. McBSP_MCR2_REG Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 4-2 | Reserved | 0 | Reserved |
| 1-0 | XMCM | | Transmit Multichannel Selection Enable |
| | | 0 | All channels enabled without masking (DX is always driven during transmission of data). |
| | | 1h | All channels disabled and therefore masked by default.<br>Required channels are selected by enabling XP(A/B)BLK and XCER(A/B) appropriately. Also, these selected channels are not masked and therefore DX is always driven. |
| | | 2h | All channels enabled, but masked.<br>Selected channels enabled via XP(A/B)BLK and XCER(A/B) are unmasked. |
| | | 3h | All channels disabled and therefore masked by default.<br>Required channels are selected by enabling RP(A/B)BLK and RCER(A/B) appropriately. Selected channels can be unmasked by RP(A/B)BLK and XCER(A/B).<br>This mode is used for symmetric transmit and receive operation. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.25 McBSP Multichannel Register 1 (MCR1_REG)

The McBSP_MCR1_REG register is shown in Figure 17-57 and described in Table 17-44.

**Figure 17-57. McBSP_MCR1_REG**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | RMCME | RPBBLK | | RPABLK | | Reserved | | RMCM |
| R-0 | | R/W-0 | R/W-0 | | R/W-0 | | R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-44. McBSP_MCR1_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved |
| 9 | RMCME | | RMCME Receive multichannel partition mode bit 0 . Receive multichannel partition mode determines whether only 32 channels or all 128 channels are to be individually selectable.<br>RMCME is only applicable if channels can be individually enabled or disabled for reception (RMCM = 1). |
| | | 0 | **2-partition mode (legacy only).**<br>Only partitions A and B are used. You can control up to 32 channels in the receive multichannel selection mode (RMCM = 1).<br>Assign 16 channels to partition A with the RPABLK bits. Assign 16 channels to partition B with the RPBBLK bits.<br>You can control the channels with the appropriate receive channel enable registers:<br>RCERA: Channels in partition A<br>RCERB: Channels in partition B |
| | | 1 | **8-partition mode:**<br>All partitions (A through H) are used. Control up to 128 channels in the receive multichannel selection mode.<br>You can control the channels with the appropriate receive channel enable registers:<br>RCERA: Channels 0 through 15<br>RCERB: Channels 16 through 31<br>RCERC: Channels 32 through 47<br>RCERD: Channels 48 through 63<br>RCERE: Channels 64 through 79<br>RCERF: Channels 80 through 95<br>RCERG: Channels 96 through 111<br>RCERH: Channels 112 through 127 |
| 8-7 | RPBBLK | | Receive Partition B Block (legacy only) |
| | | 0 | Block 1. Channel 16 to channel 31 |
| | | 1h | Block 3. Channel 48 to channel 63 |
| | | 2h | Block 5. Channel 80 to channel 95 |
| | | 3h | Block 7. Channel 112 to channel 127 |
| 6-5 | RPABLK | | Receive Partition A Block (legacy only) |
| | | 0 | Block 0. Channel 0 to channel 15 |
| | | 1h | Block 2. Channel 32 to channel 47 |
| | | 2h | Block 4. Channel 64 to channel 79 |
| | | 3h | Block 6. Channel 96 to channel 111 |
| 4-1 | Reserved | 0 | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-44. McBSP_MCR1_REG Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | RMCM | | Receive Multichannel Selection Enable |
| | | 0 | All 128 channels |
| | | 1 | All channels disabled by default. |
| | | | Required channels are selected by enabling RP(A/B)BLK and RCER(A/B) appropriately. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.26 McBSP Receive Channel Enable Register Partition A (RCERA_REG)

The McBSP_RCERA_REG register is shown in Figure 17-58 and described in Table 17-45.

**Figure 17-58. McBSP_RCERA_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | RCERA | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-45. McBSP_RCERA_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | RCERA | | Receive Channel Enable.<br>RCERA n = 0. Disables reception of n-th channel in an even-numbered block in partition A.<br>RCERA n = 1. Enables reception of n-th channel in an even-numbered block in partition A. |

### 17.3.27 McBSP Receive Channel Enable Register Partition B (RCERB_REG)

The McBSP_RCERB_REG register is shown in Figure 17-59 and described in Table 17-46.

**Figure 17-59. McBSP_RCERB_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | RCERB | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-46. McBSP_RCERB_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | RCERB | | Receive Channel Enable.<br>RCERB n = 0. Disables reception of n-th channel in a odd-numbered block in partition B.<br>RCERB n = 1. Enables reception of n-th channel in a odd-numbered block in partition B. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.28 McBSP Transmit Channel Enable Register Partition A (XCERA_REG)

The McBSP_XCERA_REG register is shown in Figure 17-60 and described in Table 17-47.

**Figure 17-60. McBSP_XCERA_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | XCERA | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-47. McBSP_XCERA_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15-0 | XCERA | | Transmit Channel Enable.<br>XCERA n = 0. Disables transmission of n-th channel in an even-numbered block in partition A.<br>XCERA n = 1. Enables transmission of n-th channel in an even-numbered block in partition A. |

### 17.3.29 McBSP Transmit Channel Enable Register Partition B (XCERB_REG)

The McBSP_XCERB_REG register is shown in Figure 17-61 and described in Table 17-48.

**Figure 17-61. McBSP_XCERB_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | XCERB | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-48. McBSP_XCERB_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15-0 | XCERB | | Transmit Channel Enable.<br>XCERB n = 0. Disables transmission of n-th channel in an odd-numbered block in partition B.<br>XCERB n = 1. Enables transmission of n-th channel in an odd-numbered block in partition B. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.30 McBSP Pin Control Register (PCR_REG)

The McBSP_PCR_REG register is shown in Figure 17-62 and described in Table 17-49.

**Figure 17-62. McBSP_PCR_REG**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | IDLE_EN | XIOEN | RIOEN | FSXM | FSRM | CLKXM | CLKRM |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SCLKME | CLKS_STAT | DX_STAT | DR_STAT | FSXP | FSRP | CLKXP | CLKRP |
| R/W-0 | R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-49. McBSP_PCR_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. |
| 14 | IDLE_EN | | Idle enable (legacy only). |
| | | | This bit allows stopping all the clocks in the McBSP module. (legacy) |
| | | 0 | The McBSP is running |
| | | 1 | The clocks in the McBSP are shut off when both IDLE_EN = 1 and this power domain is in idle mode (Force idle or Smart idle) |
| 13 | XIOEN | | Transmit General Purpose I/O Mode only when XRST = 0 in SPCR[1,2]. (legacy only) |
| | | 0 | DX, FSX and CLKX are configured as serial port pins and do not function as general-purpose I/Os. |
| | | 1 | DX pin is a general purpose output. FSX and CLKX are general purpose I/Os. These serial port pins do not perform serial port operation. |
| 12 | RIOEN | | Receive General Purpose I/O Mode when RRST = 0 in SPCR[1,2]. (legacy) |
| | | 0 | DR, FSR, CLKR and CLKS are configured as serial port pins and do not function as general-purpose I/Os. |
| | | 1 | DR and CLKS pins are general purpose inputs; FSR and CLKR are general purpose I/Os. These serial port pins do not perform serial port operation. The CLKS pin is affected by a combination of RRST and RIOEN signals of the receiver. |
| 11 | FSXM | | Transmit Frame-Synchronization Mode. |
| | | 0 | Frame-synchronization signal derived from an external source. |
| | | 1 | Frame synchronization is determined by the SRG frame-synchronization mode bit FSGM in SRGR2. |
| 10 | FSRM | | Receive Frame-Synchronization Mode. |
| | | 0 | Frame-Synchronization pulses generated by an external device. FSR is an input pin. |
| | | 1 | Frame synchronization generated internally by SRG. FSR is an output pin except when GSYNC=1 in SRGR. |
| 9 | CLKXM | | Transmitter Clock Mode. |
| | | | When digital loop-back mode set (McBSPi.McBSP_XCCR_REG[5] DLB = 1), CLKXM bit is ignored. The internal transmit clock (not the mcbspi_clkx pin) is driven by the internal SRG and mcbspi_clkx pin is in high-impedance. |
| | | 0 | Transmitter clock is driven by an external clock with CLKX as an input pin. |
| | | 1 | CLKX is an output pin and is driven by the internal SRG. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-49. McBSP_PCR_REG Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 8 | CLKRM | | Receiver Clock Mode. |
| | | 0 | Case 1: Analog loop back mode not set (ALB = 0) in SPCR1: |
| | | | Receive clock (CLKR) is an input driven by an external clock. |
| | | | Case 2: Analog loop back mode set (ALB = 1) in SPCR1: |
| | | | Receive clock (not the CLKR pin) is driven by transmit clock (CLKX), which is based on the CLKXM bit in the PCR. CLKR pin is in high-impedance. |
| | | 1 | Case 1: Analog loop back mode not set (ALB = 0) in SPCR1: |
| | | | CLKR is an output pin and is driven by the internal sample rate generator. |
| | | | Case 2: Analog loop back mode set (ALB = 1) in SPCR1: |
| | | | CLKR is an output pin and is driven by the transmit clock. The transmit clock is derived based on the CLKRM bit in the PCR. |
| 7 | SCLKME | | SCLKME Sample rate generator input clock mode bit 0. |
| | | | The sample rate generator can produce a clock signal, CLKG. The frequency of CLKG is: |
| | | | CLKG frequency = (Input clock frequency) / (CLKGDV + 1) |
| | | | SCLKME is used in conjunction with the CLKSM bit to select the input clock: |
| | | 0 | CLKSM = 0: Signal on CLKS pin. |
| | | | CLKSM = 1: McBSP_OCP clock. |
| | | 1 | CLKSM = 0: Signal on CLKR pin. |
| | | | CLKSM = 1: Signal on CLKX pin. |
| 6 | CLKS_STAT | | CLKS pin status. Reflects value on CLKS pin when selected as a general purpose input. (legacy) |
| | | 0 | The signal on the CLKS pin is low. |
| | | 1 | The signal on the CLKS pin is high. |
| 5 | DX_STAT | | DX pin status. Reflects value driven on to DX pin when selected as a general purpose output. (legacy) |
| | | 0 | Drive the signal on the DX pin low. |
| | | 1 | Drive the signal on the DX pin high. |
| 4 | DR_STAT | | DR pin status. Reflects value on DR pin when selected as a general purpose input. (legacy) |
| | | 0 | The signal on DR pin is low. |
| | | 1 | The signal on DR pin is high. |
| 3 | FSXP | | Transmit Frame-Synchronization Polarity. |
| | | 0 | Frame-synchronization pulse FSX is active high. |
| | | 1 | Frame-synchronization pulse FSX is active low. |
| 2 | FSRP | | Receive Frame-Synchronization Polarity. |
| | | 0 | Frame-synchronization pulse FSR is active high. |
| | | 1 | Frame-synchronization pulse FSR is active low. |
| 1 | CLKXP | | Transmit Clock Polarity. |
| | | 0 | Transmit data driven on rising edge of CLKX. |
| | | 1 | Transmit data driven on falling edge of CLKX. |
| 0 | CLKRP | | Receive Clock Polarity. |
| | | 0 | Receive data sampled on falling edge of CLKR. |
| | | 1 | Receive data sampled on rising edge of CLKR. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.31 McBSP Receive Channel Enable Register Partition C (RCERC_REG)

The McBSP_RCERC_REG register is shown in Figure 17-63 and described in Table 17-50.

**Figure 17-63. McBSP_RCERC_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | RCERC | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-50. McBSP_RCERC_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | RCERC | | Receive Channel Enable.<br>RCERC n = 0. Disables reception of n-th channel in block 2 in partition C.<br>RCERC n = 1. Enables reception of n-th channel in block 2 in partition C. |

### 17.3.32 McBSP Receive Channel Enable Register Partition D (RCERD_REG)

The McBSP_RCERD_REG register is shown in Figure 17-64 and described in Table 17-51.

**Figure 17-64. McBSP_RCERD_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | RCERD | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-51. McBSP_RCERD_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | | | Receive Channel Enable.<br>RCERD n = 0. Disables reception of n-th channel in block 3 in partition D.<br>RCERD n = 1. Enables reception of n-th channel in block 3 in partition D. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.33 McBSP Transmit Channel Enable Register Partition C (XCERC_REG)

The McBSP_XCERC_REG register is shown in Figure 17-65 and described in Table 17-52.

**Figure 17-65. McBSP_XCERC_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | XCERC | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-52. McBSP_XCERC_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | XCERC | | Transmit Channel Enable.<br>XCERC n = 0. Disables transmission of n-th channel in block 2 in partition C.<br>XCERC n = 1. Enables transmission of n-th channel in block 2 in partition C. |

### 17.3.34 McBSP Transmit Channel Enable Register Partition D (XCERD_REG)

The McBSP_XCERD_REG register is shown in Figure 17-66 and described in Table 17-53.

**Figure 17-66. McBSP_XCERD_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | XCERD | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-53. McBSP_XCERD_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | XCERD | | Transmit Channel Enable.<br>XCERD n = 0. Disables transmission of n-th channel in block 3 in partition D.<br>XCERD n = 1. Enables transmission of n-th channel in block 3 in partition D. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.35 McBSP Receive Channel Enable Register Partition E (RCERE_REG)

The McBSP_RCERE_REG register is shown in Figure 17-67 and described in Table 17-54.

**Figure 17-67. McBSP_RCERE_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | RCERE | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-54. McBSP_RCERE_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | RCERE | | Receive Channel Enable.<br>RCERE n = 0. Disables reception of n-th channel in block 4 in partition E.<br>RCERE n = 1. Enables reception of n-th channel in block 4 in partition E. |

### 17.3.36 McBSP Receive Channel Enable Register Partition F (RCERF_REG)

The McBSP_RCERF_REG register is shown in Figure 17-68 and described in Table 17-55.

**Figure 17-68. McBSP_RCERF_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | RCERF | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-55. McBSP_RCERF_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | RCERF | | Receive Channel Enable.<br>RCERF n = 0. Disables reception of n-th channel in block 5 in partition F.<br>RCERF n = 1. Enables reception of n-th channel in block 5 in partition F. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.37 McBSP Transmit Channel Enable Register Partition E (XCERE_REG)

The McBSP_XCERE_REG register is shown in Figure 17-69 and described in Figure 24-35.

**Figure 17-69. McBSP_XCERE_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | XCERE | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-56. McBSP_XCERE_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | XCERE | | Transmit Channel Enable.<br>XCERE n = 0. Disables transmission of n-th channel in block 4 in partition E.<br>XCERE n = 1. Enables transmission of n-th channel in block 4 in partition E. |

### 17.3.38 McBSP Transmit Channel Enable Register Partition F (XCERF_REG)

The McBSP_XCERF_REG register is shown in Figure 17-70 and described in Table 17-57.

**Figure 17-70. McBSP_XCERF_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | XCERF | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-57. McBSP_XCERF_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | XCERF | | Transmit Channel Enable.<br>XCERF n = 0. Disables transmission of n-th channel in block 5 in partition F.<br>XCERF n = 1. Enables transmission of n-th channel in block 5 in partition F. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.39 McBSP Receive Channel Enable Register Partition G (RCERG_REG)

The McBSP_RCERG_REG register is shown in Figure 17-71 and described in Table 17-58.

**Figure 17-71. McBSP_RCERG_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | RCERG | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-58. McBSP_RCERG_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | RCERG | | Receive Channel Enable.<br>RCERG n = 0. Disables reception of n-th channel in block 6 in partition G.<br>RCERG n = 1. Enables reception of n-th channel in block 6 in partition G. |

### 17.3.40 McBSP Receive Channel Enable Register Partition H (RCERH_REG)

The McBSP_RCERH_REG register is shown in Figure 17-72 and described in Table 17-59.

**Figure 17-72. McBSP_RCERH_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | RCERH | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-59. McBSP_RCERH_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | RCERH | | Receive Channel Enable.<br>RCERH n = 0. Disables reception of n-th channel in block 7 in partition H.<br>RCERH n = 1. Enables reception of n-th channel in block 7 in partition H. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.41 McBSP Transmit Channel Enable Register Partition G (XCERG_REG)

The McBSP_XCERG_REG register is shown in Figure 17-73 and described in Table 17-60.

**Figure 17-73. McBSP_XCERG_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | XCERG | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-60. McBSP_XCERG_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | XCERG | | Transmit Channel Enable.<br>XCERG n = 0. Disables transmission of n-th channel in block in partition G.<br>XCERG n = 1. Enables transmission of n-th channel in block in partition G. |

### 17.3.42 McBSP Transmit Channel Enable Register Partition H (XCERH_REG)

The McBSP_XCERH_REG register is shown in Figure 17-74 and described in Table 17-61.

**Figure 17-74. McBSP_XCERH_REG**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | XCERH | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-61. McBSP_XCERH_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-0 | XCERH | | Transmit Channel Enable.<br>XCERH n = 0. Disables transmission of n-th channel in block in partition H.<br>XCERH n = 1. Enables transmission of n-th channel in block in partition H. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.43 McBSP Transmit Buffer Threshold Register (DMA or IRQ Trigger) (THRSH2_REG)

The McBSP_THRSH2_REG register is shown in Figure 17-75 and described in Table 17-62.

**Figure 17-75. McBSP_THRSH2_REG**

| 31 | 12 | 11 | 0 |
|---|---|---|---|
| Reserved | | XTHRESHOLD | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 17-62. McBSP_THRSH2_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | Reserved | 0 | Reserved |
| 11-0 | XTHRESHOLD | | Transmit buffer threshold value.<br>The DMA request (if enabled) of interrupt assertion (if enabled) will be triggered if the number of free locations inside transmit buffer are above or equal to the XTHRESHOLD value + 1. Also, this value (XTHRESHOLD value + 1) indicates the number of words transferred during a transmit data DMA request, if transmit DMA is enabled. |

### 17.3.44 McBSP Receive Buffer Threshold Register (DMA or IRQ trigger) (THRSH1_REG)

The McBSP_THRSH1_REG register is shown in Figure 17-76 and described in Table 17-63.

**Figure 17-76. McBSP_THRSH1_REG**

| 31 | 12 | 11 | 0 |
|---|---|---|---|
| Reserved | | RTHRESHOLD | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 17-63. McBSP_THRSH1_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | Reserved | 0 | Reserved. |
| 11-0 | RTHRESHOLD | | Receive buffer threshold value.<br>The DMA request (if enabled) of interrupt assertion (if enabled) will be triggered if the number of occupied locations inside receive buffer are above or equal to the RTHRESHOLD value + 1. Also, this value (RTHRESHOLD value + 1) indicates the number of words transferred during a receive data DMA request, if receive DMA is enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.45 McBSP Interrupt Status Register (OCP compliant IRQ line) (IRQSTATATUS)

The McBSP_IRQSTATUS_REG register is shown in Figure 17-77 and described in Table 17-64.

**Figure 17-77. McBSP_IRQSTATUS_REG**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XEMPTYEOF | Reserved | XOVFLSTAT | XUNDFLSTAT | XRDY | XEOF | XFSX |
| R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| XSYNCERR | Reserved | ROVFLSTAT | RUNDFLSTAT | RRDY | REOF | RFSR | RSYNCERR |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-64. McBSP_IRQSTATUS_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved |
| 14 | XEMPTYEOF | | Transmit Buffer Empty at end of frame (XEMPTYEOF is set to 1 when a complete frame was transmitted and the transmit buffer is empty). |
| | | 0 | Transmit buffer NOT Empty at end of frame. |
| | | 1 | Transmit buffer Empty at end of frame; Writing 1 to this bit clears the bit. |
| 13 | Reserved | 0 | Reserved. |
| 12 | XOVFLSTAT | | Transmit Buffer Overflow (XOVFLSTAT bit is set to 1 when transmit buffer overflow; the data which is written while overflow condition is discarded). |
| | | 0 | Transmit buffer NOT overflow. |
| | | 1 | Transmit buffer overflow; Writing 1 to this bit clears the bit. |
| 11 | XUNDFLSTAT | | Transmit Buffer Underflow (XUNDFLSTAT bit is set to 1 when the transmit data buffer is empty new data is required to be transmitted). |
| | | 0 | The transmit data buffer is NOT empty new data is required to be transmitted. |
| | | 1 | The transmit data buffer is empty new data is required to be transmitted. Writing 1 to this bit clears the bit. |
| 10 | XRDY | | Transmit Buffer Threshold Reached (XRDY bit is set to 1 when the transmit buffer free locations are equal or above the THRSH2_REG value). |
| | | 0 | Transmit buffer occupied locations are below the THRSH2_REG value). |
| | | 1 | Transmit buffer occupied locations are equal or above the THRSH2_REG value). Writing 1 to this bit clears the bit. |
| 9 | XEOF | | Transmit End Of Frame (XEOF is set to 1when a complete frame was transmitted). |
| | | 0 | Complete frame was NOT transmitted. |
| | | 1 | Complete frame was transmitted; Writing 1 to this bit clears the bit. |
| 8 | XFSX | | Transmit Frame Synchronization (XFSX bit is set to 1 when a new transmit frame synchronization is asserted). |
| | | 0 | New transmit frame synchronization is NOT asserted. |
| | | 1 | New transmit frame synchronization is asserted; Writing 1 to this bit clears the bit. |
| 7 | XSYNCERR | | Transmit Frame Synchronization Error (XSYNCERR is set to 1 when a transmit frame synchronization error is detected). |
| | | 0 | Transmit frame synchronization error is NOT detected. |
| | | 1 | Transmit frame synchronization error is detected. Writing 1 to this bit clears the bit. |
| 6 | Reserved | 0 | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-64. McBSP_IRQSTATUS_REG Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 5 | ROVFLSTAT | | Receive Buffer Overflow (ROVFLSTAT bit is set to 1 when receive buffer overflow; the data which is written while overflow condition is discarded). |
| | | 0 | Receive buffer NOT overflow. |
| | | 1 | Receive buffer overflow; Writing 1 to this bit clears the bit. |
| 4 | RUNDFLSTAT | | Receive Buffer Underflow (RUNDFLSTAT bit is set to 1 when read operation is performed to the receive data register while receive buffer is empty; data read while underflow condition is undefined). |
| | | 0 | Read operation is performed to the receive data register while receive buffer is NOT empty. |
| | | 1 | Read operation is performed to the receive data register while receive buffer is empty; Writing 1 to this bit clears the bit. |
| 3 | RRDY | | Receive Buffer Threshold Reached (RRDY bit is set to 1 when the receive buffer occupied locations are equal or above the THRSH1_REG value). |
| | | 0 | Receive buffer occupied locations are below the THRSH1_REG value). |
| | | 1 | Receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit clears the bit. |
| 2 | REOF | | Receive End Of Frame (REOF is set to 1 when a complete frame was received). Writing 1 to this bit clears the bit. |
| | | 0 | Complete frame was NOT received. |
| | | 1 | Complete frame was received; Writing 1 to this bit clears the bit. |
| 1 | RFSR | | Receive Frame Synchronization (RFSR bit is set to 1 when a new receive frame synchronization is asserted). |
| | | 0 | New receive frame synchronization is NOT asserted. |
| | | 1 | New receive frame synchronization is asserted; Writing 1 to this bit clears the bit. |
| 0 | RSYNCERR | | Receive Frame Synchronization Error (RSYNCERR is set RW 0x0 to 1 when a receive frame synchronization error is detected). |
| | | 0 | Receive frame synchronization error is NOT detected. |
| | | 1 | Receive frame synchronization error is detected. Writing 1 to this bit clears the bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.46 McBSP Interrupt Enable Register (OCP compliant IRQ line) (IRQENABLE)

The McBSP_IRQENABLE_REG register is shown in Figure 17-78 and described in Table 17-65.

**Figure 17-78. McBSP_IRQENABLE_REG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | XEMPTYEOFEN | Reserved | XOVFLEN | XUNDFLEN | XRDYEN | XEOFEN | XFSXEN |
| R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| XSYNCERREN | Reserved | ROVFLEN | RUNDFLEN | RRDYEN | REOFEN | RFSREN | RSYNCERREN |
| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-65. McBSP_IRQENABLE_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. |
| 14 | XEMPTYEOFEN | | Transmit buffer empty at end of frame enable bit. |
| | | 0 | Transmit buffer Empty at end of frame NOT enabled. |
| | | 1 | Transmit buffer Empty at end of frame enabled. |
| 13 | Reserved | 0 | Reserved. |
| 12 | XOVFLEN | | Transmit Buffer Overflow enable bit. |
| | | 0 | Transmit Buffer Overflow NOT enabled. |
| | | 1 | Transmit Buffer Overflow enabled. |
| 11 | XUNDFLEN | | Transmit Buffer Underflow enable bit. |
| | | 0 | Transmit Buffer Underflow NOT enabled. |
| | | 1 | Transmit Buffer Underflow enabled. |
| 10 | XRDYEN | | Transmit Buffer Threshold Reached enable bit. |
| | | 0 | Transmit Buffer Threshold Reached NOT enabled. |
| | | 1 | Transmit Buffer Threshold Reached enabled. |
| 9 | XEOFEN | | Transmit End Of Frame enable bit. |
| | | 0 | Transmit End Of Frame NOT enabled. |
| | | 1 | Transmit End Of Frame enabled. |
| 8 | XFSXEN | | Transmit Frame Synchronization enable bit. |
| | | 0 | Transmit Frame Synchronization NOT enabled. |
| | | 1 | Transmit Frame Synchronization enabled. |
| 7 | XSYNCERREN | | Transmit Frame Synchronization Error enable bit. |
| | | 0 | Transmit Frame Synchronization Error NOT enabled. |
| | | 1 | Transmit Frame Synchronization Error enabled. |
| 6 | Reserved | 0 | Reserved. |
| 5 | ROVFLEN | | Receive Buffer Overflow enable bit. |
| | | 0 | Receive Buffer Overflow NOT enabled. |
| | | 1 | Receive Buffer Overflow enabled. |
| 4 | RUNDFLEN | | Receive Buffer Underflow enable bit. |
| | | 0 | Receive Buffer Underflow NOT enabled. |
| | | 1 | Receive Buffer Underflow enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-65. McBSP_IRQENABLE_REG Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 3 | RRDYEN | | Receive Buffer Threshold enable bit. |
| | | 0 | Receive Buffer Threshold NOT enabled. |
| | | 1 | Receive Buffer Threshold enabled. |
| 2 | REOFEN | | Receive End Of Frame enable bit. |
| | | 0 | Receive End Of Frame NOT enabled. |
| | | 1 | Receive End Of Frame enabled. |
| 1 | RFSREN | | Receive Frame Synchronization enable bit. |
| | | 0 | Receive Frame Synchronization NOT enabled. |
| | | 1 | Receive Frame Synchronization enabled. |
| 0 | RSYNCERREN | | Receive Frame Synchronization Error enable bit. |
| | | 0 | Receive Frame Synchronization Error NOT enabled. |
| | | 1 | Receive Frame Synchronization Error enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.47 McBSP Wakeup Enable Register (WAKEUPEN)

The McBSP_WAKEUPEN_REG register is shown in Figure 17-79 and described in Table 17-66.

**Figure 17-79. McBSP_WAKEUPEN_REG**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 13 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| Reserved | XEMPTYEOFEN | Reserved | | XRDYEN | XEOFEN | XFSXEN |
| R-0 | R/W-0 | R-0 | | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| XSYNCERREN | Reserved | | RRDYEN | REOFEN | RFSREN | RSYNCERREN |
| R/W-0 | R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-66. McBSP_WAKEUPEN_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. |
| 14 | XEMPTYEOFEN | | Transmit buffer empty at end of frame WK enable bit. |
| | | 0 | Transmit buffer Empty at end of frame WK enable is NOT active. |
| | | 1 | Transmit buffer Empty at end of frame WK enable is active. |
| 13-11 | Reserved | 0 | Reserved. |
| 10 | XRDYEN | | Transmit Buffer Threshold Reached WK enable bit. |
| | | 0 | Transmit Buffer Threshold WK enable is NOT active. |
| | | 1 | Transmit Buffer Threshold WK enable is active. |
| 9 | XEOFEN | | Transmit End Of Frame WK enable bit. |
| | | 0 | Transmit End Of Frame WK enable is NOT active. |
| | | 1 | Transmit End Of Frame WK enable is active. |
| 8 | XFSXEN | | Transmit Frame Synchronization WK enable bit. |
| | | 0 | Transmit Frame Synchronization WK enable is NOT active. |
| | | 1 | Transmit Frame Synchronization WK enable is active. |
| 7 | XSYNCERREN | | Transmit Frame Synchronization Error WK enable bit. |
| | | 0 | Transmit Frame Synchronization Error WK enable is NOT active. |
| | | 1 | Transmit Frame Synchronization Error WK enable is active. |
| 6-4 | Reserved | 0 | Reserved. |
| 3 | RRDYEN | | Receive Buffer Threshold wake-up enable bit. |
| | | 0 | Receive Buffer Threshold WK enable is NOT active. |
| | | 1 | Receive Buffer Threshold WK enable is active. |
| 2 | REOFEN | | Receive End Of Frame WK enable bit. |
| | | 0 | Receive End Of Frame WK enable is NOT active. |
| | | 1 | Receive End Of Frame WK enable is active. |
| 1 | RFSREN | | Receive Frame Synchronization WK enable bit. |
| | | 0 | Receive Frame Synchronization WK enable is NOT active. |
| | | 1 | Receive Frame Synchronization WK enable is active. |
| 0 | RSYNCERREN | | Receive Frame Synchronization Error WK enable bit. |
| | | 0 | Receive Frame Synchronization Error WK enable is NOT active. |
| | | 1 | Receive Frame Synchronization Error WK enable is active. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.48 McBSP Transmit Configuration Control Register (XCCR_REG)

The McBSP_XCCR_REG register is shown in Figure 17-80 and described in Table 17-67.

**Figure 17-80. McBSP_XCCR_REG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EXTCLKGATE | PPCONNECT | DXENDLY | | XFULL_CYCLE | Reserved | | |
| R/W-0 | R/W-0 | R/W-1 | | R/W-0 | R-0 | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | DLB | Reserved | XDMAEN | Reserved | | XDISABLE |
| R-0 | | R/W-0 | R-0 | R/W-1 | R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 17-67. McBSP_XCCR_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15 | EXTCLKGATE | | External clock gating enable (CLKX and FSX master only). When this bit is set and the transmit clock and FSX are set as output, the CLKX is enabled when FSX is active plus 3 clock cycles after (clock is provided for FWID + 4 clock cycles, assuming that the FSX width, active, is FWID + 1 clock cycles); outside this window the external transmit clock is gated. The receive uses the same gated transmit clock and transmit frame synchronization signals regardless of the CLKRM/FSRM settings. When using this mode the frame synchronization signal must be active during reception of the entire frame (FWID must be programmed accordingly) to ensure the proper receive process, which requires at least 3 cycles after the frame complete to transfer the data into the receive buffer. |
| | | 0 | External clock gating disabled. |
| | | 1 | External clock gating enable. |
| 14 | PPCONNECT | | Pair to pair connection. When set the DXENO pin is always set to 0, regardless of the frame boundary, setting the tree state buffer as output. |
| | | 0 | non Pair-to-pair connection. The DX pin will go to high-impedance state when there is no frame to transmit. |
| | | 1 | Pair-to-pair connection. When set, the DXENO pin is always set to 0, regardless of the frame boundary, setting the tree state buffer as output. This means the DX pin will be driven outside valid frame window. In that case, data sent by McBSP module during inactive channel are not guaranteed. |
| 13-12 | DXENDLY | | When McBSPi.McBSP_SPCR1_REG[7] DXENA bit is set to 1, this field selects the added delay as follow: |
| | | 0 | 8 ns |
| | | 1h | 14 ns (default) |
| | | 2h | 20 ns |
| | | 3h | 28 ns |
| 11 | XFULL_CYCLE | | Transmit full cycle mode select: |
| | | 0 | McBSP module operates in transmit half-cycle mode (transmit frame synchronization is sampled by the opposite edge of the clock used to drive transmit data). |
| | | 1 | McBSP module operates in transmit full-cycle mode (transmit frame synchronization is sampled by the same edge of the clock used to drive transmit data). |
| 10-6 | Reserved | 0 | Reserved. |
| 5 | DLB | | Digital Loopback. When this bit is set the transmit FSX, CLKX and DX are connected to FSR, CLKR, DR. The outputs of the McBSP (CLKR/CLKX, FSR/FSX, DX are not enabled) |
| | | 0 | No DLB. |
| | | 1 | DLB. |
| 4 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 17-67. McBSP_XCCR_REG Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 3 | XDMAEN | | Transmit DMA Enable bit. When cleared to 0 this bit will gate the external transmit DMA request, without resetting the DMA state machine. It is recommended to change this bit value only during transmit reset. |
| | | 0 | Will gate the external transmit DMA request. |
| | | 1 | Will NOT gate the external transmit DMA request. |
| 2-1 | Reserved | 0 | Reserved |
| 0 | XDISABLE | | Transmit Disable bit. When this bit is set the transmit process will stop at the next frame boundary. |
| | | 0 | The transmit process will NOT stop at the next frame boundary. |
| | | 1 | The transmit process will stop at the next frame boundary. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.49 McBSP Receive Configuration Control Register (RCCR_REG)

The McBSP_RCCR_REG register is shown in Figure 17-81 and described in Table 17-68.

**Figure 17-81. McBSP_RCCR_REG**

| 31 | 12 | 11 | 10 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | RFULL_CYCLE | Reserved | | RDMAEN | Reserved | | RDISABLE |
| R-0 | | R/W-1 | R-0 | | R/W-1 | R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 17-68. McBSP_RCCR_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | Reserved | 0 | Reserved |
| 11 | RFULL_CYCLE | | Receive full cycle mode select: |
| | | 0 | McBSP module operates in receive half-cycle mode (receive frame synchronization is sampled by the opposite edge of the clock used to sample receive data). |
| | | 1 | McBSP module operates in receive full-cycle mode (receive frame synchronization is sampled by the same edge of the clock used to sample receive data). |
| 10-4 | Reserved | 0 | Reserved |
| 3 | RDMAEN | | Receive DMA Enable bit. When cleared to 0 this bit will gate the external transmit DMA request, without resetting the DMA state machine. It is recommended to change this bit value only during receive reset. |
| | | 0 | When cleared to 0 this bit will gate the external transmit DMA request. |
| | | 1 | When set to 1 this bit will NOT gate the external transmit DMA request. |
| 2-1 | Reserved | 0 | Reserved |
| 0 | RDISABLE | | Receive Disable bit. When this bit is set the receive process will stop at the next frame boundary. |
| | | 0 | The receive process will NOT stop at the next frame boundary. |
| | | 1 | When this bit is set the receive process will stop at the next frame boundary. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.50 McBSP Transmit Buffer Status Registers (XBUFFSTAT_REG)

The McBSP_XBUFFSTAT_REG registers are shown in Figure 17-82 and described in Table 17-69.

**Figure 17-82. McBSP_XBUFFSTAT_REG**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | XBUFFSTAT | |
| R-0 | | R-80h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 17-69. McBSP_XBUFFSTAT_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | XBUFFSTAT | 80h | Transmit buffer status (indicates the number of free locations inside transmit buffer).<br>The XBUFFSTAT value reflects the buffer status on the OCP clock domain and it can be smaller than the real number of the free locations which are seen by the transmit state machine. |

### 17.3.51 McBSP Receive Buffer Status Registers (RBUFFSTAT_REG)

The McBSP_RBUFFSTAT_REG registers are shown in Figure 17-83 and described in Table 17-70.

**Figure 17-83. McBSP_RBUFFSTAT_REG**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RBUFFSTAT | |
| R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 17-70. McBSP_RBUFFSTAT_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7-0 | RBUFFSTAT | 0 | Receive Buffer Status (indicates the number of occupied locations inside receive buffer).<br>The RBUFFSTAT value reflects the buffer status on the OCP clock domain and it can be smaller than the real number of the occupied locations which are seen by the receive state machine. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 17.3.52 McBSP_STATUS_REG

The McBSP_STATUS_REG register is shown in Figure 17-84 and described in Table 17-71.

**Figure 17-84. McBSP_STATUS_REG**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | CLKMUXSTATUS |
| R-0 | | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 17-71. McBSP_STATUS_REG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | CLKMUXSTATUS | | Indicates that the McBSP AUDIOBUFFER clock muxing is done after exiting SmartIdle mode. When this bit is set one the response to a different register access is delayed until the muxing process is done. In order to avoid such a situation pooling can be performed to status register in order to evaluate when McBSP is ready. Note that this information is relevant only for the McBSP having AUDIOBUFFER. |


Copyright © 2011–2016, Texas Instruments Incorporated

# Multimedia Card (MMC)/Secure Digital (SD)/ Secure Digital I/O (SDIO) Card Interface

This chapter describes the multimedia card/secure digital/secure digital I/O (MMC/SD/SDIO) card interface.

**Topic** **Page**

**18.1 Introduction** ..... **3164**
**18.2 Architecture** ..... **3166**
**18.3 Low-Level Programming Models** ..... **3199**
**18.4 MMC/SD/SDIO Registers** ..... **3204**


Copyright © 2011–2016, Texas Instruments Incorporated

## 18.1 Introduction

### 18.1.1 Overview

This device contains a multimedia card high-speed/secure data/secure digital I/O (MMC/SD/SDIO) host controller that provides an interface between a local host (LH), and either MMC, SD memory cards, or SDIO cards and handles MMC/SD/SDIO transactions with minimal LH intervention.

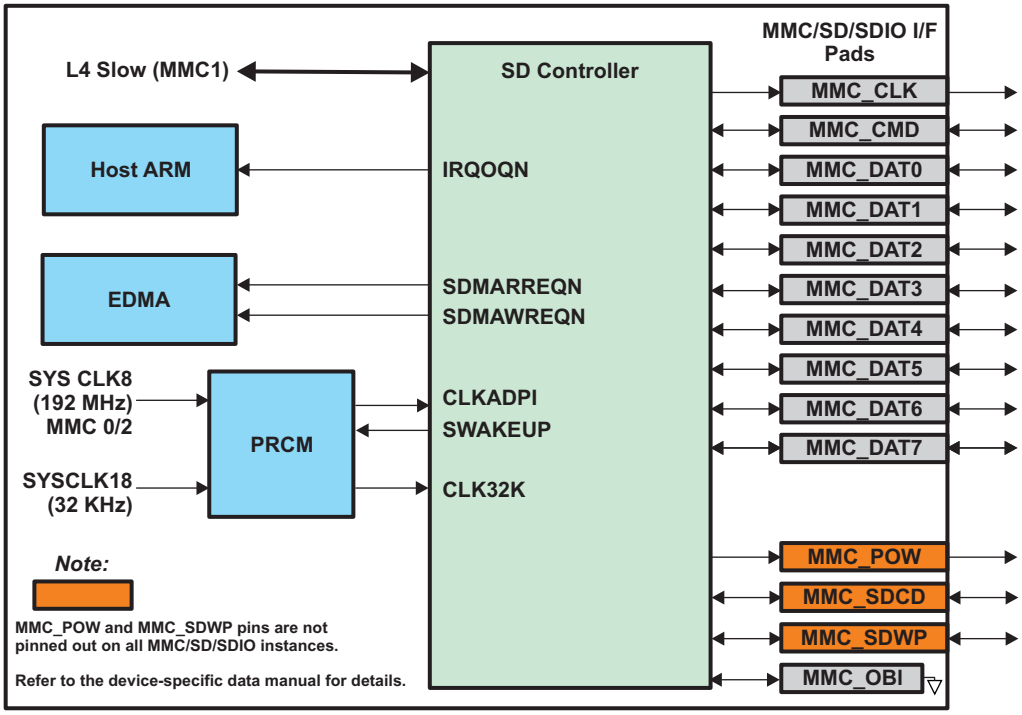
The application interface manages transaction semantics. The MMC/SD/SDIO host controller deals with MMC/SD/SDIO protocol at transmission level, data packing, adding cyclic redundancy checks (CRC), start/end bit, and checking for syntactical correctness.

The application interface can send every MMC/SD/SDIO command and either poll for the status of the adapter or wait for an interrupt request, which is sent back in case of exceptions or to warn of end of operation.

The application interface can read card responses or flag registers. It can also mask individual interrupt sources. All these operations can be performed by reading and writing control registers. The MMC/SD/SDIO host controller also supports two DMA channels.

There are typically three MMC/SD/SDIO host controllers inside the device. Figure 18-1 gives an overview of one MMC/SD/SDIO controller instance. As shown in Figure 18-1, not all pins are available on each instance of the MMC/SD/SDIO peripheral. Specifically, the MMC_SDCD, MMC_SDWP, and MMC_POW may not be pinned out on all instances. In most devices MMC/SD/SDIO1 provides the most full featured connectivity.

**Figure 18-1. MMC/SD/SDIO Overview**





Copyright © 2011–2016, Texas Instruments Incorporated

### 18.1.2 Features

The main features of the MMC/SD/SDIO host controller are:

- Built-in 1024-byte buffer for read or write
- Full compliance with MMC command/response sets as defined in the *Multimedia Card System Specification*, v4.3.
- Full compliance with SD command/response sets as defined in the *SD Physical Layer Specification*, v2.00.
- Full compliance with SDIO command/response sets and interrupt/read-wait mode as defined in the *SDIO Card Specification, Part E1*, v2.00
- Full compliance with SD Host Controller Standard Specification sets as defined in the *SD Card Specification, Part A2, SD Host Controller Standard Specification*, v2.00
- Full compliance with CE-ATA command/response sets as defined in the *CE-ATA Standard Specification*
- Full compliance with ATA for MMCA specification
- Support command completion signal (CCS) and command completion signal disable (CCSD) management as specified in the *CE-ATA Standard Specification*
- Flexible architecture allowing support for new command structure
- Support:
  - 1-bit or 4-bit transfer mode specifications for SD and SDIO cards
  - 1-bit, 4-bit, or 8-bit transfer mode specifications for MMC cards
  - 1.8v or 3.3v cards (Support varies per MMC/SD/SDIO instance.)
  - Built-in 1024-byte buffer for read or write
  - 32-bit-wide access bus to maximize bus throughput
  - Single interrupt line for multiple interrupt source events
  - Two slave DMA channels (1 for TX, 1 for RX)
  - Designed for low power
  - Programmable clock generation
  - SDIO Read Wait and Suspend/Resume functions
  - Stop at block gap
  - SDA 2.0 Part A2 programming model

**NOTE:** The device supports both 1.8v and 3.3v cards; however, one or more MMC/SD/SDIO instances may only support 3.3V operation. In addition, on some devices, multiple MMC/SD/SDIO instances may be required to both operate at the same voltage (for example, MMC/SD/SDIO0 and MMC/SD/SDIO1 may be required to both operate at 1.8V, or both at 3.3V). On other devices, the supply voltage can be independent (for example, MMC/SD/SDIO0 may be at 1.8V while MMC/SD/SDIO1 may be at 3.3V).

Refer to the device-specific data manual power group for details.

- Supported Data Rates
  - 192 MHz functional clock source input
  - Up to 384 Mbit/sec (48MByte/sec) in MMC mode 8-bit data transfer
  - Up to 192 Mbit/sec (24MByte/sec) in High-Speed SD mode 4-bit data transfer
  - Up to 24 Mbit/sec (3MByte/sec) in Default SD mode 1-bit data transfer


Copyright © 2011–2016, Texas Instruments Incorporated

## 18.2 Architecture

One MMC/SD/SDIO host controller can support one MMC memory card, one SD card, or one SDIO card. Other combinations (for example, two SD cards, one MMC card, and one SD card) are not supported through a single controller.

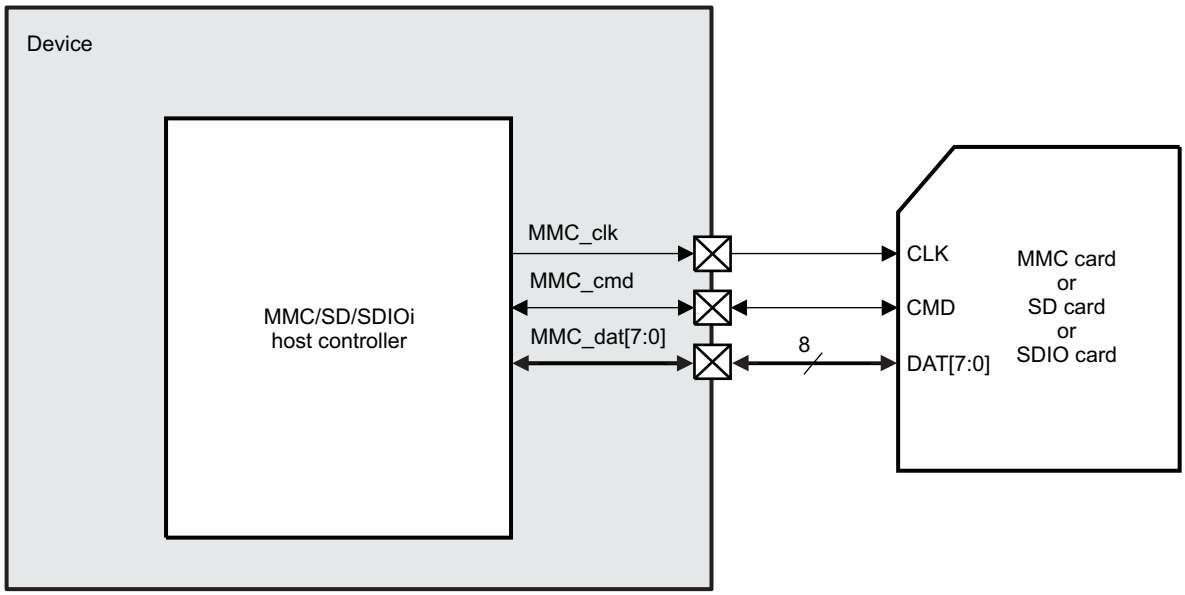
There are up to 3 MMC/SD/SDIO instances named: HSMMC0, HSMMC1, and HSMMC2.

### *18.2.1 MMC/SD/SDIO Functional Modes*

#### 18.2.1.1 MMC/SD/SDIO Connected to an MMC, an SD Card, or an SDIO Card

Figure 18-2 shows the MMC/SD/SDIO host controller connected to an MMC, an SD, or an SDIO card and its related external connections.

**Figure 18-2. MMC/SD Connectivity to an MMC/SD Card**




The following MMC/SD/SDIO controller pins are used

- **MMC_CMD** This pin is used for two-way communication between the connected card and the MMC/SD/SDIO controller. The MMC/SD/SDIO controller transmits commands to the card and the memory card drives responses to the commands on this pin.
- **MMC_DAT7-0** Depending on which type of card you are using, you may need to connect 1, 4, or 8 data lines. The number of DAT pins (the data bus width) is set by the Data Transfer Width (DTW) bit in the MMC control register (MMCHS_HCTL). For more information, see the registers section of this document.
- **MMC_CLK** This pin provides the clock to the memory card from the MMC/SD controller.

  **Note:** The MMC_CLK pin functions as an output but must be configured as an I/O to internally loopback the clock to time the inputs.

Table 18-1 provides a summary of these pins.



Copyright © 2011–2016, Texas Instruments Incorporated

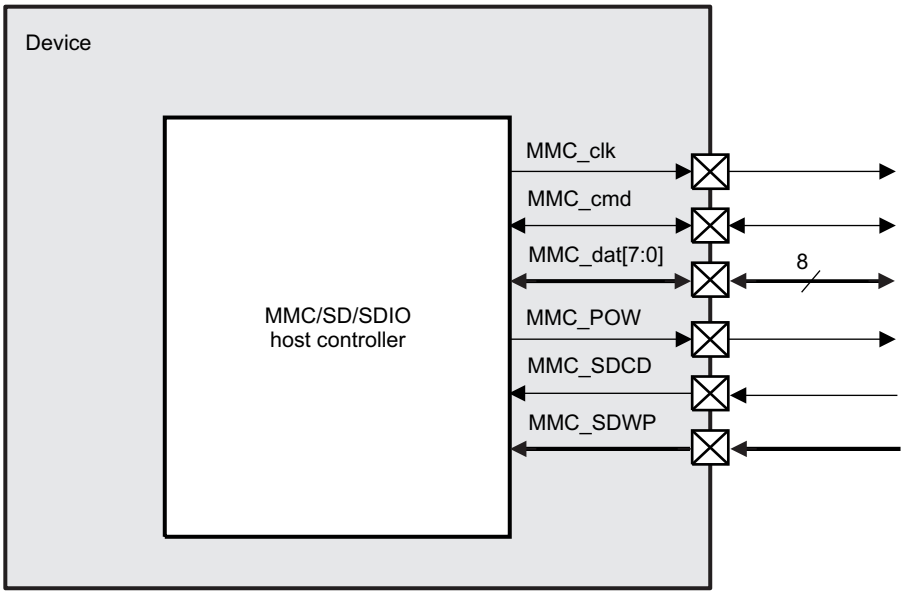
**Table 18-1. MMC/SD/SDIO Controller Pins and Descriptions**

| Pin | Type | 1-Bit Mode | 4-Bit Mode | 8-Bit Mode | Reset Value |
|---|---|---|---|---|---|
| MMC_CLK[1] | O | Clock Line | Clock Line | Clock Line | High impedance |
| MMC_CMD | I/O | Command Line | Command Line | Command Line | High impedance |
| MMC_DAT0 | I/O | Data Line 0 | Data Line 0 | Data Line 0 | 0 |
| MMC_DAT1 | I/O | (not used) | Data Line 1 | Data Line 1 | 0 |
| MMC_DAT2 | I/O | (not used) | Data Line 2 | Data Line 2 | 0 |
| MMC_DAT3 | I/O | (not used) | Data Line 3 | Data Line 3 | 0 |
| MMC_DAT4 | I/O | (not used) | (not used) | Data Line 4 | 0 |
| MMC_DAT5 | I/O | (not used) | (not used) | Data Line 5 | 0 |
| MMC_DAT6 | I/O | (not used) | (not used) | Data Line 6 | 0 |
| MMC_DAT7 | I/O | (not used) | (not used) | Data Line 7 | 0 |

(1) The MMC_CLK pin functions as an output but must be configured as an I/O to internally loopback the clock to time the inputs.

Figure 18-3 shows the MMC/SD/SDIO host controller connected to an MMC, SD, or SDIO card and its related external connections. These are described below.

**Figure 18-3. MMC/SD Connectivity to an MMC/SD Card**




The following MMC/SD/SDIO controller pins are not provided on all MMC/SD/SDIO instances. Refer to device-specific data manual for details.

- **MMC_POW** Used for MMC/SD card's cards on/off power supply control. When high, denotes power-on condition.
- **MMC_SDCD** This input pin serves as the MMC/SD/SDIO carrier detect. This signal is received from a mechanical switch on the slot.
- **MMC_SDWP** This input pin is used for the SD/SDIO card's write protect. This signal is received from a mechanical protect switch on the slot (system dependant). Applicable only for SD and SDIO cards that have a mechanical sliding tablet on the side of the card.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 18.2.1.2 Protocol and Data Format

The bus protocol between the MMC/SD/SDIO host controller and the card is message-based. Each message is represented by one of the following parts:

**Command:** A command starts an operation. The command is transferred serially from the \MMC/SD/SDIO host controller to the card on the mmc_cmd line.

**Response:** A response is an answer to a command. The response is sent from the card to the MMC/SD/SDIO host controller. It is transferred serially on the mmc_cmd line.

**Data:** Data are transferred from the MMC/SD/SDIO host controller to the card or from a card to the MMC/SD/SDIO host controller using the DATA lines.

**Busy:** The mmc_dat0 signal is maintained low by the card as far as it is programming the data received.

**CRC status:** CRC result is sent by the card through the mmc_dat0 line when executing a write transfer. In the case of transmission error, occurring on any of the active data lines, the card sends a negative CRC status on mmc_dat0. In the case of successful transmission, over all active data lines, the card sends a positive CRC status on mmc_dat0 and starts the data programming procedure.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 18.2.1.2.1 Protocol

There are two types of data transfer:

- Sequential operation
- Block-oriented operation

There are specific commands for each type of operation (sequential or block-oriented). See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, and the *SDIO Card Specification, Part E1* for details about commands and programming sequences supported by the MMC, SD, and SDIO cards.

**CAUTION**

Stream commands are supported only by MMC cards.

Figure 18-4 and Figure 18-5 show how sequential operations are defined. Sequential operation is only for 1-bit transfer and initiates a continuous data stream. The transfer terminates when a stop command follows on the mmchs_cmd line.

**Figure 18-4. Sequential Read Operation (MMC Cards Only)**

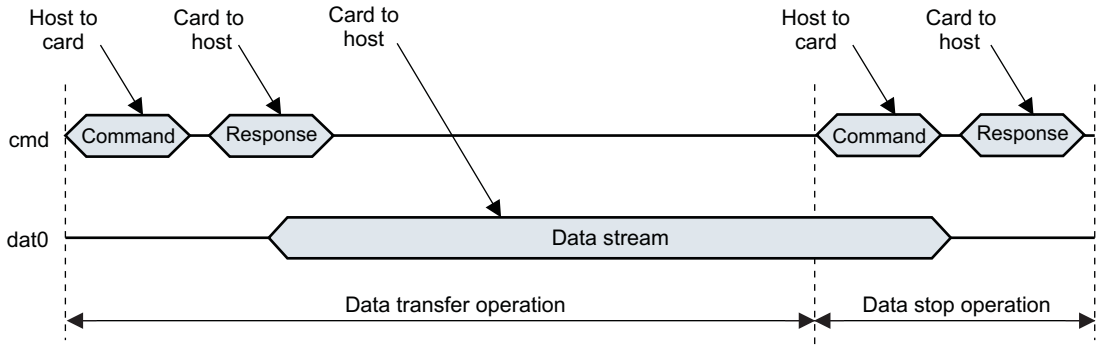
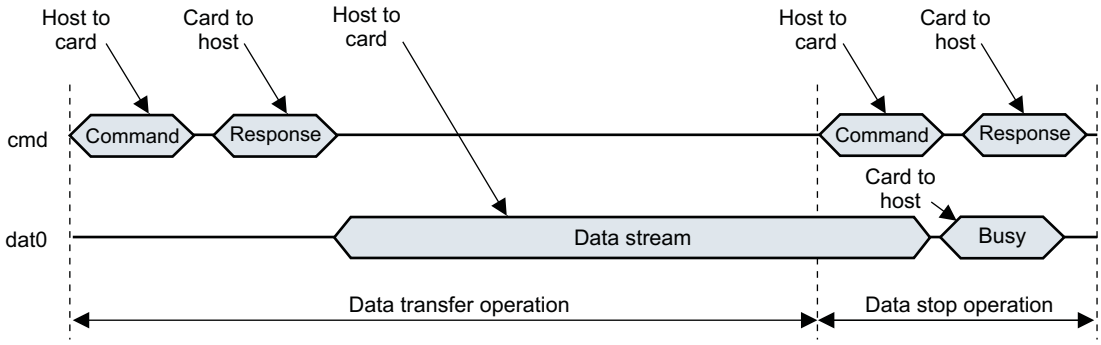



**Figure 18-5. Sequential Write Operation (MMC Cards Only)**





Copyright © 2011–2016, Texas Instruments Incorporated

Figure 18-6 and Figure 18-7 show how multiple block-oriented operations are defined. A multiple block-oriented operation sends a data block plus CRC bits. The transfer terminates when a stop command follows on the mmchs_cmd line. These operations are available for all kinds of cards.

**Figure 18-6. Multiple Block Read Operation (MMC Cards Only)**

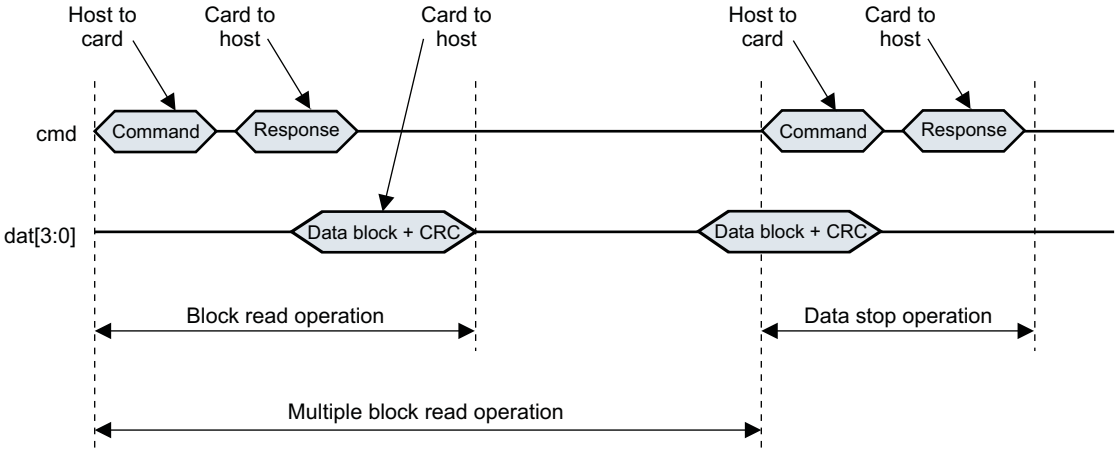
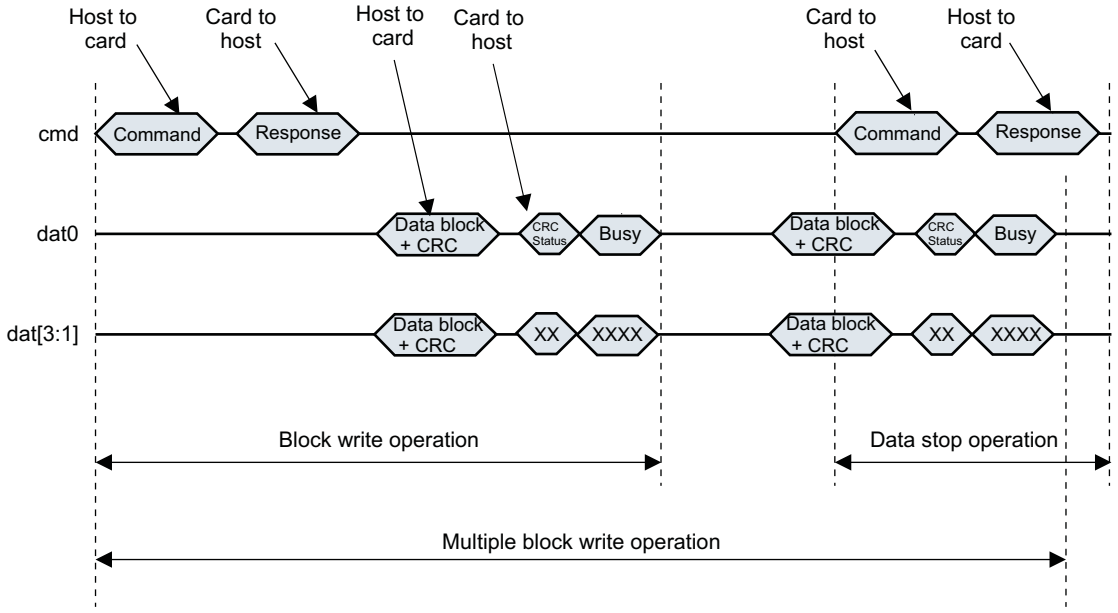



**Figure 18-7. Multiple Block Write Operation (MMC Cards Only)**




**NOTE:**

1. The card busy signal is not always generated by the card; the previous examples show a particular case.
2. It is the software's responsibility to do a software reset after a data timeout to ensure that mmc_clk is stopped. The software reset is done by setting the SRD bit in the MMCHS_SYSCTL register to 1.
3. For multiblock transfer, and especially for MMC cards, you can abort a transfer without using a stop command. Use a CMD23 before a data transfer to define the number of blocks that will be transferred, then the transfer stops automatically after the last block (provided the MMC card supports this feature).

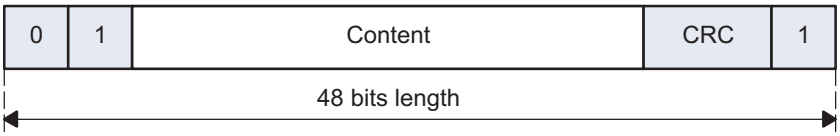

Copyright © 2011–2016, Texas Instruments Incorporated

#### 18.2.1.2.2 Data Format

**Coding Scheme for Command Token**

Command packets always start with 0 and end with 1. The second bit is a transmitter bit1 for a host command. The content is the command index (coded by 6 bits) and an argument (for example, an address), coded by 32 bits. The content is protected by 7-bit CRC checksum (see Figure 18-8).

**Figure 18-8. Command Token Format**




**Coding Scheme for Response Token**

Response packets always start with 0 and end with a 1. The second bit is a transmitter bit0 for a card response. The content is different for each type of response (R1, R2, R3, R4, R5, and R6) and the content is protected by 7-bit CRC checksum. Depending on the type of commands sent to the card, the MMCHS_CMD register must be configured differently to avoid false CRC or index errors to be flagged on command response (see Table 18-2). For more details about response types, see the *Multimedia Card System Specification*, the *SD Memory Card Specification*, or the *SDIO Card Specification*.

**Table 18-2. Response Type Summary**

| Response Type MMCHS_CMD[17:16] RSP_TYPE | Index Check Enable MMCHS_CMD[20] CICE | CRC Check Enable MMCHS_CMD[19] CCCE | Name of Response Type |
|---|---|---|---|
| 00 | 0 | 0 | No Response |
| 01 | 0 | 1 | R2 |
| 10 | 0 | 0 | R3 (R4 for SD cards) |
| 10 | 1 | 1 | R1, R6, R5 (R7 for SD cards) |
| 11 | 1 | 1 | R1b, R5b |

Figure 18-9 and Figure 18-10 depict the 48-bit and 136-bit response packets.

**Figure 18-9. 48-Bit Response Packet (R1, R3, R4, R5, R6)**

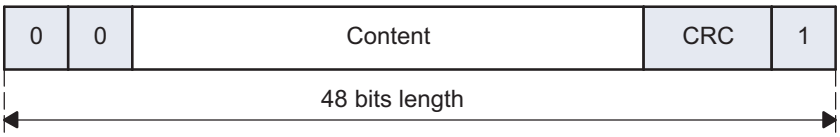
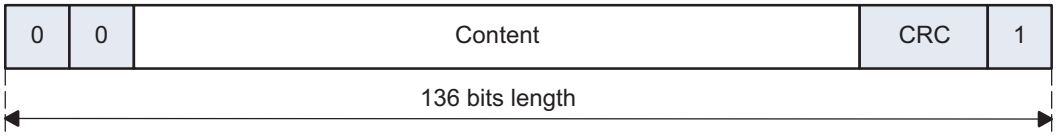



**Figure 18-10. 136-Bit Response Packet (R2)**





Copyright © 2011–2016, Texas Instruments Incorporated

### Coding Scheme for Data Token

Data tokens always start with 0 and end with 1 (see Figure 18-11, Figure 18-12, Figure 18-13, and Figure 18-14).

**Figure 18-11. Data Packet for Sequential Transfer (1-Bit)**

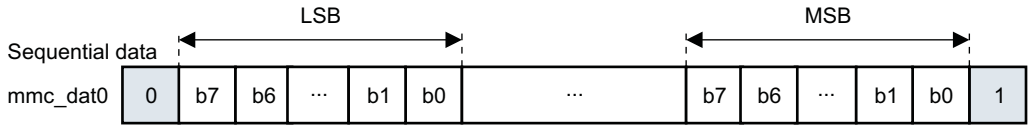



**Figure 18-12. Data Packet for Block Transfer (1-Bit)**

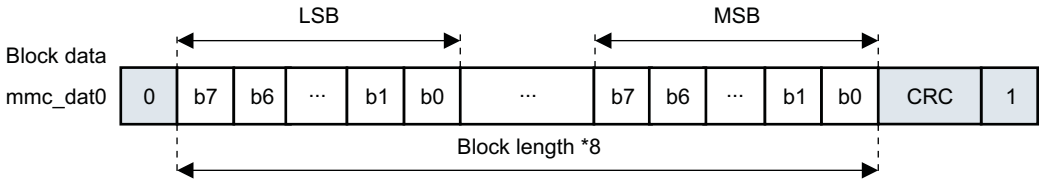
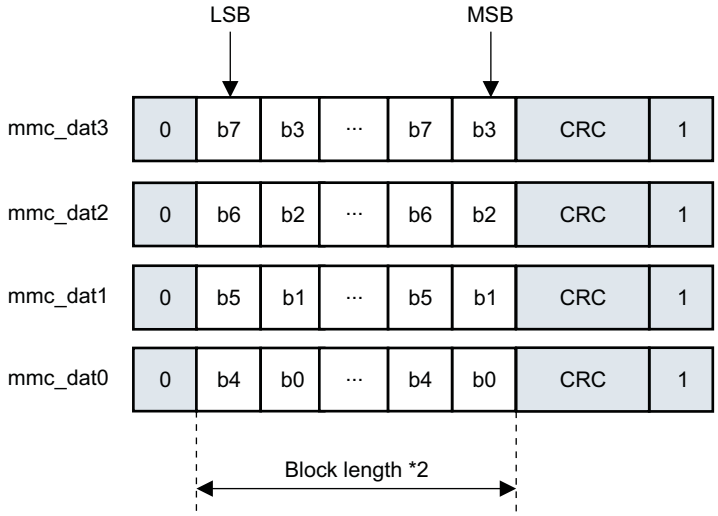


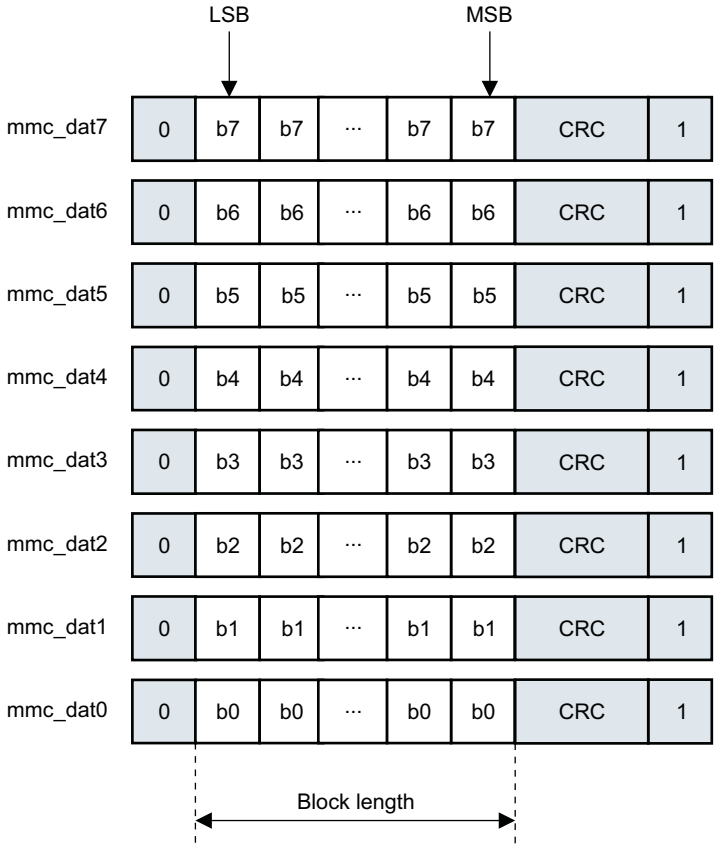

**Figure 18-13. Data Packet for Block Transfer (4-Bit)**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 18-14. Data Packet for Block Transfer (8-Bit)**




## *18.2.2 Resets*

### 18.2.2.1 Hardware Reset

The module is reinitialized by the hardware.

The MMCHS_SYSSTATUS[0] RESETDONE bit can be monitored by the software to check if the module is ready-to-use after a hardware reset.

This hardware reset signal has a global reset action on the module. All configuration registers and all state machines are reset in all clock domains.

This hardware reset signal has a global reset action on the module. All configuration registers and all state-machines are reset in all clock domains.

### 18.2.2.2 Software Reset

The module is reinitialized by software through the MMCHS_SYSCONFIG[1] SOFTRESET bit. This bit has the same action on the module logic as the hardware reset signal except for:

- Debounce logic
- MMCHS_PSTATE and MMCHS_CAPA registers (see corresponding register descriptions)

The SOFTRESET bit is active high. The bit is automatically reinitialized to 0 by the hardware. The MMCHS_SYSCTL[24] SRA bit has the same action as the SOFTRESET bit on the design.

The MMCHS_SYSSTATUS[0] RESETDONE bit can be monitored by the software to check if the module is ready-to-use after a software reset.


Copyright © 2011–2016, Texas Instruments Incorporated

Moreover, two partial software reset bits are provided:

- MMCHS_SYSCTL[26] SRD bit
- MMCHS_SYSCTL[25] SRC bit

These two reset bits are useful to reinitialize data or command processes respectively in case of line conflict. When set to 1, a reset process is automatically released when the reset completes:

- The MMCHS_SYSCTL[26] SRD bit resets all finite state-machines and status management that handle data transfers on both the interface and functional side.
- The MMCHS_SYSCTL[25] SRC bit resets all finite state-machines and status management that handle command transfers on both the interface and functional side.

**NOTE:** If **any** of the clock inputs are not present for the MMC/SD/SDIO peripheral, the software reset will not complete.

### 18.2.3 Clocks

The module receives 3 clocks: OCP CLK (named Interface clock), CLKADPI (named functional clock), and CLK32K. Concerning the activity of these external clocks, two cases can be considered:

- Module is being used: These 2 clocks must be activated in the same time when you want to make an OCP access on register due to the synchronization that is performed on some configuration registers (MMCHS_CMD) but also during the data transfer phase in order to synchronize data and status bits between the functional and the OCP part.
- Module is not being used: The 2 clocks can be turned off because the module is the master, which provides clock to the different type of card, and any transfer can be only initiated by the MMC module.

The CLK32K clock is used for debouncing.

### 18.2.4 Power Management

The MMC/SD/SDIO host controller can enter into different modes and save power:

- Normal mode
- Idle mode

The two modes are mutually exclusive (the module can be in normal mode or in idle mode). The MMC/SD/SDIO host controller is compliant with the PRCM module handshake protocol. When the MMC/SD/SDIO power domain is off, the only way to wake up the power domain and different MMC/SD/SDIO clocks is to monitor the mmc_dat1 input pin state via a different GPIO line for each MMC/SD/SDIO interface.

#### 18.2.4.1 Normal Mode

The autogating of interface clock (OCP clock) and functional clock (ADPI clock) occurs when the following conditions are met:

- The MMCHS_SYSCONFIG[0] AUTOIDLE bit is set to 1.
- There is no transaction on the MMC interface.

The autogating of interface and functional clocks stops when the following conditions are met:

- A register access occurs through the L3 (or L4) interconnect.
- A wake-up event occurs (an interrupt from a SDIO card).
- A transaction on the MMC/SD/SDIO interface starts.

Then the MMC/SD/SDIO host controller enters in low-power state even if MMCHS_SYSCONFIG[0] AUTOIDLE is cleared to 0. The functional clock is internally switched off and only interconnect read and write accesses are allowed.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 18.2.4.2 Idle Mode

The clocks provided to MMC/SD/SDIO are switched off upon a PRCM module request. They are switched back upon module request. The MMC/SD/SDIO host controller complies with the PRCM module handshaking protocol:

- Idle request from the system power manager
- Idle acknowledgment from the MMC/SD/SDIO host controller
- Wake-up request from the MMC/SD/SDIO host controller

The idle acknowledgment varies according to the MMCHS_SYSCONFIG[4:3] SIDLEMODE bit field:

- 0: Force-idle mode. The MMC/SD/SDIO host controller acknowledges the system power manager request unconditionally.
- 1h: No-idle mode. The MMC/SD/SDIO host controller ignores the system power manager request and behaves normally as if the request was not asserted.
- 2h: Smart-idle mode. The MMC/SD/SDIO host controller acknowledges the system power manager request according to its internal state.
- 3h: Smart-idle wake-up-capable mode. The MMC/SD/SDIO host controller acknowledges the system power manager request according to its internal state. However, the module may generate wake-up events when in idle state ( related to IRQ or DMA requests)

During the smart-idle mode period, the MMC/SD/SDIO host controller acknowledges that the OCP and Functional clocks may be switched off whatever the value set in the MMCHS_SYSCONFIG[9:8] CLOCKACTIVITY field.

#### 18.2.4.3 Transition from Normal Mode to Smart-Idle Mode

Smart-idle mode is enabled when the MMCHS_SYSCONFIG[4:3] SIDLEMODE bit field is set to 2h or 3h. The MMC/SD/SDIO host controller goes into idle mode when the PRCM issues an idle request, according to its internal activity. The MMC/SD/SDIO host controller acknowledges the idle request from the PRCM after ensuring the following:

- The current multi/single-block transfer is completed.
- Any interrupt or DMA request is asserted.
- There is no card interrupt on the mmchs_dat1 signal.

As long as the MMC/SD/SDIO controller does not acknowledge the idle request, if an event occurs, the MMC/SD/SDIO host controller can still generate an interrupt or a DMA request. In this case, the module ignores the idle request from the PRCM.

As soon as the MMC/SD/SDIO controller acknowledges the idle request from the PRCM:

- If Smart-Idle mode: the module does not assert any new interrupt or DMA request
- If Smart-Idle wake-up-capable mode: the module may generate wake-up events related to interrupt or DMA request.

#### 18.2.4.4 Transition from Smart-Idle Mode to Normal Mode

The MMC/SD/SDIO host controller detects the end of the idle period when the PRCM deasserts the idle request. For the wake-up event, there is a corresponding interrupt status in the MMCHS_STAT register. The MMC/SD/SDIO host controller operates the conversion between wake-up and interrupt (or DMA request) upon exit from smart-idle mode if the associated enable bit is set in the MMCHS_ISE register.

Interrupts and wake-up events have independent enable/disable controls, accessible through the MMCHS_HCTL and MMCHS_ISE registers. The overall consistency must be ensured by software.

The interrupt status register MMCHS_STAT is updated with the event that caused the wake-up in the CIRQ bit when the MMCHS_IE[8] CIRQ_ENABLE associated bit is enabled. Then, the wake-up event at the origin of the transition from smart-idle mode to normal mode is converted into its corresponding interrupt or DMA request. (The MMCHS_STAT register is updated and the status of the interrupt signal changes.)


Copyright © 2011–2016, Texas Instruments Incorporated

When the idle request from the PRCM is deasserted, the module switches back to normal mode. The module is fully operational.

#### 18.2.4.5 Force-Idle Mode

Force-idle mode is enabled when the MMCHS_SYSCONFIG[4:3] SIDLEMODE bit field is cleared to 0. Force-idle mode is an idle mode where the MMC/SD/SDIO host controller responds unconditionally to the idle request from the PRCM. Moreover, in this mode, the MMC/SD/SDIO host controller unconditionally deasserts interrupts and DMA request lines asserted.

The transition from normal mode to force-idle mode does not affect the bits of the MMCHS_STAT register. In force-idle mode, the interrupt and DMA request lines are deasserted. Interface Clock (OCP) and functional clock (CLKADPI) can be switched off.

**CAUTION**

In Force-ilde mode, an idle request from the PRCM during a command or a data transfer can lead to an unexpected and unpredictable result. When the module is idle, any access to the module generates an error as long as the OCP clock is alive.

The module exits the force-idle mode when the PRCM deassertes the idle request. Then the module switches back to normal mode. The module is fully operational. Interrupt and DMA request lines are optionally asserted one clock cycle later.

#### 18.2.4.6 Local Power Management

Table 18-3 describes power-management features available for the MMC/SD/SDIO modules.

**CAUTION**

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the CLOCKACTIVITY and MMC clock PRCM control bits.

**Table 18-3. Local Power Management Features**

| Feature | Registers | Description |
|---|---|---|
| Clock Auto Gating | MMCHS_SYSCONFIG AUTOIDLE bit | This bit allows a local power optimization inside module, by gating the OCP clock upon the interface activity or gating the CLKADPI clock upon the internal activity. |
| Slave Idle Modes | MMCHS_SYSCONFIG SIDLEMODE bit | Force-idle, No-idle, and Smart-idle modes |
| Clock Activity | MMCHS_SYSCONFIG CLOCKACTIVITY bit | See Table 18-4 for configuration details. |
| Master Standby Modes | MMCHS_SYSCONFIG STANDBYMODE bit | Force-idle, No-idle, and Smart-idle modes |
| Global Wake-Up Enable | MMCHS_SYSCONFIG ENAWAKEUP bit | This bit enables the wake-up feature at module level. |
| Wake-Up Sources Enable | MMCHS_HCTL register | This register holds one active high enable bit per event source able to generate wake-up signal. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-4. Clock Activity Settings**

| CLOCKACTIVITY Values | Clock State When Module is in IDLE State | | Features Available when Module is in IDLE State | Wake-Up Events |
|---|---|---|---|---|
| | OCP Clock | CLKADPI | | |
| 00 | OFF | OFF | None | Card Interrupt |
| 10 | OFF | ON | None | |
| 01 | ON | OFF | None | |
| 11 | ON | ON | All | |

### 18.2.5 Interrupt Requests

Several internal module events can generate an interrupt. Each interrupt has a status bit, an interrupt enable bit, and a signal status enable:

- The status of each type of interrupt is automatically updated in the MMCHS_STAT register; it indicates which service is required.
- The interrupt status enable bits of the MMCHS_IE register enable/disable the automatic update of the MMCHS_STAT register on an event-by-event basis.
- The interrupt signal enable bits of the MMCHS_ISE register enable/disable the transmission of an interrupt request on the interrupt line MMC_IRQ (from the MMC/SD/SDIOi host controller to the MPU subsystem interrupt controller) on an event-by-event basis.

If an interrupt status is disabled in the MMCHS_IE register, then the corresponding interrupt request is not transmitted, and the value of the corresponding interrupt signal enable in the MMCHS_ISE register is ignored.

When an interrupt event occurs, the corresponding status bit is automatically set to 1 (the MMC/SD/SDIO host controller updates the status bit) in the MMCHS_STAT register. If later a mask is applied on the interrupt in the MMCHS_ISE register, the interrupt request is deactivated.

When the interrupt source has not been serviced, if the interrupt status is cleared in the MMCHS_STAT register and the corresponding mask is removed from the MMCHS_ISE register, the interrupt status is not asserted again in the MMCHS_STAT register and the MMC/SD/SDIOi host controller does not transmit an interrupt request.

**CAUTION**

If the buffer write ready interrupt (BWR) or the buffer read ready only interrupt (BRR) are not serviced and are cleared in the MMCHS_STAT register, and the corresponding mask is removed, then the MMC/SD/SDIOi host controller will wait for the service of the interrupt without updating the status MMCHS_STAT register or transmitting an interrupt request.

Table 18-5 lists the event flags, and their mask, that can cause module interrupts.

**Table 18-5. Events**

| Event Flag | Event Mask | Map To | Description |
|---|---|---|---|
| MMCHS_STAT[29] BADA | MMCHS_IE[29] BADA_ENABLE | MMC_IRQ | Bad Access to Data space. This bit is set automatically to indicate a bad access to buffer when not allowed. This bit is set during a read access to the data register (MMCHS_DATA) while buffer reads are not allowed (MMCHS_PSTATE[11] BRE=0). This bit is set during a write access to the data register (MMCHS_DATA) while buffer writes are not allowed (MMCHS_STATE[10] BWE=0) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-5. Events (continued)**

| Event Flag | Event Mask | Map To | Description |
|---|---|---|---|
| MMCHS_STAT[28] CERR | MMCHS_IE[28] CERR_ENABLE | MMC_IRQ | Card Error. This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5 or R5b. Only bits referenced as type E(error) in status field in the response can set a card status error. An error bit in the response is flagged only if corresponding bit in card status response errros register (MMCHS_CSRE) is set. There is not card detection for auto CMD12 command. |
| MMCHS_STAT[25] ADMAE | MMCHS_IE[25] ADMAE_ENABLE | MMC_IRQ | AMDA error. This bit is set when the host controller detects errors during ADMA based data transfer. The stat of the ADMA at an error occurrence is saved in the ADMA Error Status Register. In addition, the host controller generates this interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. |
| MMCHS_STAT[24] ACE | MMCHS_IE[24] ACE_ENABLE | MMC_IRQ | Auto CMD12 error. This bit is set automatically when one of the bits in Auto CMD12 Error status register has changed from 0 to 1 |
| MMCHS_STAT[22] DEB | MMCHS_IE[22] DEB_ENABLE | MMC_IRQ | Data End Bit error. This bit is set automatically when detecting a 0 at the end bit position of read data on DAT line or at the end position of the CRC status in write mode. |
| MMCHS_STAT[21] DCRC | MMCHS_IE[21] DCRC_ENABLE | MMC_IRQ | Data CRC error. This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status difference of a position "010" token during a block write command. |
| MMCHS_STAT[20] DTO | MMCHS_IE[20] DTO_ENABLE | MMC_IRQ | Data Timeout error. This bit is set automatically according to the following conditions: A) busy timeout for R1b, R5b response. B) busy timeout after write CRC status. C) write CRC status timeout, or D) read data timeout. |
| MMCHS_STAT[19] CIE | MMCHS_IE[19] CIE_ENABLE | MMC_IRQ | Command Index Error. This bit is set automatically when response index differs from corresponding command index previously emitted. The check is enabled through MMCHS_CMD[20] CICE bit. |
| MMCHS_STAT[18] CEB | MMCHS_IE[18] CEB_ENABLE | MMC_IRQ | Command End Bit error. This bit is set automatically when detecting a 0 at the end bit position of a command response. |
| MMCHS_STAT[17] CCRC | MMCHS_IE[17] CCRC_ENABLE | MMC_IRQ | Command CRC error. This bit is set automatically when there is a CRC7 error in the command response. CRC check is enabled through the MMCHS_CMD[19] CCCE bit. |
| MMCHS_STAT[16] CTO | MMCHS_IE[16] CTO_ENABLE | MMC_IRQ | Command Timeout error. This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands that reply within 5 clock cycles, the timeout is still detected at 64 clock cycles. |
| MMCHS_STAT[15] ERRI | MMCHS_IE[15] ERRI_ENABLE | MMC_IRQ | Error Interrupt. If any of the bits in the Error Interrupt Status register (MMCHS_STAT[24:15]) are set, this bit is set to 1. |
| MMCHS_STAT[10] BSR | MMCHS_IE[10] BSR_ENABLE | MMC_IRQ | Boot Status Received interrupt. This bit is set automatically when MMCHS_CON[18] BOOT_CF0 is set to 1 or 2h and boot status is received on the dat0 line. This interrupt is only used for MMC cards. |
| MMCHS_STAT[8] CIRQ | MMCHS_IE[8] CIRQ_ENABLE | MMC_IRQ | Card Interrupt. This bit is only used for SD, SDIO, and CE-ATA cards. In 1-bit mode, interrupt source is asynchronous (can be a source of asynchronous wake-up). In 4-bit mode, interrupt source is sampled during the interrupt cycle. In CE-ATA mode, interrupt source is detected when the card drive CMD line to zero during one cycle after data transmission end. |
| MMCHS_STAT[5] BRR | MMCHS_IE[5] BRR_ENABLE | MMC_IRQ | Buffer Read ready. This bit is set automatically during a read operation to the card when one block specified by MMCHS_BLK[10:0] BLEN is completely written in the buffer. It indicates that the memory card has filled out the buffer and the local host needs to empty the buffer by reading it. |
| MMCHS_STAT[4] BWR | MMCHS_IE[4] BWR_ENABLE | MMC_IRQ | Buffer Write ready. This bit is automatically set during a write operation to the card when the host can write a complete block as specified by MMCHS_BLK[10:0] BLEN. It indicates that the memory card has emptied one block from the buffer and the local host is able to write one block of data into the buffer. |
| MMCHS_STAT[3] DMA | MMCHS_IE[3] DMA_ENABLE | MMC_IRQ | DMA interrupt. This status is set when an interrupt is required in the ADMA instruction and after the data transfer is complete. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-5. Events (continued)**

| Event Flag | Event Mask | Map To | Description |
|---|---|---|---|
| MMCHS_STAT[2] BGE | MMCHS_IE[2] BGE_ENABLE | MMC_IRQ | Block Gap event. When a stop at block gap is requested (MMCHS_HCTL[16] SBGR), this bit is automatically set when transaction is stopped at the block gap during a read or write operation. |
| MMCHS_STAT[1] TC | MMCHS_IE[1] TC_ENABLE | MMC_IRQ | Transfer completed. This bit is always set when a read/write transfer is completed or between two blocks when the transfer is stopped due to a stop at block gap requested (MMCHS_HCTL[16 SBGR). In read mode this bit is automatically set on completion of a read transfer (MMCHS_PSTATE[9] RTA). In write mode, this bit is automatically set on completion of the DAT line use (MMCHS_PSTATE[2] DLA). |
| MMCHS_STAT[0] CC | MMCHS_IE[0] CC_ENABLE | MMC_IRQ | Command complete. This bit is set when a 1-to-0 transition occurs in the register command inhibit (MMCHS_PSTATE[0] CMDI). If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command. A command timeout error (MMCHS_STAT[16] CTO) has higher priority than command complete (MMCHS_STAT[0] CC). If a response is expected but none is received, then a Command Timeout error is detected and signaled instead of the Command Complete interrupt. |

#### 18.2.5.1 Interrupt-Driven Operation

An interrupt enable bit must be set in the MMCHS_IE register to enable the module internal source of interrupt.

When an interrupt event occurs, the single interrupt line is asserted and the LH must:

- Read the MMCHS_STAT register to identify which event occurred.
- Write 1 into the corresponding bit of the MMCHS_STAT register to clear the interrupt status and release the interrupt line (if a read is done after this write, this would return 0).

**NOTE:** In the MMCHS_STAT register, Card Interrupt (CIRQ) and Error Interrupt (ERRI) bits cannot be cleared.

The MMCHS_STAT[8] CIRQ status bit must be masked by disabling the MMCHS_IE[8] CIRQ_ENABLE bit (cleared to 0), then the interrupt routine must clear SDIO interrupt source in SDIO card common control register (CCCR).

The MMCHS_STAT[15] ERRI bit is automatically cleared when all status bits in MMCHS_STAT[31:16] are cleared.

#### 18.2.5.2 Polling

When the interrupt capability of an event is disabled in the MMCHS_ISE register, the interrupt line is not asserted:

- Software can poll the status bit in the MMCHS_STAT register to detect when the corresponding event occurs.
- Writing 1 into the corresponding bit of the MMCHS_STAT register clears the interrupt status and does not affect the interrupt line state.

**NOTE:** See the note in Section 18.2.5.1 concerning CIRQ and ERRI bits clearing.


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.6 DMA Modes

The device supports DMA slave mode only. In this case, the controller is slave on DMA transaction managed by two separated requests (SDMAWREQN and SDMARREQN)

#### 18.2.6.1 DMA Slave Mode Operations

The MMC/SD/SDIO controller can be interfaced with a DMA controller. At system level, the advantage is to discharge the local host (LH) of the data transfers. The module does not support wide DMA access (above 1024 bytes) for SD cards as specified in the *SD Card Specification*and *SD Host Controller Standard Specification*.

The DMA request is issued if the following conditions are met:

- The MMCHS_CMD[0] DE bit is set to 1 to trigger the initial DMA request (the write must be done when running the data transfer command).
- A command was emitted on the SD_cmd line.
- There is enough space in the buffer of the MMC/SD/SDIO controller to write an entire block (BLEN writes).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 18.2.6.1.1 DMA Receive Mode

In a DMA block read operation (single or multiple), the request signal SDMARREQN is asserted to its active level when a complete block is written in the buffer. The block size transfer is specified in the MMCHS_BLK BLEN field.

The SDMARREQN signal is deasserted to its inactive level when the sDMA has read one single word from the buffer. Only one request is sent per block; the DMA controller can make a 1-shot read access or several DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to block size BLEN field.

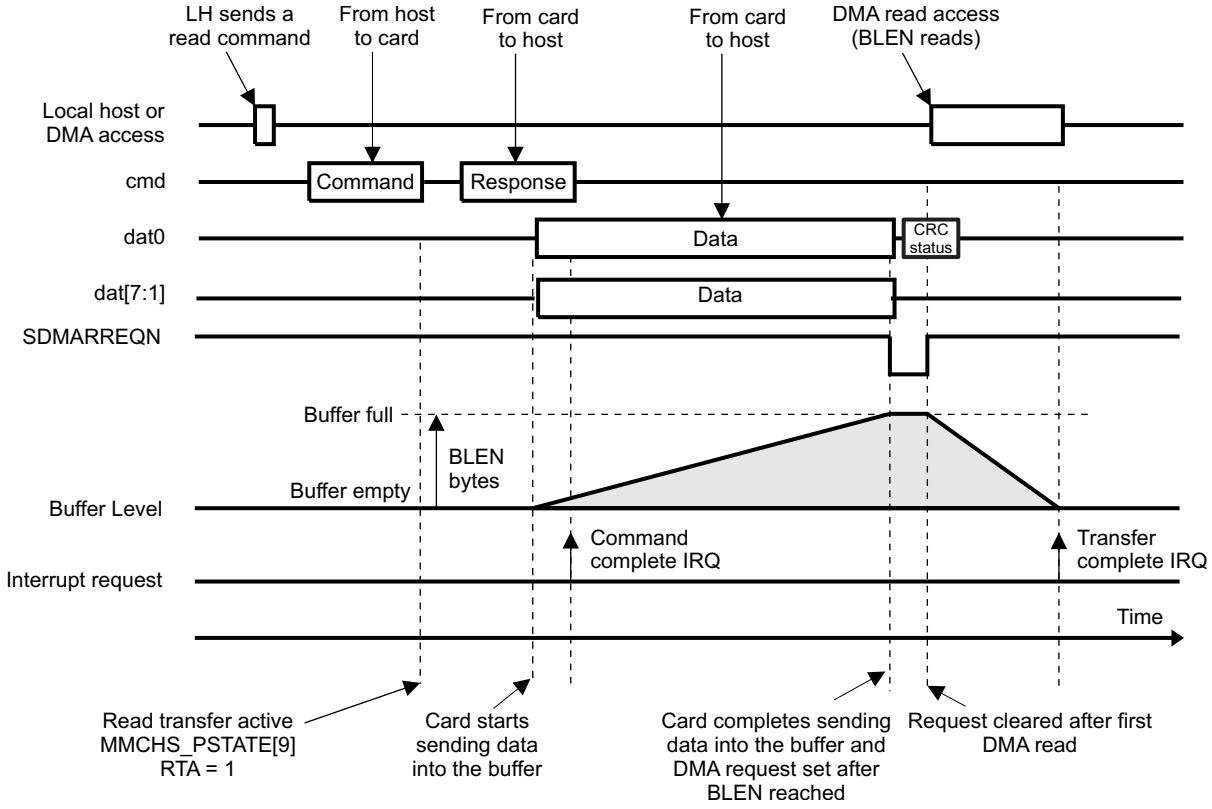
New DMA requests are internally masked if the sDMA has not read exactly BLEN bytes and a new complete block is not ready. As DMA accesses are in 32-bit, then the number of sDMA read is Integer(BLEN/4)+1.

The receive buffer never overflows. In multiple block transfers for block size above 512 bytes, when the buffer gets full, the MMC_CLK clock signal (provided to the card) is momentarily stopped until the sDMA or the MPU performs a read access, which reads a complete block in the buffer.

Figure 18-15 provides a summary:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

**Figure 18-15. DMA Receive Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 18.2.6.1.2 DMA Transmit Mode

In a DMA block write operation (single or multiple), the request signal SDMAWREQN is asserted to its active level when a complete block is to be written to the buffer. The block size transfer is specified in the MMCHS_BLK BLEN field.

The SDMAWREQN signal is deasserted to its inactive level when the sDMA has written one single word to the buffer.

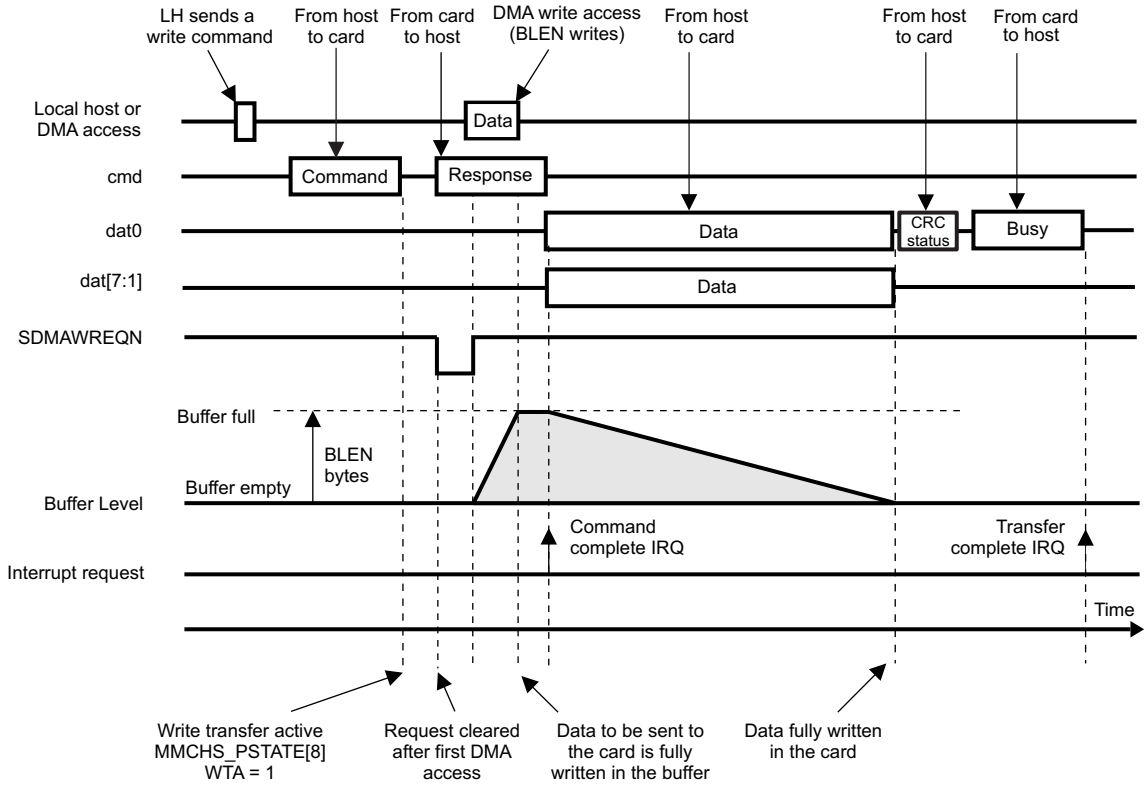
Only one request is sent per block; the DMA controller can make a 1-shot write access or multiple write DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to block size BLEN field.

New DMA requests are internally masked if the sDMA has not written exactly BLEN bytes (as DMA accesses are in 32-bit, then the number of sDMA read is Integer(BLEN/4)+1) and if there is not enough memory space to write a complete block in the buffer.

Figure 18-16 provides a summary:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

**Figure 18-16. DMA Transmit Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.7 Mode Selection

The MMC/SD/SDIO host controller can be use in two modes, MMC and SD/SDIO modes. It has been designed to be the most transparent with the type of card. The type of the card connected is differentiated by the software initialization procedure.

Software identifies the type of card connected during software initialization. For each given card type, there are corresponding commands. Some commands are not supported by all cards. See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, and the *SDIO Card Specification, Part E1* for more details.

The purpose of the module is to transfer commands and data, to whatever card is connected, respecting the protocol of the connected card. Writes and reads to the card must respect the appropriate protocol of that card.

### 18.2.8 Buffer Management

#### 18.2.8.1 Data Buffer

The MMC/SD/SDIO host controller uses a data buffer. This buffer transfers data from one data bus (Interconnect) to another data bus (SD, SDIO, or MMC card bus) and vice versa.

The buffer is the heart of the interface and ensures the transfer between the two interfaces (L4 and the card). To enhance performance, the data buffer is completed by a prefetch register and a post-write buffer that are not accessible by the host controller.

The read access time of the prefetch register is faster than the one of the data buffer. The prefetch register allows data to be read from the data buffer at an increased speed by preloading data into the prefetch register.

The entry point of the data buffer, the prefetch buffer, and the post-write buffer is the 32-bit register MMCHS_DATA. A write access to the MMCHS_DATA register followed by a read access from the MMCHS_DATA register corresponds to a write access to the post-write buffer followed by a read access to the prefetch buffer. As a consequence, it is normal that the data of the write access to the MMCHS_DATA register and the data of the read access to the MMCHS_DATA register are different.

The number of 32-bit accesses to the MMCHS_DATA register that are needed to read (or write) a data block with a size of MMCHS_BLK BLEN, and equals the rounded up result of BLEN divided by 4. The maximum block size supported by the host controller is hard-coded in the register MMCHS_CAPA[17:16] MBL field and cannot be changed.

A read access to the MMCHS_DATA register is allowed only when the buffer read enable status is set to 1 (MMCHS_PSTATE[11] BRE); otherwise, a bad access (MMCHS_STAT[29] BADA) is signaled.

A write access to the MMCHS_DATA register is allowed only when the buffer write enable status is set to 1 (MMCHS_PSTATE[10] BWE); otherwise, a bad access (MMCHS_STAT[29] BADA) is signaled and the data is not written.

The data buffer has two modes of operation to store and read of the first and second portions of the data buffer:

- When the size of the data block to transfer is less than or equal to MEM_SIZE/2 (in double buffering), two data transfers can occur from one data bus to the other data bus and vice versa at the same time. The MMC/SD/SDIO controller uses the two portions of the data buffer in a ping-pong manner so that storing and reading of the first and second portions of the data buffer are automatically interchanged from time to time so that data may be read from one portion (for instance, through a DMA read access on the interconnect bus) while data (for instance, from the card) is being stored into the other portion and vice versa. When BLEN is less than or equal to 200h (that is, less or equal to 512Bytes), each of the two portions of the buffer that can be used have a size of BLEN (that is, 32-bits x BLEN div by 4). Not more than this total size of 2 times 32-bits × BLEN div by 4 can be used.
- When the size of the data block to transfer is larger than MEM_SIZE/2, only one data transfer can occur from one data bus to the other data bus at a time. The MMC/SD/SDIO host controller uses the entire data buffer as a single portion. In this mode, a bad access (MMCHS_STAT[29] BADA) is signaled when two data transfers occur from one data bus to the other data bus and vice versa at the same time.

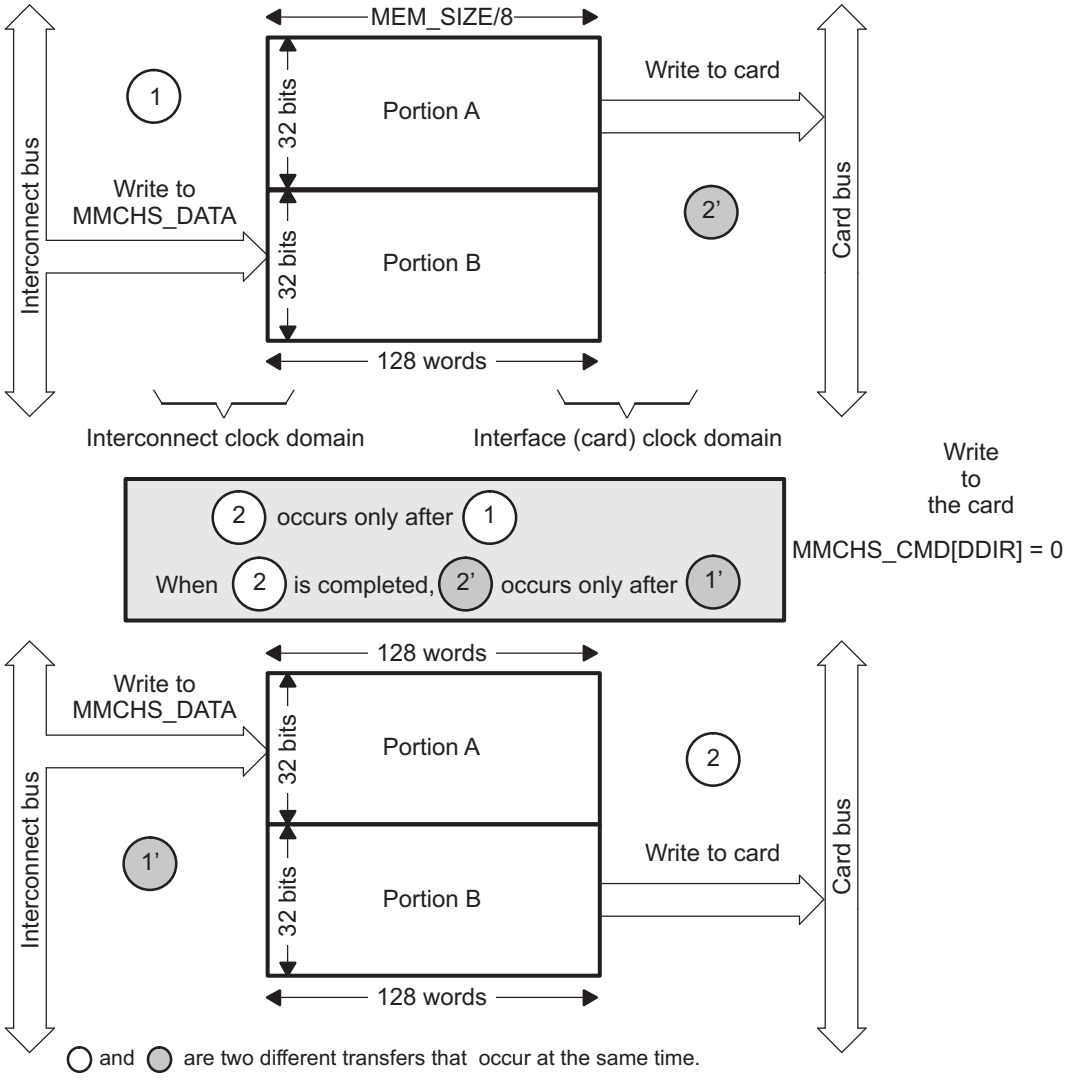

Copyright © 2011–2016, Texas Instruments Incorporated

**CAUTION**

The MMCHS_CMD[4] DDIR bit must be configured before a transfer to indicate the direction of the transfer.

Figure 18-17 shows the buffer management for writing and Figure 18-18 shows the buffer management for reading.

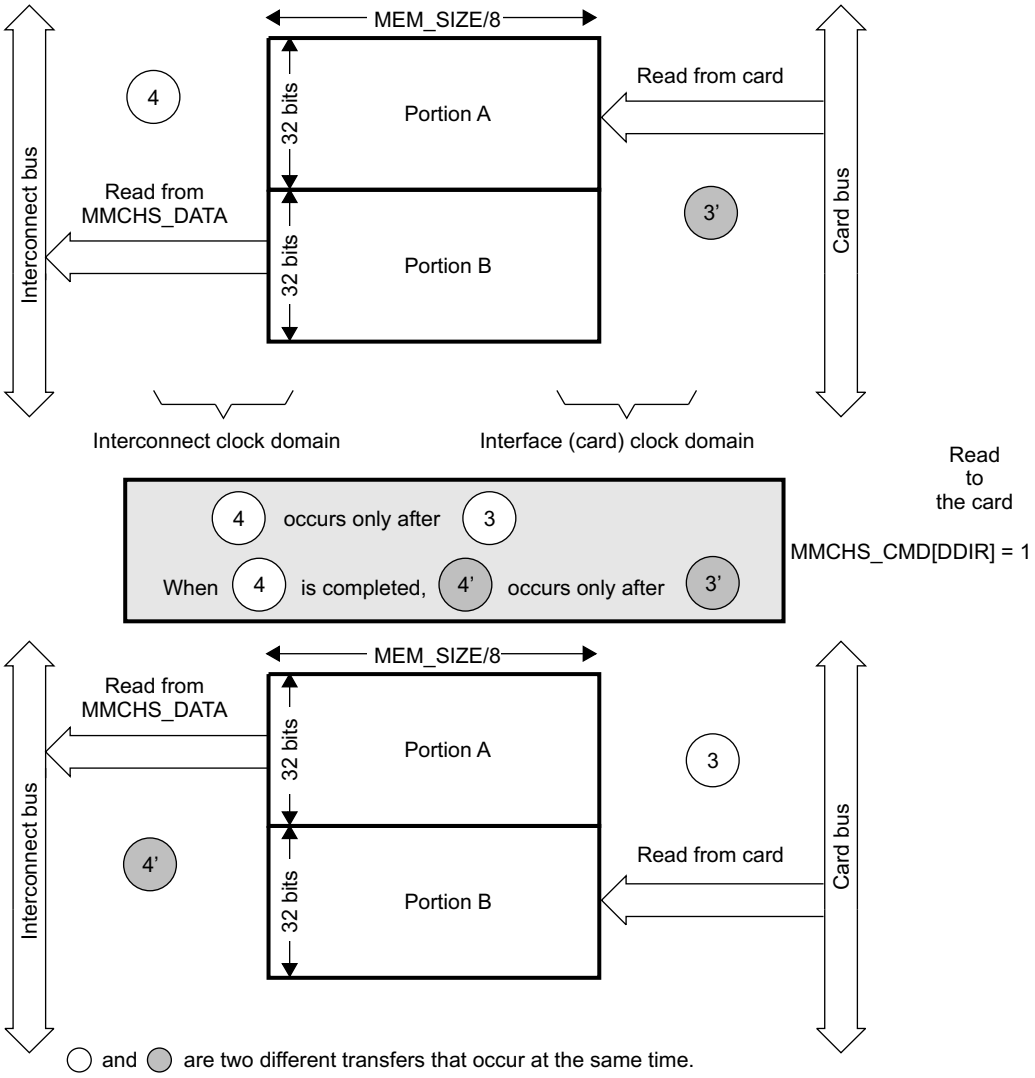
**Figure 18-17. Buffer Management for a Write**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 18-18. Buffer Management for a Read**




#### *18.2.8.1.1 Memory Size, Block Length, and Buffer Management Relationship*

The maximum block length and buffer management that can be targeted by system depend on memory depth setting.

**Table 18-6. Memory Size, BLEN, and Buffer Relationship**

| Memory Size([5:2] MEMSIZE in bytes) | 512 | 1024 | 2048 | 4096 |
|---|---|---|---|---|
| Maximum block length supported | 512 | 1024 | 2048 | 2048 |
| Double-buffering for maximum block length | N/A | BLEN <= 512 | BLEN <= 1024 | BLEN <= 2048 |
| Single-buffering for block length | BLEN<=512 | 512 < BLEN <= 1024 | 1024 < BLEN <= 2048 | N/A |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 18.2.8.1.2 Data Buffer Status

The data buffer status is defined in the following interrupt status register and status register:

- Interrupt status registers (see Figure 18-56):
  - MMCHS_STAT[29] BADA: Bad access to data space
  - MMCHS_STAT[5] BRR: Buffer read ready
  - MMCHS_STAT[4] BWR: Buffer write ready
- Status registers (see Figure 18-53):
  - MMCHS_PSTATE[11] BRE: Buffer read enable
  - MMCHS_PSTATE[10] BWE: Buffer write enable

### 18.2.9 Transfer Process

The process of a transfer is dependent on the type of command. It can be with or without a response, with or without data.

#### 18.2.9.1 Different Types of Commands

Different types of commands are specific to MMC, SD, or SDIO cards. See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, the *SDIO Card Specification, Part E1*, or the *SD Card Specification, Part A2, SD Host Controller Standard Specification* for more details.

#### 18.2.9.2 Different Types of Responses

Different types of responses are specific to MMC, SD, or SDIO cards. See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, the *SDIO Card Specification, Part E1*, or the *SD Card Specification, Part A2, SD Host Controller Standard Specification* for more details.

Table 18-7 shows how the MMC, SD, and SDIO responses are stored in the MMCHS_RSPxx registers.

**Table 18-7. MMC, SD, SDIO Responses in the MMCHS_RSP*n* Registers**

| Kind of Response | Response Field | Response Register |
|---|---|---|
| R1, R1b (normal response), R3, R4, R5, R5b, R6, R7 | RESP[39:8](1) | MMCHS_RSP10[31:0] |
| R1b (Auto CMD12 response) | RESP[39:8](1) | MMCHS_RSP76[31:0] |
| R2 | RESP[127:0](1) | MMCHS_RSP76[31:0]<br>MMCHS_RSP54[31:0]<br>MMCHS_RSP32[31:0]<br>MMCHS_RSP10[31:0] |

(1) RESP refers to the command response format described in the specifications mentioned above.

When the host controller modifies part of the MMCHS_RSPxx registers, it preserves the unmodified bits.

The host controller stores the Auto CMD12 response in the MMCHS_RSP76[31:0] register because the Host Controller may have a multiple block data DAT line transfer executing concurrently with a command. This allows the host controller to avoid overwriting the Auto CMD12 response with the command response stored in MMCHS_RSP10 register and conversely.


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.10 Transfer or Command Status and Error Reporting

Flags in the MMC/SD/SDIO host controller show status of communication with the card:

- A timeout (of a command, a data, or a response)
- A CRC

Error conditions generate interrupts. See Table 18-8 and register description for more details.

**Table 18-8. CC and TC Values Upon Error Detected**

| Error Hold in the MMCHS_STAT Register | | CC | TC | Comments |
|---|---|---|---|---|
| 29 | BADA | | | No dependency with CC or TC.<br>BADA is related to the register accesses. Its assertion is not dependent of the ongoing transfer. |
| 28 | CERR | 1 | | CC is set upon CERR. |
| 22 | DEB | | 1 | TC is set upon DEB. |
| 21 | DCRC | | 1 | TC is set upon DCRC. |
| 20 | DTO | | | DTO and TC are mutually exclusive.<br>DCRC and DEB cannot occur with DTO. |
| 19 | CIE | 1 | | CC is set upon CIE. |
| 18 | CEB | 1 | | CC is set upon CEB. |
| 17 | CCRC | 1 | | CC can be set upon CCRC - See CTO comment |
| 16 | CTO | | | CTO and CC are mutually exclusive.<br>CIE, CEB and CERR cannot occur with CTO.<br>CTO can occur at the same time as CCRC it indicates a command abort due to a contention on CMD line. In this case no CC appears. |

MMCHS_STAT[21] DCRC event can be asserted in the following conditions:

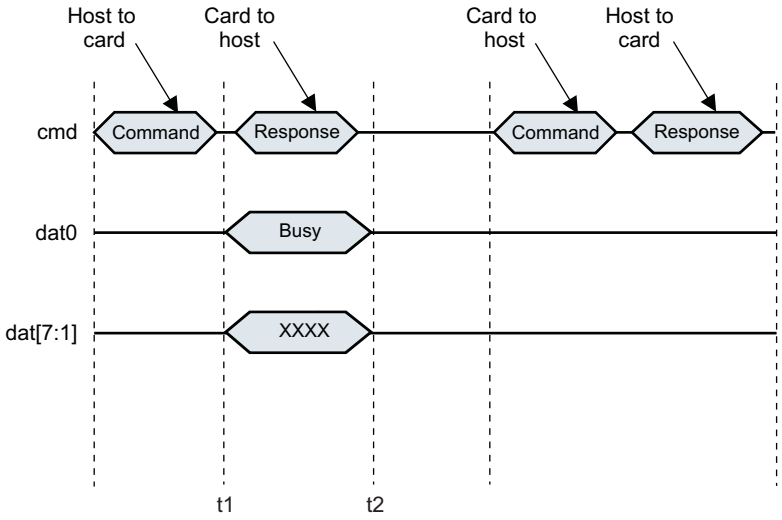
- Busy timeout for R1b, R5b response type
- Busy timeout after write CRC status
- Write CRC status timeout
- Read data timeout
- Boot acknowledge timeout


Copyright © 2011–2016, Texas Instruments Incorporated

#### 18.2.10.1 Busy Timeout for R1b, R5b Response Type

Figure 18-19 shows DCRD event condition asserted when there is a busy timeout for R1b or R5b responses.

**Figure 18-19. Busy Timeout for R1b, R5b Responses**



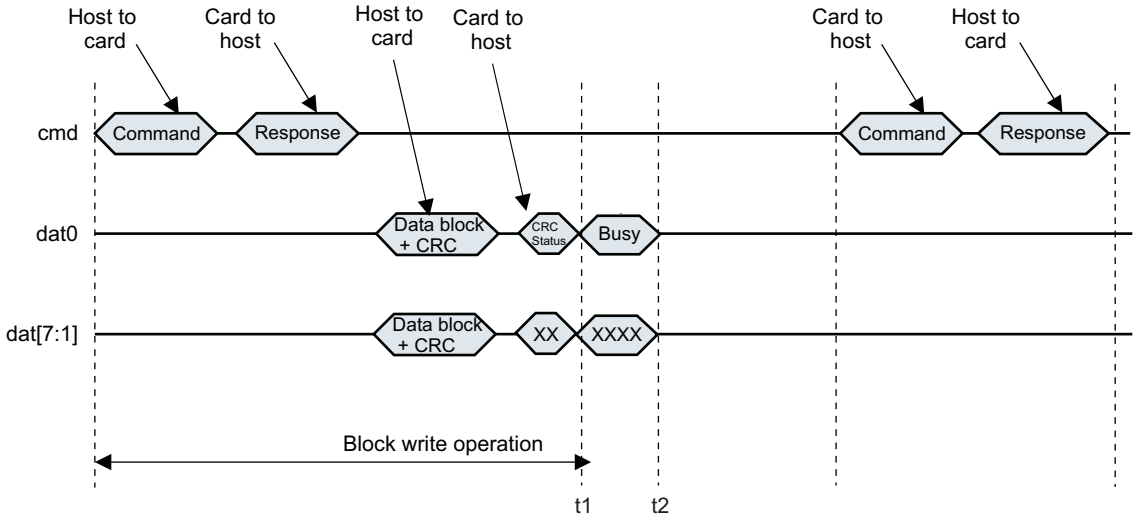

t1 - Data timeout counter is loaded and starts after R1b, R5b response type.

t2 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.

#### 18.2.10.2 Busy Timeout After Write CRC Status

Figure 18-20 shows DCRC event condition asserted when there is busy timeout after write CRC status.

**Figure 18-20. Busy Timeout After Write CRC Status**




t1 - Data timeout counter is loaded and starts after CRC status.

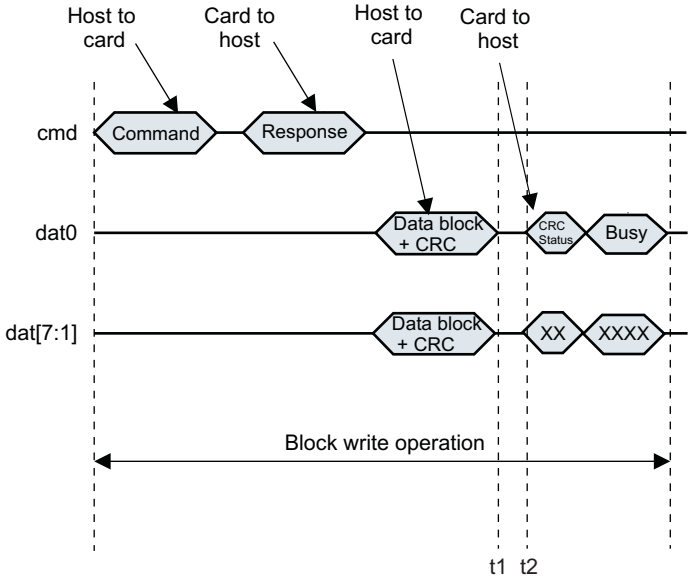
t2 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.10.3 Write CRC Status Timeout

Figure 18-21 shows DCRC event condition asserted when there is write CRC status timeout.

**Figure 18-21. Write CRC Status Timeout**



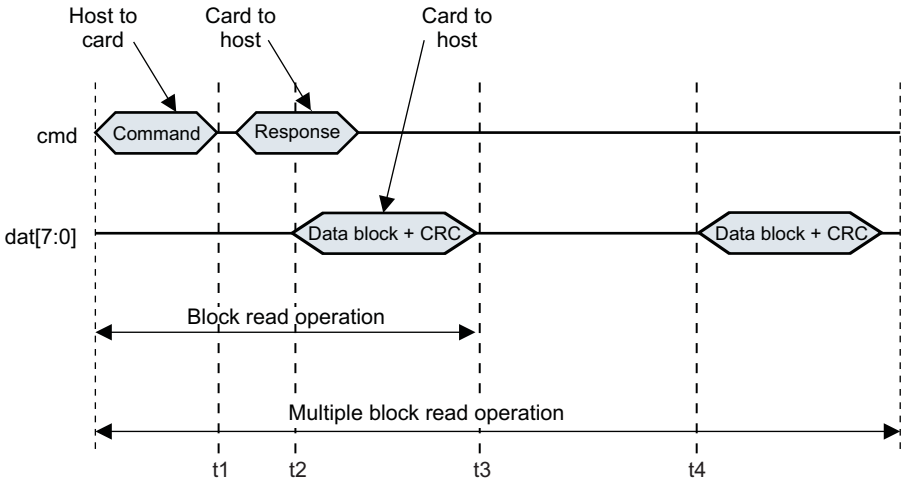

t1 - Data timeout counter is loaded and starts after Data block + CRC.

t2 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.

### 18.2.10.4 Read Data Timeout

Figure 18-22 shows DCRC event condition asserted when there is read data timeout.

**Figure 18-22. Read Data Timeout**




t1 - Data timeout counter is loaded and starts after Command transmission.

t2 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.

t3 - Data timeout counter is loaded and starts after Data block + CRC transmission.

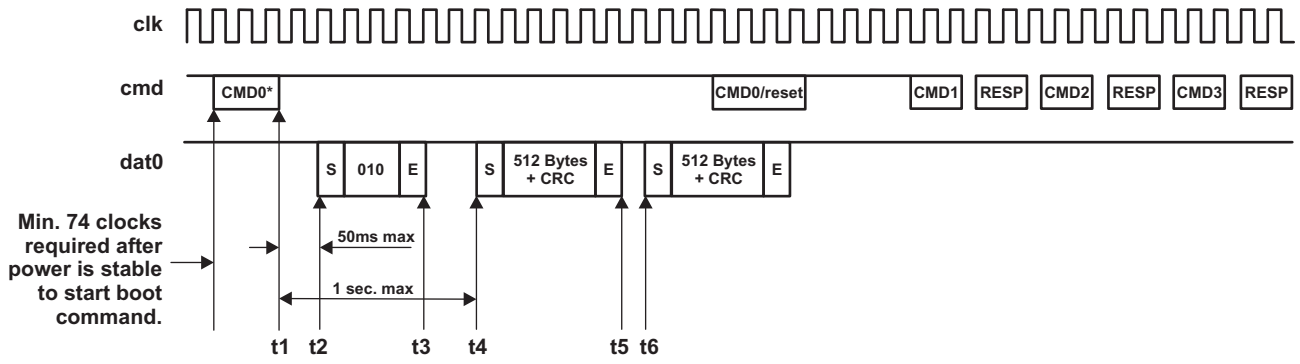
t4 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.10.5 Boot Acknowledge Timeout

Figure 18-23 shows DCRC event condition asserted when there is boot acknowledge timeout and CMD0 is used.

**Figure 18-23. Boot Acknowledge Timeout When Using CMD0**




*** Refer to MMC Specification for correct argument.**

t1 - Data timeout counter is loaded and starts after CMD0.

t2 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.

t3 - Data timeout counter is loaded and starts.

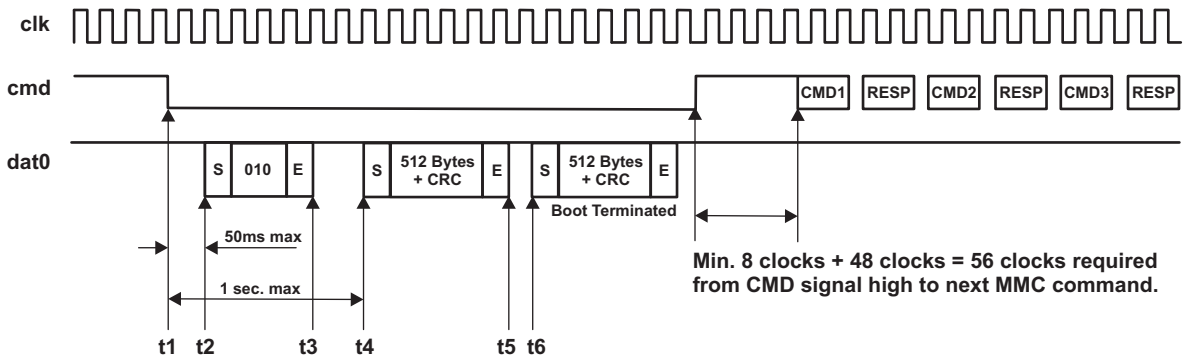
t4 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.

t5 - Data timeout counter is loaded and starts after Data + CRC transmission.

t6 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.

Figure 18-24 shows DCRC event condition asserted when there is boot acknowledge timeout and CMD line is held low.

**Figure 18-24. Boot Acknowledge Timeout When CMD Held Low**




t1 - Data timeout counter is loaded and starts after cmd line is tied to 0.

t2 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.

t3 - Data timeout counter is loaded and starts.

t4 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.

t5 - Data timeout counter is loaded and starts after Data + CRC transmission.

t6 - Data timeout counter stops and if it is 0, MMCHS_STAT[21] DCRC is generated.


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.11 Auto Command 12 Timings

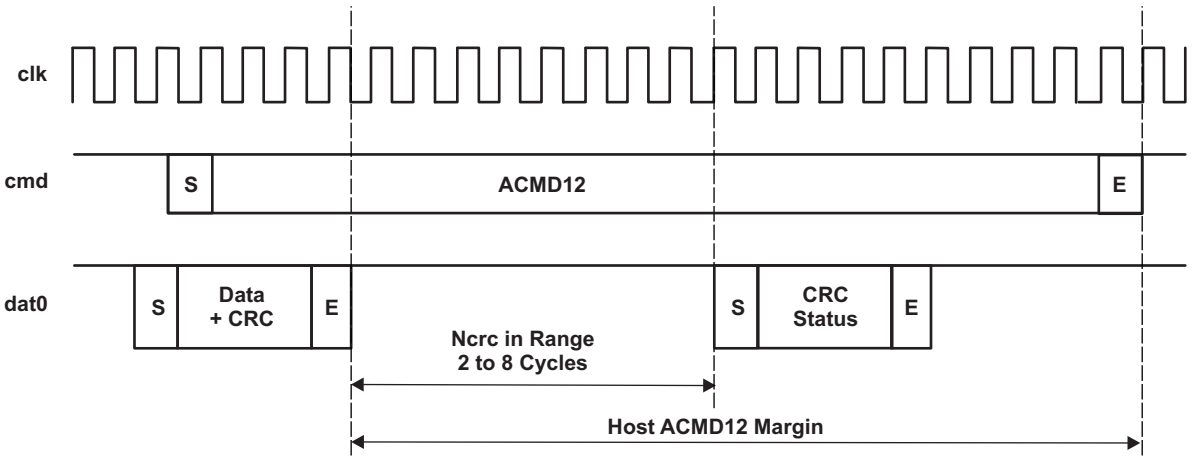
With the UHS definition of SD cards with higher frequency for MMC clocks up to 208, SD standard imposes a specific timing for Auto CMD12 "end bit" arrival.

#### 18.2.11.1 Auto Command 12 Timings During Write Transfer

A margin named Ncrc in range of 2 to 8 cycles has been defined for SDR50 and SDR104 card components for write data transfers, as auto command 12 'end bit' shall arrive after the CRC status "end bit".

Figure 18-25 shows auto CMD12 timings during write transfer.

**Figure 18-25. Auto CMD12 Timing During Write Transfer**




The Host controller has a margin of 18 clock cycles to make sure that auto CMD12 'end bit' arrives after the CRC status. This margin does not depend on MMC bus configuration, DDR or standard transfer, 1,4 or 8 bus width.

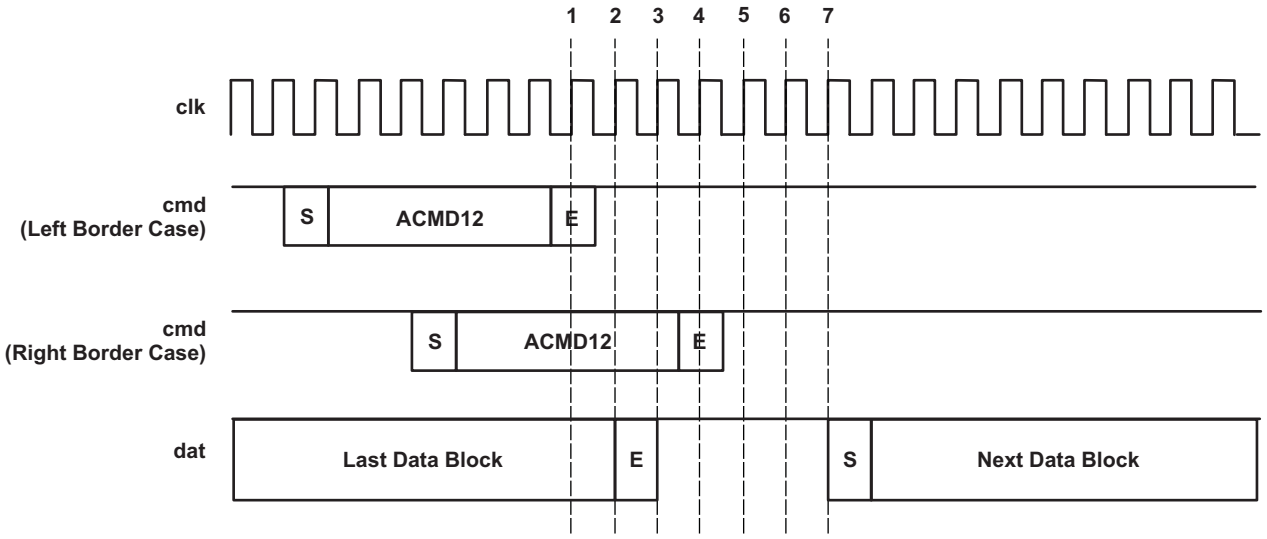

Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.11.2 Auto Command 12 Timings During Read Transfer

With UHS very-high speed cards gap timing between 2 successive cards has been extended to 4 cycles instead of 2. By the way, it gives more flexibility for Host Auto CMD12 arrival in order to receive the last complete and reliable block. SD controller only follows the 'Left Border Case' defined by SD UHS specification.

Figure 18-26 shows ACMD12 timings during read transfer.

**Figure 18-26. Auto Command 12 Timings During Read Transfer**




The Auto CMD12 arrival sent by the Host controller is not sensitive to the MMC bus configuration whether it is DDR or standard transfer and whether it is a 1, 4, or 8 bit bus width transfer.


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.12 Transfer Stop

Whenever a transfer is initiated, the transmission may be willed to stop whereas it is still not finished. Several cases can be faced depending on the transfer type:

- Multiple blocks oriented transfers (for which transfer length is known)
- Continuous stream transfers (which have an infinite length)

**NOTE:** Since the MMC/SD/SDIO controller manages transfers based on a block granularity, the buffer will accept a block only if there is enough space to completely store it. Consequently, if a block is pending in the buffer, no command will be sent to the card because the card clock will be shut off by the controller.

The MMC/SD/SDIO controller includes two features which make a transfer stop more convenient and easier to manage:

- Auto CMD12 (for MMC and SD only).

  This feature is enabled by setting the MMCHS_CMD[2] ACEN bit to 1 (this setting is relevant for a MMC/SD transfer with a known number of blocks to transfer). When the Auto CMD12 feature is enabled, the MMC/SD/SDIO controller will automatically issue a CMD12 command when the expected number of blocks has been exchanged.

- Stop at block gap

  This feature is enabled by setting the MMCHS_HCTL[16] SBGR bit to 1. When enabled, this capability holds the transfer on until the end of a block boundary. If a stop transmission is needed, software can use this pause to send a CMD12 to the card.

Table 18-9 shows the common ways to stop a transfer, indicating command to send and features to enable.

**Table 18-9. MMC/SD/SDIO Controller Transfer Stop Command Summary**

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="2">WRITE Transfer</th><th colspan="2">READ Transfer</th></tr>
<tr><th>MMC/SD</th><th>SDIO</th><th>MMC/SD</th><th>SDIO</th></tr>
<tr><td colspan="2">Single block</td><td>Transfer ends automatically<br>Wait TC</td><td>Transfer ends automatically<br>Wait TC</td><td>Transfer ends automatically<br>Wait TC</td><td>Transfer ends automatically<br>Wait TC</td></tr>
<tr><td rowspan="3">Multi blocks (finite or infinite)</td><td>Before the programmed block boundary</td><td>Send CMD12<br>Wait TC</td><td>Send CMD52<br>Wait TC</td><td>Send CMD12<br>Wait TC</td><td>Send CMD52<br>Wait TC</td></tr>
<tr><td rowspan="2">Stop at the end of the transfer (finite transfer only)</td><td rowspan="2">Auto CMD12 active<br>Transfer ends automatically<br>Wait TC</td><td rowspan="2">Set MMCHS_HCTL[16] SBGR bit to 1.<br>Send CMD52<br>Wait TC</td><td rowspan="2">Auto CMD12 active<br>Transfer ends automatically<br>Wait TC</td><td>If READ_WAIT supported<br>Stop at block gap<br>Wait TC</td></tr>
<tr><td>If READ_WAIT not supported<br>Send CMD52<br>Wait TC</td></tr>
</table>

**NOTE:** The MMC/SD/SDIO controller will send the stop command to the card on a block boundary, regardless the moment the command was written to the controller registers.


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.13 Output Signals Generation

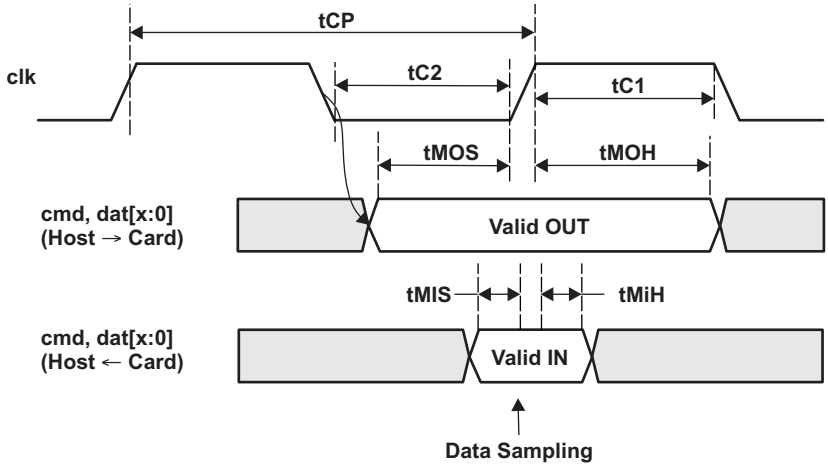
The MMC/SD/SDIO output signals can be driven on either falling edge or rising edge depending on the MMCHS_HCTL[2] HSPE bit. This feature allows to reach better timing performance, and thus to increase data transfer frequency.

#### 18.2.13.1 Generation on Falling Edge of MMC Clock

The controller is by default in this mode to maximize hold timings. In this case, MMCHS_HCTL[2] HSPE bit is cleared to 0.

Figure 18-27 shows the output signals of the module when generating from the falling edge of the MMC clock.

**Figure 18-27. Output Driven on Falling Edge**





Copyright © 2011–2016, Texas Instruments Incorporated

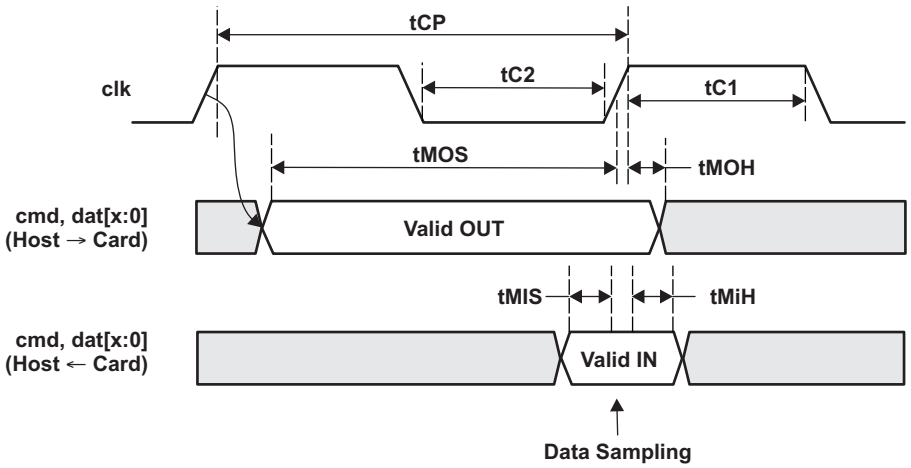
### 18.2.13.2 Generation on Rising Edge of MMC Clock

This mode increases setup timings and allows reaching higher bus frequency. This feature is activated by setting MMCHS_HCTL[2] HSPE bit to 1. The controller shall be set in this mode to support SDR transfers.

**NOTE:** Do not use this feature in Dual Data Rate mode (when MMCHS_CON[19] DDR is set to 1).

Figure 18-28 shows the output signals of the module when generating from the rising edge of the MMC clock.

**Figure 18-28. Output Driven on Rising Edge**





Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.14 Card Boot Mode Management

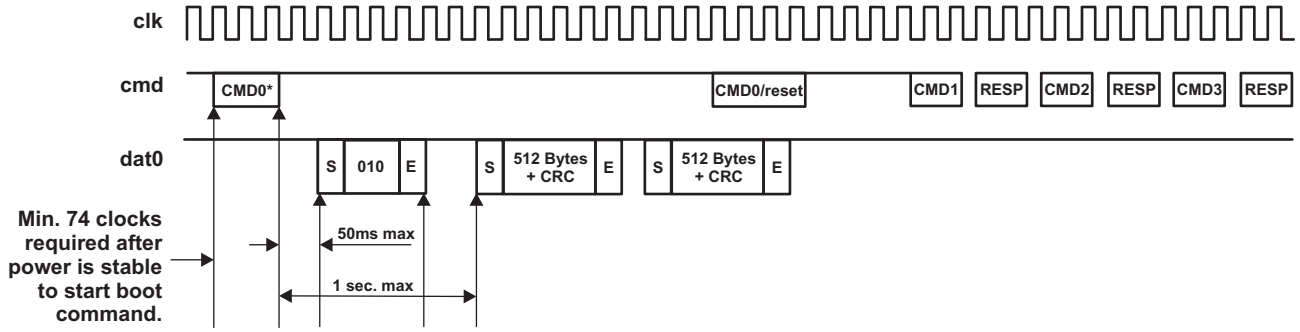
Boot Operation Mode allows the MMC/SD/SDIO host controller to read boot data from the connected slave (MMC device) by keeping CMD line low after power-on (or sending CMD0 with specific argument) before issuing CMD1. The data can be read from either boot area or user area, depending on register setting. Power-on boot defines a way for the boot-code to be accessed by the MMC/SD/SDIO host controller without an upper-level software driver, speeding the time it takes for a controller to access the boot code.

The two possible ways to issue a boot command (either issuing a CMD0 or driving the CMD line to 0 during the whole boot phase) are described in the following sections.

#### 18.2.14.1 Boot Mode Using CMD0

Figure 18-29 shows the timing diagram of a boot sequence using CMD0.

**Figure 18-29. Boot Mode With CMD0**

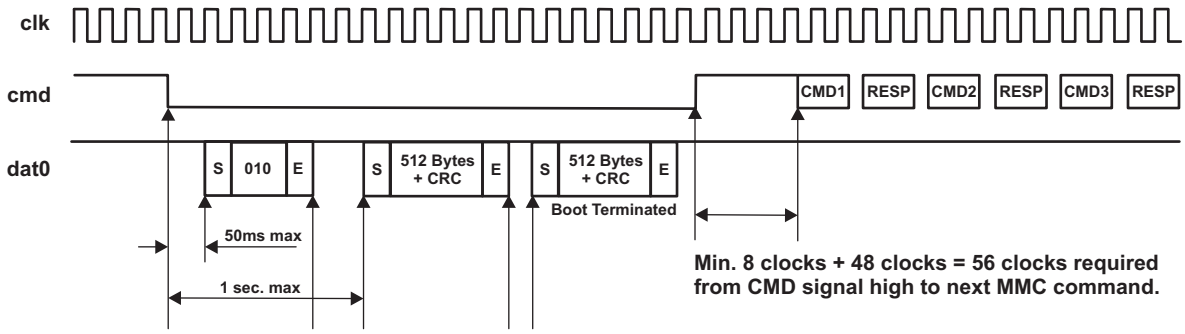



*** Refer to MMC Specification for correct argument.**

#### 18.2.14.2 Boot Mode With CMD Held Low

Figure 18-30 shows the timing diagram of a boot sequence with CMD line tied to 0.

**Figure 18-30. Boot Mode With CMD Line Tied to 0**




For more information on how to configure the MMC/SD/SDIO host controller, see the section titled Boot with CMD line tied to 0.


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.15 CE-ATA Command Completion Disable Management

The MMC/SD/SDIO controller supports CE-ATA features, in particular the detection of command completion token. When a command that requires a command completion signal (MMCHS_CON[12] CEATA and MMCHS_CMD[2] ACEN set to 1) is launched, the host system is no longer allowed to emit a new command in parallel of data transfer unless it is a command completion disable token.

The settings to emit a command completion disable token follow:

- MMCHS_CON[12] CEATA is set to 1.
- MMCHS_CON[2] HR set to 1.
- Clear the MMCHS_ARG register.
- Write into MMCHS_CMD register with value 0000 0000h.

When a command completion disable token was emitted (that is, MMCHS_STAT[0] CC received), the host system is again allowed to emit another type of command (for example a transfer abort command CMD12 to abort transfer).

A critical case can be met when command completion signal disable (CCSD) is emitted during the last data block transfer, the sequence on command line could be sent very close to command completion signal (CCS) token sent by the card.

Three cases can be met:

- CCS is receive just before CCSD is emitted:

  An interrupt CIRQ is generated with CCS detection, CCSD is transmitted to card then an interrupt CC is generated when CCSD ends. In this case, card consider the CCSD sequence.
- CCS is not generated or generated during the CCSD transfer:

  The CCS bit cannot be detected (conflict is not possible as they drive the same level on command line, then no CIRQ interrupt is generated; besides CC interrupt is generated when CCSD ends).
- CCS is generated without CCSD token required:

  Only the interrupt CIRQ is generated when CCS is detected.

### 18.2.16 Test Registers

Test registers are available to be compliant with SD Host controller specification. This feature is useful to generate interrupts manually for driver debugging. The Force Event register (MMCHS_FE) is used to control the Error Interrupt Status and Auto CMD12 Error Status. The System Test register (MMCHS_SYSTEST) is used to control the signals that connect to I/O pins when the module is configured in system test (MMCHS_CON[4] MODE = 1) mode for boundary connectivity verification.


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.2.17 MMC/SD/SDIO Hardware Status Features

Table 18-10 summarizes the MMC/SD/SDIO hardware status features.

**Table 18-10. MMC/SD/SDIO Hardware Status Features**

| Feature | Type | Register/Bit Field/Observability Control | Description |
|---|---|---|---|
| Interrupt flags | | See Section 18.2.5. | |
| CMD line signal level | Status | [24] CLEV | Indicates the level of the cmd line |
| DAT lines signal level | Status | [23:20] DLEV | Indicates the level of the data lines |
| Buffer read enable | Status | [11] BRE | Readable data exists in the buffer. |
| Buffer write enable | Status | [10] BWE | Indicates whether there is enough space in the buffer to write BLEN bytes of data |
| Read transfer active | Status | [9] RTA | This status is used for detecting completion of a read transfer. |
| Write transfer active | Status | [8] WTA | This status indicates a write transfer active. |
| Data line active | Status | [2] DLA | Indicates whether the data lines are active |
| Command Inhibit (data lines) | Status | [1] DATI | Indicates whether issuing of command using data lines is allowed |
| Command inhibit (CMD line) | Status | [0] CMDI | Indicates whether issuing of command using CMD line is allowed |


Copyright © 2011–2016, Texas Instruments Incorporated

## 18.3 Low-Level Programming Models

### *18.3.1 Surrounding Modules Global Initialization*

This section identifies the requirements of initializing the surrounding modules when the module has to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the MMC/SD/SDIO modules.

**Table 18-11. Global Init for Surrounding Modules**

| Surrounding Modules | Comments |
|---|---|
| PRCM | Module interface and functional clocks must be enabled. For more information, see Power, Reset, and Clock Management (PRCM) chapter |
| Control module | Module-specific pad muxing and configuration must be set in the control module. See Control Module chapter. |
| (optional) MPU INTC (or DSP INTC) | MPU INTC configuration must be done to enable the interrupts from the MMCHS module. See Interrupt Controller chapter |
| (optional) sDMA (or dDMA) | DMA configuration must be done to enable the module DMA channel requests. See SDMA chapter |
| (optional) Interconnect | For more information about the interconnect configuration, see Interconnect chapter |

**NOTE:** The MPU/DSP interrupt controller and the sDMA/dDMA configurations are necessary, if the interrupt and DMA based communication modes are used.

### *18.3.2 MMC/SD/SDIO Controller Initialization Flow*

The next sections outline the four steps to initialize the MMC/SD/SDIO controller:

- Initialize Clocks
- Software reset of the controller
- Set module's hardware capabilities
- Set module's Idle and Wake-Up modes

#### 18.3.2.1 Enable OCP and CLKADPI Clocks

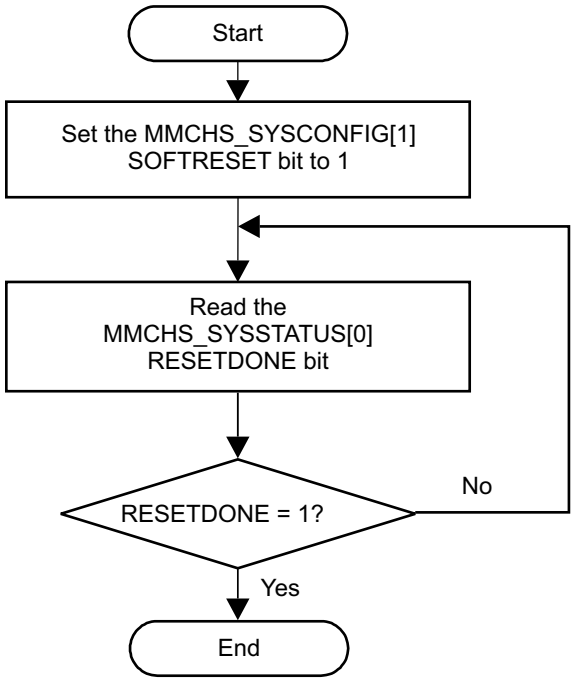
Prior to any MMCHS register access, you must enable the OCP clock and CLKADPI clock in PRCM module registers. Refer to the *Power, Reset, and Clock Management* chapter.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 18.3.2.2 SD Soft Reset Flow

Figure 18-31 shows the soft reset process of MMC/SD/SDIO controller.

**Figure 18-31. MMC/SD/SDIO Controller Software Reset Flow**




#### 18.3.2.3 Set SD Default Capabilities

Software must read capabilities (in boot ROM for instance) and is allowed to set (write) MMCHS_CAPA[26:24] before the MMC/SD/SDIO host driver is started.

#### 18.3.2.4 Wake-Up Configuration

Table 18-12 details SD controller wake-up configuration.

**Table 18-12. MMC/SD/SDIO Controller Wake-Up Configuration**

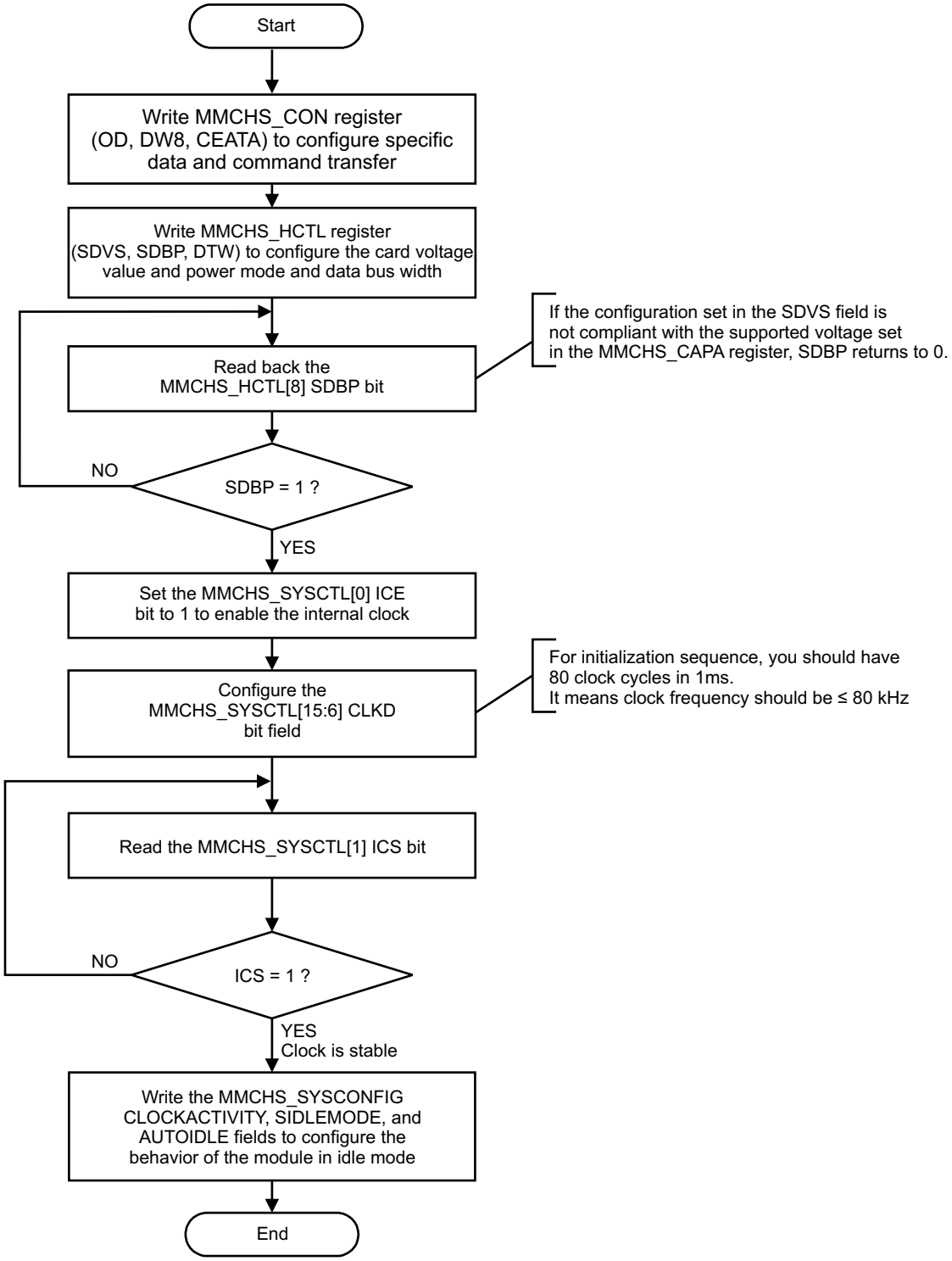
| Step | Access Type | Register/Bit Field/Programming Model | Value |
|---|---|---|---|
| Configure wake-up bit (if necessary). | W | MMCHS_SYSCONFIG[2] ENAWAKEUP | 1 |
| Enable wake-up events on SD card interrupt (if necessary). | W | MMCHS_HCTL[24] IWE | 1 |
| SDIO Card onlyEnable card interrupt (if necessary). | W | MMCHS_IE[8] CIRQENABLE | 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.3.2.5 MMC Host and Bus Configuration

Figure 18-32 details the MMC bus configuration process.

**Figure 18-32. MMC/SD/SDIO Controller Bus Configuration Flow**





Copyright © 2011–2016, Texas Instruments Incorporated

## 18.3.3 Operational Modes Configuration

### 18.3.3.1 Basic Operations for MMC/SD/SDIO Host Controller

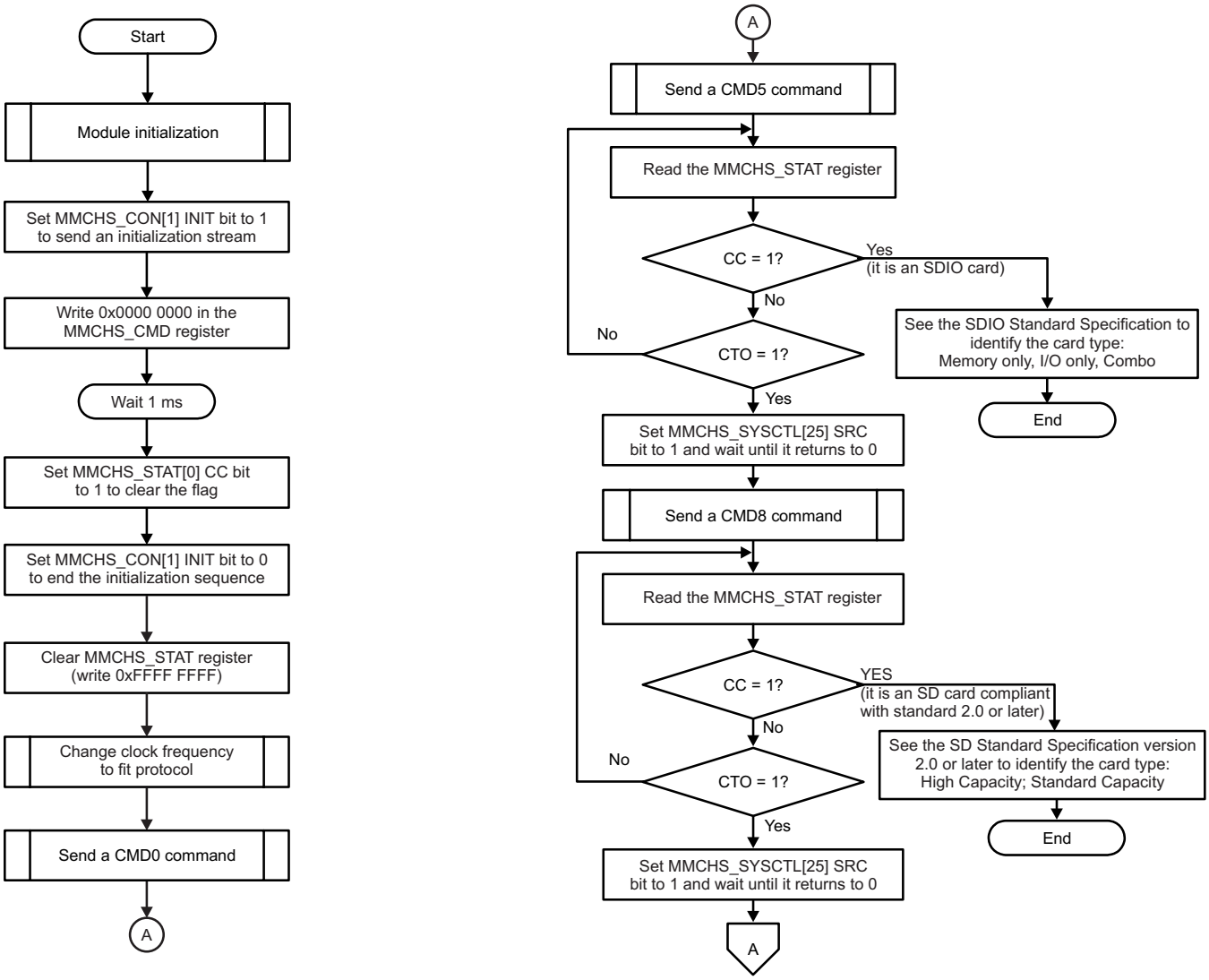
The MMC/SD/SDIO controller performs data transfers: data to card (referred to as write transfers) and data from card (referred to as read transfers).

The host controller requires transfers to run on a block-by-block basis, rather than on a DMA burst size basis. A single DMA request (or block request interrupt) is signaled for each block. Pipelining is supported as long as the block size is less than one half of the memory buffer size.

### 18.3.3.2 Card Detection, Identification, and Selection

Figure 18-33 and Figure 18-34 show the card identification and selection process.

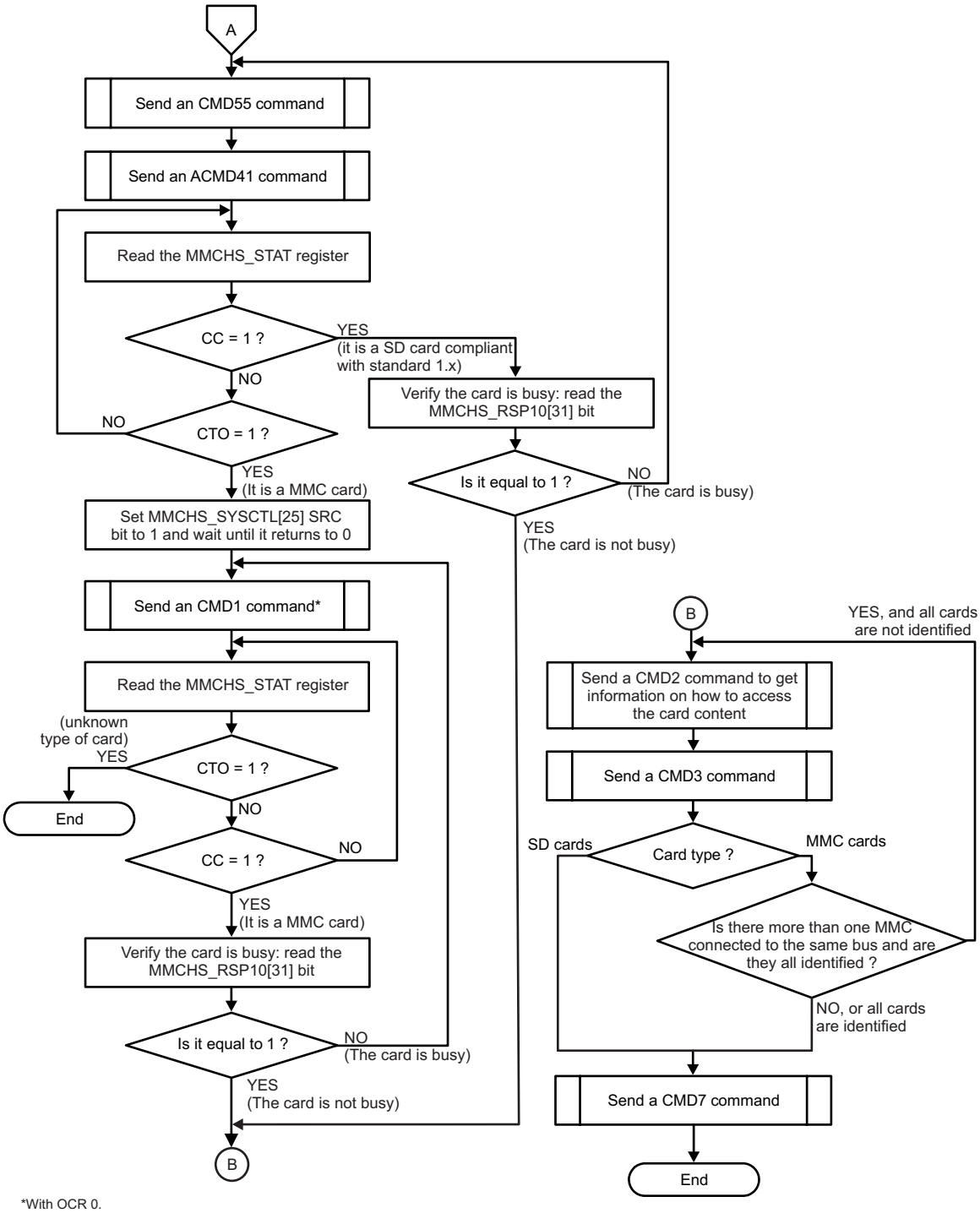
**Figure 18-33. MMC/SD/SDIO Controller Card Identification and Selection - Part 1**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 18-34. MMC/SD/SDIO Controller Card Identification and Selection - Part 2**





Copyright © 2011–2016, Texas Instruments Incorporated

## 18.4 MMC/SD/SDIO Registers

Table 18-13 lists the MMC/SD/SDIO registers.

**CAUTION**

The MMC/SD/SDIO registers are limited to 32-bit data accesses. 16-bit and 8-bit are not allowed and can corrupt register content.

**Table 18-13. MMC/SD/SDIO Registers**

| Address Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0 | MMCHS_HL_REV | IP Revision Identifier | Section 18.4.1 |
| 4h | MMCHS_HL_HWINFO | Hardware Configuration | Section 18.4.2 |
| 10h | MMCHS_HL_SYSCONFIG | Clock Management Configuration | Section 18.4.3 |
| 110h | MMCHS_SYSCONFIG | System Configuration | Section 18.4.4 |
| 114h | MMCHS_SYSSTATUS | System Status | Section 18.4.5 |
| 124h | MMCHS_CSRE | Card status response error | Section 18.4.6 |
| 128h | MMCHS_SYSTEST | System Test | Section 18.4.7 |
| 12Ch | MMCHS_CON | Configuration | Section 18.4.8 |
| 130h | MMCHS_PWCNT | Power counter | Section 18.4.9 |
| 200h | MMCHS_SDMASA | SDMA System address: | Section 18.4.10 |
| 204h | MMCHS_BLK | Transfer Length Configuration | Section 18.4.11 |
| 208h | MMCHS_ARG | Command argument | Section 18.4.12 |
| 20Ch | MMCHS_CMD | Command and transfer mode | Section 18.4.13 |
| 210h | MMCHS_RSP10 | Command Response 0 and 1 | Section 18.4.14 |
| 214h | MMCHS_RSP32 | Command Response 2 and 3 | Section 18.4.15 |
| 218h | MMCHS_RSP54 | Command Response 4 and 5 | Section 18.4.16 |
| 21Ch | MMCHS_RSP76 | Command Response 6 and 7 | Section 18.4.17 |
| 220h | MMCHS_DATA | Data | Section 18.4.18 |
| 224h | MMCHS_PSTATE | Present state | Section 18.4.19 |
| 228h | MMCHS_HCTL | Host Control | Section 18.4.20 |
| 22Ch | MMCHS_SYSCTL | SD system control | Section 18.4.21 |
| 230h | MMCHS_STAT | SD interrupt status | Section 18.4.22 |
| 234h | MMCHS_IE | SD interrupt enable | Section 18.4.23 |
| 238h | MMCHS_ISE | SD interrupt enable set | Section 18.4.24 |
| 23Ch | MMCHS_AC12 | Auto CMD12 Error Status | Section 18.4.25 |
| 240h | MMCHS_CAPA | Capabilities | Section 18.4.26 |
| 250h | MMCHS_FE | Force Event | Section 18.4.27 |
| 254h | MMCHS_ADMAES | ADMA Error Status | Section 18.4.28 |
| 258h | MMCHS_ADMASAL | ADMA System address Low bits | Section 18.4.29 |
| 25Ch | MMCHS_ADMASAH | ADMA System address High bits | Section 18.4.30 |
| 2FCh | MMCHS_REV | Versions | Section 18.4.31 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.1 IP Revision Identifier Register (MMCHS_HL_REV)

This register holds the IP Revision Identifier (X.Y.R) information.

**Figure 18-35. IP Revision Identifier Register (MMCHS_HL_REV)**

| 31 | 30 | 29 | 28 | 27 | 16 |
|---|---|---|---|---|---|
| SCHEME | | Reserved | | FUNC | |
| R-1 | | R-0 | | R-20h | |

| 15 | 11 | 10 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| R_RTL | | X_MAJOR | | CUSTOM | | Y_MINOR | |
| R-5h | | R-3h | | R-0 | | R-1 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 18-14. IP Revision Identifier Register (MMCHS_HL_REV) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | SCHEME | 1 | Used to distinguish between old Scheme and current.<br>1: Current scheme |
| 29-28 | Reserved | 0 | Reserved |
| 27-16 | FUNC | 20h | Function indicates a software compatible module family. A new Function number is assigned when there is no level of software compatibility.<br>20h: MMCHS |
| 15-11 | R_RTL | 5h | RTL version.<br>5h: Current RTL Version. |
| 10-8 | X_MAJOR | 3h | Major Revision.<br>3h: Current Major Revision |
| 7-6 | CUSTOM | 0 | Indicates a special version for a particular device. |
| 5-0 | Y_MINOR | 1 | Minor Revision<br>1: Current Minor Revision |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.2 IP Module Hardware Configuration Register (MMCHS_HL_HWINFO)

This register provides information about the IP module's hardware configuration, typically the module's HDL generics (if any).

**Figure 18-36. IP Module Hardware Configuration Register (MMCHS_HL_HWINFO)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 5 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | RETMODE | MEM_SIZE | | MERGE_MEM | MADMA_EN |
| R-0 | | R-0 | R-2h | | R-1 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 18-15. IP Module Hardware Configuration Register (MMCHS_HL_HWINFO) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | RETMODE | 0 | Indicates whether the retention mode is supported using the pin PIRFFRET.<br>0: Retention mode disabled. |
| 5-2 | MEM_SIZE | 2h | Memory size for FIFO buffer.<br>2h: Memory of 1024 bytes, max block length is 1024 bytes. |
| 1 | MERGE_MEM | 1 | Memory merged for FIFO buffer.<br>1: A single memory is used with multiplexed addresses, data, and clocks. |
| 0 | MADMA_EN | 0 | Master DMA enabled generic parameter.<br>0: No Master DMA (ADMA) management supported. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.3 Clock Management Configuration (MMCHS_HL_SYSCONFIG)

This register provides information about the clock management configuration.

**Figure 18-37. Clock Management Configuration (MMCHS_HL_SYSCONFIG**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | STANDBYMODE | | IDLEMODE | | FREEEMU | SOFTRESET |
| R-0 | | R/W-2h | | R/W-2h | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-16. Clock Management Configuration (MMCHS_HL_SYSCONFIG Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5-4 | STANDBYMODE | | Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state. |
| | | 0 | Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only. |
| | | 1h | No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only. |
| | | 2h | Smart-standby mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator.<br>IP module shall not generate (initiator-related) wakeup events. |
| | | 3h | Smart-Standby wakeup-capable mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator.<br>IP module may generate (master-related) wakeup events when in standby state. Mode is only relevant if the appropriate IP module "mwakeup" output is implemented. |
| 3-2 | IDLEMODE | | Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. |
| | | 0 | Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements. Backup mode, for debug only. |
| | | 1h | No-idle mode: local target never enters idle state. Backup mode, for debug only. |
| | | 2h | Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements.<br>IP module shall not generate (IRQ- or DMA-request-related) wakeup events. |
| | | 3h | Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements.<br>IP module may generate (IRQ- or DMA request- related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented. |
| 1 | FREEEMU | | Sensitivity to emulation (debug) suspend input signal. Functionality NOT implemented in MMCHS. |
| | | 0 | IP module is sensitive to emulation suspend |
| | | 1 | IP module is not sensitive to emulation suspend |
| 0 | SOFTRESET | | Software reset. |
| | | Read 0 | Reset done, no pending action |
| | | Write 0 | No action |
| | | Read 1 | Reset (software or other) ongoing |
| | | Write 1 | Initiate software reset |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.4 System Configuration Register (MMCHS_SYSCONFIG)

This register allows controlling various parameters of the OCP interface. The system configuration register (MMCHS_SYSCONFIG) is shown in Figure 18-38 and described in Table 18-17.

**Figure 18-38. System Configuration Register (MMCHS_SYSCONFIG)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | STANDBYMODE | | Reserved | | CLOCKACTIVITY | |
| R-0 | | R/W-2h | | R-0 | | R/W-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | SIDLEMODE | | ENAWAKEUP | SOFTRESET | AUTOIDLE |
| R-0 | | | R/W-2h | | R/W-1 | R/W-0 | R/W-1 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 18-17. System Configuration Register (MMCHS_SYSCONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-14 | Reserved | 0 | These bits are initialized to zero, and writes to them are ignored. Reads return 0. |
| 13-12 | STANDBYMODE | | Master interface power Management, standby/wait control. The bit field is only useful when generic parameter MMCHS_HL_HWINFO[0] ADMA_EN bit is set as active; otherwise, it is a read only register read a 0. |
| | | 0 | Force-standby: Mstandby is forced unconditionally. |
| | | 1h | No-standby: Mstandby is never asserted. |
| | | 2h | Smart-standby mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator.<br>IP module shall not generate (initiator-related) wake-up events. |
| | | 3h | Smart-Standby wake-up-capable mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator.<br>IP module can generate (master-related) wake-up events when in standby state. Mode is only relevant if the appropriate IP module "mwake-up" output is implemented. Functional clock is maintained. Interface clock may be switched off. |
| 11-10 | Reserved | 0 | These bits are initialized to zero, and writes to them are ignored. Reads return 0. |
| 9-8 | CLOCKACTIVITY | | Clocks activity during wake up mode period. |
| | | Bit 8 | Interface clock |
| | | Bit 9 | Functional clock |
| | | 0 | Interface and Functional clock may be switched off. |
| | | 1h | Interface clock is maintained. Functional clock may be switched-off. |
| | | 2h | Functional clock is maintained. Interface clock may be switched-off. |
| | | 3h | Interface and Functional clocks are maintained. |
| 7-5 | Reserved | 0 | These bits are initialized to zero, and writes to them are ignored. Reads return 0. |
| 4-3 | SIDLEMODE | | Power management |
| | | 0 | If an idle request is detected, the MMC/SD/SDIO host controller acknowledges it unconditionally and goes in Inactive mode. Interrupt and DMA requests are unconditionally deasserted. |
| | | 1h | If an idle request is detected, the request is ignored and the module keeps on behaving normally. |
| | | 2h | If an idle request is detected, the module will switch to wake up mode based on its internal activity, and the wake up capability can be used if the wake up capability is enabled (bit MMCHS_SYSCONFIG[2] ENAWAKEUP bit is set to 1). |
| | | 3h | Smart-idle Wake-up capable mode: local target's idle state eventually acknowledges the system's idle requests, depending on the IP modules internal requirements. IP module may generate (IRQ- or DMA-request-related) wake-up events when in idle state. Mode is only relevant if the appropriate IP module "swake-up" output(s) is (are) implemented. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-17. System Configuration Register (MMCHS_SYSCONFIG) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 2 | ENAWAKEUP | | Wake-up feature control |
| | | 0 | Wake-up capability is disabled. |
| | | 1 | Wake-up capability is enabled. |
| 1 | SOFTRESET | | Software reset. The bit is automatically reset by the hardware. During reset, it always returns 0. |
| | | Read 0 | Normal mode |
| | | Write 0 | No effect |
| | | Read 1 | The module is reset. |
| | | Write 1 | Trigger a module reset. |
| 0 | AUTOIDLE | | Internal Clock gating strategy |
| | | 0 | Clocks are free-running. |
| | | 1 | Automatic clock gating strategy is applied, based on the interconnect and MMC interface activity. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.5 System Status Register (MMCHS_SYSSTATUS)

This register provides status information about the module excluding the interrupt status information. The system status register (MMCHS_SYSSTATUS) is shown in Figure 18-39 and described in Table 18-18.

**Figure 18-39. System Status Register (MMCHS_SYSSTATUS)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | RESETDONE |
| R-0 | | R-X |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-18. System Status Register (MMCHS_SYSSTATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 0 | RESETDONE | | Internal Reset Monitoring. Notethe debounce clock , the interface clock and the functional clock shall be provided to the MMC/SD/SDIO host controller to allow the internal reset monitoring. |
| | | Read 0 | Internal module reset is on-going |
| | | Read 1 | Reset completed |

### 18.4.6 Card Status Response Error (MMCHS_CSRE)

This register enables the host controller to detect card status errors of response type R1, R1b for all cards and of R5, R5b and R6 response for cards types SD or SDIO. When a bit MMCHS_CSRE[i] is set to 1, if the corresponding bit at the same position in the response MMCHS_RSP10[I] is set to 1, the host controller indicates a card error (MMCHS_STAT[28] CERR bit) interrupt status to avoid the host driver reading the response register (MMCHS_RSP10).

**NOTE:** No automatic card error detection for autoCMD12 is implemented; the host system has to check autoCMD12 response register (MMCHS_RSP76) for possible card errors.

The card status response error (MMCHS_CSRE) is shown in Figure 18-40 and described in Table 18-19.

**Figure 18-40. Card Status Response Error (MMCHS_CSRE)**

| 31 | 0 |
|---|---|
| CSRE | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-19. Card Status Response Error (MMCHS_CSRE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | CSRE | 0-FFFF FFFFh | Card status response error |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.7 System Test Register (MMCHS_SYSTEST)

This register is used to control the signals that connect to I/O pins when the module is configured in system test (SYSTEST) mode for boundary connectivity verification.

**NOTE:** In SYSTEST mode, a write into MMCHS_CMD register will not start a transfer. The buffer behaves as a stack accessible only by the local host (push and pop operations). In this mode, the Transfer Block Size (MMCHS_BLK[10:0] BLEN bits) and the Blocks count for current transfer (MMCHS_BLK[31:16] NBLK bits) are needed to generate a Buffer write ready interrupt (MMCHS_STAT[4] BWR bit) or a Buffer read ready interrupt (MMCHS_STAT[5] BRR bit) and DMA requests if enabled.

The system test register (MMCHS_SYSTEST) is shown in Figure 18-41 and described in Table 18-20.

**Figure 18-41. System Test Register (MMCHS_SYSTEST)**

| 31 | | | | | | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | OBI |
| R-0 | | | | | | | R/W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SDCD | SDWP | WAKD | SSB | D7D | D6D | D5D | D4D |
| R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| D3D | D2D | D1D | D0D | DDIR | CDAT | CDIR | MCKD |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-20. System Test Register (MMCHS_SYSTEST) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved bit field. Do not write any value. Reads return 0. |
| 16 | OBI | | Out-of-band interrupt (OBI) data value. |
| | | 0 | The out-of-band interrupt pin is driven low. |
| | | 1 | The out-of-band interrupt pin is driven high. |
| 15 | SDCD | | Card detect input signal (SDCD) data value |
| | | 0 | The card detect pin is driven low. |
| | | 1 | The card detect pin is driven high. |
| 14 | SDWP | | Write protect input signal (SDWP) data value |
| | | 0 | The write protect pin SDWP is driven low. |
| | | 1 | he write protect pin SDWP is driven high. |
| 13 | WAKD | | Wake request output signal data value |
| | | Read 0 | No action. Returns 0. |
| | | Write 0 | The pin SWAKEUP is driven low. |
| | | Read 1 | No action. Returns 1. |
| | | Write 1 | The pin SWAKEUP is driven high. |
| 12 | SSB | | Set status bit. This bit must be cleared prior attempting to clear a status bit of the interrupt status register (MMCHS_STAT). |
| | | Read 0 | No action. Returns 0. |
| | | Write 0 | Clear this SSB bit field. Writing 0 does not clear already set status bits. |
| | | Read 1 | No action. Returns 1. |
| | | Write 1 | Force to 1 all status bits of the interrupt status register (MMCHS_STAT) only if the corresponding bit field in the Interrupt signal enable register (MMCHS_ISE) is set. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-20. System Test Register (MMCHS_SYSTEST) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 11 | D7D | | DAT7 input/output signal data value |
| | | Read 0 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT7 line (low).<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. |
| | | Write 0 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT7 line is driven low.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| | | Read 1 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT7 line (high)<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. |
| | | Write 1 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT7 line is driven high.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| 10 | D6D | | DAT6 input/output signal data value |
| | | Read 0 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT6 line (low).<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. |
| | | Write 0 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT6 line is driven low.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| | | Read 1 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT6 line (high)<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. |
| | | Write 1 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT6 line is driven high.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| 9 | D5D | | DAT5 input/output signal data value |
| | | Read 0 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT5 line (low).<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. |
| | | Write 0 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT5 line is driven low.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| | | Read 1 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT5 line (high)<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. |
| | | Write 1 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT5 line is driven high.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| 8 | D4D | | DAT4 input/output signal data value |
| | | Read 0 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT4 line (low).<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. |
| | | Write 0 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT4 line is driven low.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| | | Read 1 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT4 line (high)<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. |
| | | Write 1 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT4 line is driven high.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| 7 | D3D | | DAT3 input/output signal data value |
| | | Read 0 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT3 line (low).<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. |
| | | Write 0 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT3 line is driven low.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| | | Read 1 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT3 line (high)<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. |
| | | Write 1 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT3 line is driven high.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-20. System Test Register (MMCHS_SYSTEST) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 6 | D2D | | DAT2 input/output signal data value |
| | | Read 0 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT2 line (low).<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. |
| | | Write 0 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT2 line is driven low.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| | | Read 1 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT2 line (high)<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. |
| | | Write 1 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT2 line is driven high.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| 5 | D1D | | DAT1 input/output signal data value |
| | | Read 0 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT1 line (low).<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. |
| | | Write 0 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT1 line is driven low.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| | | Read 1 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT1 line (high)<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. |
| | | Write 1 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT1 line is driven high.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| 4 | D0D | | DAT0 input/output signal data value |
| | | Read 0 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT0 line (low).<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0. |
| | | Write 0 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT0 line is driven low.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| | | Read 1 | If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT0 line (high)<br>If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1. |
| | | Write 1 | If MMCHS_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT0 line is driven high.<br>If MMCHS_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect. |
| 3 | DDIR | | Control of the DAT[7:0] pins direction |
| | | Read 0 | No action. Returns 0. |
| | | Write 0 | The DAT lines are outputs (host to card). |
| | | Read 1 | No action. Returns 1. |
| | | Write 1 | The DAT lines are inputs (card to host). |
| 2 | CDAT | | CMD input/output signal data value |
| | | Read 0 | If MMCHS_SYSTEST[1] CDIR bit = 1 (input mode direction), returns the value on the CMD line (low).<br>If MMCHS_SYSTEST[1] CDIR bit = 0 (output mode direction), returns 0 . |
| | | Write 0 | If MMCHS_SYSTEST[1] CDIR bit = 0 (output mode direction), the CMD line is driven low.<br>If MMCHS_SYSTEST[1] CDIR bit = 1 (input mode direction), no effect. |
| | | Read 1 | If MMCHS_SYSTEST[1] CDIR bit = 1 (input mode direction), returns the value on the CMD line (high)<br>If MMCHS_SYSTEST[1] CDIR bit = 0 (output mode direction), returns 1 . |
| | | Write 1 | If MMCHS_SYSTEST[1] CDIR bit = 0 (output mode direction), the CMD line is driven high.<br>If MMCHS_SYSTEST[1] CDIR bit = 1 (input mode direction), no effect. |
| 1 | CDIR | | Control of the CMD pin direction |
| | | Read 0 | No action. Returns 0. |
| | | Write 0 | The CMD line is an output (host to card). |
| | | Read 1 | No action. Returns 1. |
| | | Write 1 | The CMD line is an input (card to host) . |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-20. System Test Register (MMCHS_SYSTEST) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | MCKD | | MMC clock output signal data value |
| | | Read 0 | No action. Returns 0. |
| | | Write 0 | The output clock is driven low. |
| | | Read 1 | No action. Returns 1. |
| | | Write 1 | The output clock is driven high. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.8 Configuration Register (MMCHS_CON)

This register is used:

- To select the functional mode or the SYSTEST mode for any card
- To send an initialization sequence to any card
- To enable the detection on the mmc_dat[1] signal of a card interrupt for SDIO cards only

It also configures:

- The parameters related to the card detect and write protect input signals

The configuration register (MMCHS_CON) is shown in Figure 18-42 and described in Table 18-21.

**Figure 18-42. Configuration Register (MMCHS_CON)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | SDMA_LnE | DMA_MnS | DDR | BOOT_CF0 | BOOT_ACK | CLKEXTFREE |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| PADEN | Reserved | | CEATA | CTPL | DVAL | | WPP |
| R/W-0 | R-0 | | R/W-0 | R/W-0 | R/W-3h | | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CDP | MIT | DW8 | MODE | STR | HR | INIT | OD |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND:R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-21. Configuration Register (MMCHS_CON) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-22 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 21 | SDMA_LnE | | Slave DMA Level/Edge Request. The waveform of the DMA request can be configured either edge sensitive with early de-assertion on first access to MMCHS_DATA register or late de-assertion, request remains active until last allowed data written into MMCHS_DATA. |
| | | 0 | Slave DMA edge sensitive. |
| | | 1 | Slave DMA level sensitive. |
| 20 | DMA_MnS | | DMA Master or Slave selection. When this bit is set and the controller is configured to use the DMA, Ocp master interface is used to get datas from system using ADMA2 procedure (direct access to the memory).<br>This option is only available if generic parameter MADMA_EN is asserted to 1. |
| | | 0 | The controller is slave on data transfers with system. |
| | | 1 | Not available on this device. |
| 19 | DDR | | Dual Data Rate mode. When this register is set, the controller uses both clock edge to emit or receive data. Odd bytes are transmitted on falling edges and even bytes are transmitted on rise edges. It only applies on Data bytes and CRC, Start, end bits and CRC status are kept full cycle. This bit field is only meaningful and active for even clock divider ratio of MMCHS_SYSCTL[CLKD], it is insensitive to MMCHS_HCTL[HSPE] setting |
| | | 0 | Standard mode: Data are transmitted on a single edge. |
| | | 1 | Data Bytes and CRC are transmitted on both edges. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-21. Configuration Register (MMCHS_CON) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 18 | BOOT_CF0 | | Boot Status Supported. This register is set when the CMD line needs to be forced to 0 for a boot sequence. CMD line is driven to 0 after writing in MMCHS_CMD. The line is released when this bit field is de-asserted and aborts data transfer in case of a pending transaction. |
| | | Read 0 | CMD line not forced. |
| | | Read 1 | CMD line is released when it was previously forced to 0 by a boot sequence. |
| | | Write 0 | CMD line forced to 0 is enabled. |
| | | Write 1 | CMD line forced to 0 is enabled and will be active after writing into MMCHS_CMD register. |
| 17 | BOOT_ACK | | Book acknowledge received. When this bit is set the controller should receive a boot status on DAT0 line after next command issued. If no status is received a data timeout will be generated. |
| | | 0 | No acknowledge to be received. |
| | | 1 | A boot status will be received on DAT0 line after issuing a command. |
| 16 | CLKEXTFREE | | External clock free running. This register is used to maintain card clock out of transfer transaction to enable slave module (for example to generate a synchronous interrupt on mmc_dat[1] ). The Clock will be maintain only if MMCHS_SYSCTL[2] CEN bit is set. |
| | | 0 | External card clock is cut off outside active transaction period. |
| | | 1 | External card clock is maintain even out of active transaction period only if MMCHS_SYSCTL[2] CEN bit is set. |
| 15 | PADEN | | Control power for MMC lines. This register is only useful when MMC PADs contain power saving mechanism to minimize its leakage power. It works as a GPIO that directly control the ACTIVE pin of PADs. Excepted for mmc_dat[1] , the signal is also combine outside the module with the dedicated power control MMCHS_CON[11] CTPL bit. |
| | | 0 | ADPIDLE module pin is not forced, it is automatically generated by the MMC fsms. |
| | | 1 | ADPIDLE module pin is forced to active state |
| 14-13 | Reserved | 0 | Any writes to these bits will result in 0. |
| 12 | CEATA | | CE-ATA control mode (MMC cards compliant with CE-ATA). By default, this bit is cleared to 0. It is used to indicate that next commands are considered as specific CE-ATA commands that potentially use 'command completion' features. |
| | | 0 | Standard MMC/SD/SDIO mode. |
| | | 1 | CE-ATA mode. Next commands are considered as CE-ATA commands. |
| 11 | CTPL | | Control Power for mmc_dat[1] line (SD cards). By default, this bit is cleared to 0 and the host controller automatically disables all the input buffers outside of a transaction to minimize the leakage current.<br>SDIO cards. When this bit is set to 1, the host controller automatically disables all the input buffers except the buffer of mmc_dat[1] outside of a transaction in order to detect asynchronous card interrupt on mmc_dat[1] line and minimize the leakage current of the buffers. |
| | | 0 | Disable all the input buffers outside of a transaction. |
| | | 1 | Disable all the input buffers except the buffer of mmc_dat[1] outside of a transaction. |
| 10-9 | DVAL | | Debounce filter value (all cards). This register is used to define a debounce period to filter the card detect input signal (SDCD). The usage of the card detect input signal (SDCD) is optional and depends on the system integration and the type of the connector housing that accommodates the card. |
| | | 0 | 33 us debounce period |
| | | 1h | 231 us debounce period |
| | | 2h | 1 ms debounce period |
| | | 3h | 8.4 ms debounce period |
| 8 | WPP | | Write protect polarity (SD and SDIO cards only). This bit selects the active level of the write protect input signal (SDWP). The usage of the write protect input signal (SDWP) is optional and depends on the system integration and the type of the connector housing that accommodates the card. |
| | | 0 | Active high level |
| | | 1 | Active low level |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-21. Configuration Register (MMCHS_CON) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7 | CDP | | Card detect polarity (all cards). This bit selects the active level of the card detect input signal (SDCD). The usage of the card detect input signal (SDCD) is optional and depends on the system integration and the type of the connector housing that accommodates the card. |
| | | 0 | Active-high level |
| | | 1 | Active-low level |
| 6 | MIT | | MMC interrupt command (MMC cards only). This bit must be set to 1, when the next write access to the command register (MMCHS_CMD) is for writing a MMC interrupt command (CMD40) requiring the command timeout detection to be disabled for the command response. |
| | | 0 | Command timeout enabled. |
| | | 1 | Command timeout disabled. |
| 5 | DW8 | | 8-bit mode MMC select (MMC cards only). For SD/SDIO cards, this bit must be cleared to 0. For MMC card, this bit must be set following a valid SWITCH command (CMD6) with the correct value and extend CSD index written in the argument. Prior to this command, the MMC card configuration register (CSD and EXT_CSD) must be verified for compliancy with MMC standard specification. |
| | | 0 | 1-bit or 4-bit data width |
| | | 1 | 8-bit data width |
| 4 | MODE | | Mode select (all cards). This bit selects the functional mode or SYSTEST mode. |
| | | 0 | Functional mode: Transfers to the MMC/SD/SDIO cards follow the card protocol. The MMC clock is enabled. MMC/SD transfers are operated under the control of the MMCHS_CMD register. |
| | | 1 | SYSTEST mode: The signal pins are configured as general-purpose input/output and the 1024-byte buffer is configured as a stack memory accessible only by the local host or system DMA. The pins retain their default type (input, output or in-out). SYSTEST mode is operated under the control of the SYSTEST register. |
| 3 | STR | | Stream command (MMC cards only). This bit must be set to 1 only for the stream data transfers (read or write) of the adtc commands. Stream read is a class 1 command (CMD11READ_DAT_UNTIL_STOP). Stream write is a class 3 command (CMD20WRITE_DAT_UNTIL_STOP). |
| | | 0 | Block oriented data transfer |
| | | 1 | Stream oriented data transfer |
| 2 | HR | | Broadcast host response (MMC cards only). This register is used to force the host to generate a 48-bit response for bc command type. It can be used to terminate the interrupt mode by generating a CMD40 response by the core. In order to have the host response to be generated in open drain mode, the register MMCHS_CON[0] OD bit must be set to 1. When MMCHS_CON[12] CEATA bit is set to 1 and MMCHS_ARG cleared to 0, when writing 0000 0000h into MMCHS_CMD register, the host controller performs a 'command completion signal disable' token (that is, mmchs_cmd line held to 0 during 47 cycles followed by a 1). |
| | | 0 | The host does not generate a 48-bit response instead of a command. |
| | | 1 | The host generates a 48-bit response instead of a command or a command completion signal disable token. |
| 1 | INIT | | Send initialization stream (all cards). When this bit is set to 1, and the card is idle, an initialization sequence is sent to the card. An initialization sequence consists of setting the mmchs_cmd line to 1 during 80 clock cycles. The initialization sequence is mandatory - but it is not required to do it through this bit - this bit makes it easier. Clock divider (MMCHS_SYSCTL[15:6] CLKD bits) should be set to ensure that 80 clock periods are greater than 1ms.<br>Note that in this mode, there is no command sent to the card and no response is expected. A command complete interrupt will be generated once the initialization sequence is completed. MMCHS_STAT[0] CC bit can be polled. |
| | | 0 | The host does not send an initialization sequence |
| | | 1 | The host sends an initialization sequence |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-21. Configuration Register (MMCHS_CON) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | OD | | Card open drain mode (MMC cards only). This bit must be set to 1 for MMC card commands 1, 2, 3, and 40, and if the MMC card bus is operating in open-drain mode during the response phase to the command sent. Typically, during card identification mode when the card is in idle, ready, or ident state. It is also necessary to set this bit to 1, for a broadcast host response (see Broadcast host response register MMCHS_CON[2] HR bit) |
| | | 0 | No open drain |
| | | 1 | Open drain or broadcast host response |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.9 Power Counter Register (MMCHS_PWCNT)

This register is used to program a mmc counter to delay command transfers after activating the PAD power, this value depends on PAD characteristics and voltage. The power counter register (MMCHS_PWCNT) is shown in Figure 18-43 and described in Table 18-22.

**Figure 18-43. Power Counter Register (MMCHS_PWCNT)**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| Reserved | | PWRCNT | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-22. Power Counter Register (MMCHS_PWCNT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | These bits are initialized to zero, and writes to them are ignored. Reads return 0. |
| 15-0 | PWRCNT | | Power counter register. This register is used to introduce a delay between the PAD ACTIVE pin assertion and the command issued. |
| | | 0 | No additional delay added |
| | | 1h | TCF delay (card clock period) |
| | | 2h | TCF × 2 delay (card clock period) |
| | | : | |
| | | FFFEh | TCF × 65534 delay (card clock period) |
| | | FFFFh | TCF × 65535 delay (card clock period) |

### 18.4.10 SDMA System Address (MMCHS_SDMASA)

This register contains the system memory address for a SDMA transfer. When the Host Controller stops a SDMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (that is, after a transaction has stopped). The SDMA System address (MMCHS_SDMASA) is shown in Figure 18-40 and described in Table 18-19.

**Figure 18-44. Card Status Response Error (MMCHS_SDMASA)**

| 31 | 0 |
|---|---|
| SDMA_SYSADDR | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 18-23. Card Status Response Error (MMCHS_SDMASA) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SDMA_SYSADDR | | Read operations during transfers may return an invalid value. The Host Driver shall initialize this register before starting a SDMA transaction. After SDMA has stopped, the next system address of the next contiguous data position can be read from this register.<br>The SDMA transfer waits at the every boundary specified by the Host SDMA Buffer Boundary in the Block Size register. The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver sets the next system address of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller restarts the SDMA transfer. When restarting SDMA by the Resume command or by setting Continue Request in the Block Gap Control register, the Host Controller shall start at the next contiguous address stored here in the SDMA System Address register. ADMA does not use this register. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.11 Transfer Length Configuration Register (MMCHS_BLK)

This register shall be used for any card. MMCHS_BLK[BLEN] is the block size register. MMCHS_BLK[NBLK] is the block count register.

The transfer length configuration register (MMCHS_BLK) is shown in Figure 18-45 and described in Table 18-24.

**Figure 18-45. Transfer Length Configuration Register (MMCHS_BLK)**

| 31 16 | 15 12 | 11 0 |
|---|---|---|
| NBLK | Reserved | BLEN |
| R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-24. Transfer Length Configuration Register (MMCHS_BLK) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | NBLK | | Blocks count for current transfer. This register is enabled when Block Count Enable (MMCHS_CMD[1] BCE bit) is set to 1 and is valid only for multiple block transfers. Setting the block count to 0 results no data blocks being transferred.<br>Note: The host controller decrements the block count after each block transfer and stops when the count reaches zero.<br>This register can be accessed only if no transaction is executing (that is, after a transaction has stopped). Read operations during transfers may return an invalid value and write operation will be ignored. In suspend context, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, The local host shall restore the previously saved block count. |
| | | 0 | Stop count |
| | | 1h | 1 block |
| | | 2h | 2 blocks |
| | | : | |
| | | FFFFh | 65535 blocks |
| 15-12 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 11-0 | BLEN | | Transfer block size. This register specifies the block size for block data transfers. Read operations during transfers may return an invalid value, and write operations are ignored. When a CMD12 command is issued to stop the transfer, a read of the BLEN field after transfer completion (MMCHS_STAT[1] TC bit set to 1) will not return the true byte number of data length while the stop occurs but the value written in this register before transfer is launched. |
| | | 0 | No data transfer |
| | | 1h | 1 byte block length |
| | | 2h | 2 bytes block length |
| | | 3h | 3 bytes block length |
| | | 1FFh | 511 bytes block length |
| | | 200h | 512 bytes block length |
| | | 7FFh | 2047 bytes block length |
| | | 800h | 2048 bytes block length |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.12 Command Argument Register (MMCHS_ARG)

This register contains command argument specified as bit 39-8 of Command-Format. These registers must be initialized prior to sending the command itself to the card (write action into the register MMCHS_CMD register). Only exception is for a command index specifying stuff bits in arguments, making a write unnecessary.

The command argument register (MMCHS_ARG) is shown in Figure 18-46 and described in Table 18-25.

**Figure 18-46. Command Argument Register (MMCHS_ARG)**

| 31 | 0 |
|---|---|
| ARG | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-25. Command Argument Register (MMCHS_ARG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ARG | 0-FFFF FFFFh | Command argument bits [31:0] . |

### 18.4.13 Command and Transfer Mode Register (MMCHS_CMD)

- MMCHS_CMD[31:16] = the command register
- MMCHS_CMD[15:0] = the transfer mode

This register configures the data and command transfers. A write into the most significant byte send the command. A write into MMCHS_CMD[15:0] registers during data transfer has no effect.

This register can be used for any card.

**NOTE:** In SYSTEST mode, a write into MMCHS_CMD register will not start a transfer.

The command and transfer mode register (MMCHS_CMD) is shown in Figure 18-47 and described in Table 18-26.

**Figure 18-47. Command and Transfer Mode Register (MMCHS_CMD)**

| 31 | 30 | 29 | | | | | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | INDX | | | | | |
| R-0 | | R/W-0 | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| CMD_TYPE | | DP | CICE | CCCE | Reserved | RSP_TYPE | |
| R/W-0 | | R/W-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | MSBS | DDIR | Reserved | ACEN | BCE | DE |
| R-0 | | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-26. Command and Transfer Mode Register (MMCHS_CMD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 29-24 | INDX | | Command index binary encoded value from 0 to 63 specifying the command number send to card. |
| | | 0 | CMD0 or ACMD0 |
| | | 1h | CMD1 or ACMD1 |
| | | 2h | CMD2 or ACMD2 |
| | | 3h | CMD3 or ACMD3 |
| | | 4h | CMD4 or ACMD4 |
| | | 5h | CMD5 or ACMD5 |
| | | 6h | CMD6 or ACMD6 |
| | | 7h | CMD7 or ACMD7 |
| | | 8h | CMD8 or ACMD8 |
| | | 9h | CMD9 or ACMD9 |
| | | Ah | CMD10 or ACMD10 |
| | | Bh | CMD11 or ACMD11 |
| | | Ch | CMD12 or ACMD12 |
| | | Dh | CMD13 or ACMD13 |
| | | Eh | CMD14 or ACMD14 |
| | | Fh | CMD15 or ACMD15 |
| | | 10h | CMD16 or ACMD16 |
| | | 11h | CMD17 or ACMD17 |
| | | 12h | CMD18 or ACMD18 |
| | | 13h | CMD19 or ACMD19 |
| | | 14h | CMD20 or ACMD20 |
| | | 15h | CMD21 or ACMD21 |
| | | 16h | CMD22 or ACMD22 |
| | | 17h | CMD23 or ACMD23 |
| | | 18h | CMD24 or ACMD24 |
| | | 19h | CMD25 or ACMD25 |
| | | 1Ah | CMD26 or ACMD26 |
| | | 1Bh | CMD27 or ACMD27 |
| | | 1Ch | CMD28 or ACMD28 |
| | | 1Dh | CMD29 or ACMD29 |
| | | 1Eh | CMD30 or ACMD30 |
| | | 1Fh | CMD31 or ACMD31 |
| | | 20h | CMD32 or ACMD32 |
| | | 21h | CMD33 or ACMD33 |
| | | 22h | CMD34 or ACMD34 |
| | | 23h | CMD35 or ACMD35 |
| | | 24h | CMD36 or ACMD36 |
| | | 25h | CMD37 or ACMD37 |
| | | 26h | CMD38 or ACMD38 |
| | | 27h | CMD39 or ACMD39 |
| | | 28h | CMD40 or ACMD40 |
| | | 29h | CMD41 or ACMD41 |
| | | 2Ah | CMD42 or ACMD42 |
| | | 2Bh | CMD43 or ACMD43 |
| | | 2Ch | CMD44 or ACMD44 |
| | | 2Dh | CMD45 or ACMD45 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-26. Command and Transfer Mode Register (MMCHS_CMD) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| | | 2Eh | CMD46 or ACMD46 |
| | | 2Fh | CMD47 or ACMD47 |
| | | 30h | CMD48 or ACMD48 |
| | | 31h | CMD49 or ACMD49 |
| | | 32h | CMD50 or ACMD5 |
| | | 33h | CMD51 or ACMD5 |
| | | 34h | CMD52 or ACMD5 |
| | | 35h | CMD53 or ACMD5 |
| | | 36h | CMD54 or ACMD5 |
| | | 37h | CMD55 or ACMD5 |
| | | 38h | CMD56 or ACMD5 |
| | | 39h | CMD57 or ACMD5 |
| | | 3Ah | CMD58 or ACMD5 |
| | | 3Bh | CMD59 or ACMD5 |
| | | 3Ch | CMD60 or ACMD60 |
| | | 3Dh | CMD61 or ACMD61 |
| | | 3Eh | CMD62 or ACMD62 |
| | | 3Fh | CMD63 or ACMD63 |
| 23-22 | CMD_TYPE | | Command type. This register specifies three types of special commandSuspend, Resume and Abort. These bits shall be cleared to 0b00 for all other commands. |
| | | 0 | Others commands |
| | | 1h | Upon CMD52 "Bus Suspend" operation |
| | | 2h | Upon CMD52 "Function Select" operation |
| | | 3h | Upon CMD12 or CMD52 "I/O Abort" command |
| 21 | DP | | Data present select. This register indicates that data is present and mmc_dat line shall be used. It must be cleared to 0 in the following conditions:<br>• Command using only mmchs_cmd line<br>• Command with no data transfer but using busy signal on mmc_dat0<br>• Resume command |
| | | 0 | Command with no data transfer |
| | | 1 | Command with data transfer |
| 20 | CICE | | Command Index check enable. This bit must be set to 1 to enable index check on command response to compare the index field in the response against the index of the command. If the index is not the same in the response as in the command, it is reported as a command index error (MMCHS_STAT[19] CIE bit set to1) NoteThe CICE bit cannot be configured for an Auto CMD12, then index check is automatically checked when this command is issued. |
| | | 0 | Index check disable |
| | | 1 | Index check enable |
| 19 | CCCE | | Command CRC check enable. This bit must be set to 1 to enable CRC7 check on command response to protect the response against transmission errors on the bus. If an error is detected, it is reported as a command CRC error (MMCHS_STAT[17] CCRC bit set to 1). NoteThe CCCE bit cannot be configured for an Auto CMD12, and then CRC check is automatically checked when this command is issued. |
| | | 0 | CRC7 check disable |
| | | 1 | CRC7 check enable |
| 18 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 17-16 | RSP_TYPE | | Response type. This bits defines the response type of the command. |
| | | 0 | No response |
| | | 1h | Response Length 136 bits |
| | | 2h | Response Length 48 bits |
| | | 3h | Response Length 48 bits with busy after response |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-26. Command and Transfer Mode Register (MMCHS_CMD) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-6 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 5 | MSBS | | Multi/Single block select. This bit must be set to 1 for data transfer in case of multi block command. For any others command this bit shall be cleared to 0. |
| | | 0 | Single block. If this bit is 0, it is not necessary to set the register MMCHS_BLK[31:16] NBLK bits. |
| | | 1 | Multi block. When Block Count is disabled (MMCHS_CMD[1] BCE bit is cleared to 0) in Multiple block transfers (MMCHS_CMD[5] MSBS bit is set to 1), the module can perform infinite transfer. |
| 4 | DDIR | | Data transfer Direction. Select This bit defines either data transfer will be a read or a write. |
| | | 0 | Data Write (host to card) |
| | | 1 | Data Read (card to host) |
| 3 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 2 | ACEN | | Auto CMD12 Enable (SD cards only). When this bit is set to 1, the host controller issues a CMD12 automatically after the transfer completion of the last block. The Host Driver shall not set this bit to issue commands that do not require CMD12 to stop data transfer. In particular, secure commands do not require CMD12.<br>For CE-ATA commands (MMCHS_CON[12] CEATA bit set to 1), auto CMD12 is useless; therefore when this bit is set the mechanism to detect command completion signal, named CCS, interrupt is activated. |
| | | 0 | Auto CMD12 disable |
| | | 1 | Auto CMD12 enable or CCS detection enabled. |
| 1 | BCE | | Block Count Enable (Multiple block transfers only). This bit is used to enable the block count register (MMCHS_BLK[31:16] NBLK bits). When Block Count is disabled (MMCHS_CMD[1] BCE bit is cleared to 0) in Multiple block transfers (MMCHS_CMD[5] MSBS bits is set to 1), the module can perform infinite transfer. |
| | | 0 | Block count disabled for infinite transfer. |
| | | 1 | Block count enabled for multiple block transfer with known number of blocks |
| 0 | DE | | DMA Enable. This bit is used to enable DMA mode for host data access. |
| | | 0 | DMA mode disable |
| | | 1 | DMA mode enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.14 Command Response[31:0] Register (MMCHS_RSP10)

This 32-bit register holds bits positions [31:0] of command response type R1, R1b, R2, R3, R4, R5, R5b, or R6. The command response[31:0] register (MMCHS_RSP10) is shown in Figure 18-48 and described in Table 18-27.

**Figure 18-48. Command Response[31:0] Register (MMCHS_RSP10)**

| 31 16 | 15 0 |
|---|---|
| RSP1 | RSP0 |
| R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-27. Command Response[31:0] Register (MMCHS_RSP10) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | RSP1 | 0-FFFFh | Command Response [31:16] |
| 15-0 | RSP0 | 0-FFFFh | Command Response [15:0] |

### 18.4.15 Command Response[63:32] Register (MMCHS_RSP32)

This 32-bit register holds bits positions [63:32] of command response type R2. The command response[63:32] register (MMCHS_RSP32) is shown in Figure 18-49 and described in Figure 18-49.

**Figure 18-49. Command Response[63:32] Register (MMCHS_RSP32)**

| 31 16 | 15 0 |
|---|---|
| RSP3 | RSP2 |
| R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-28. Command Response[63:32] Register (MMCHS_RSP32) Field Descriptions**

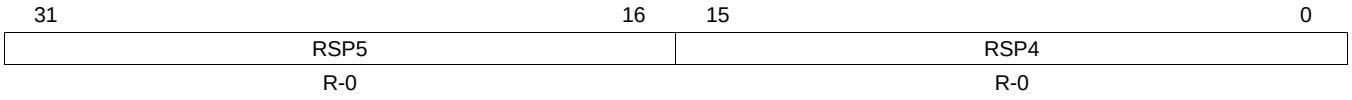
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | RSP3 | 0-FFFFh | Command Response [63:48] |
| 15-0 | RSP2 | 0-FFFFh | Command Response [47:32] |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.16 Command Response[95:64] Register (MMCHS_RSP54)

This 32-bit register holds bits positions [95:64] of command response type R2. The command response[95:64] register (MMCHS_RSP54) is shown in Figure 18-50 and described in Table 18-29.

**Figure 18-50. Command Response[95:64] Register (MMCHS_RSP54)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

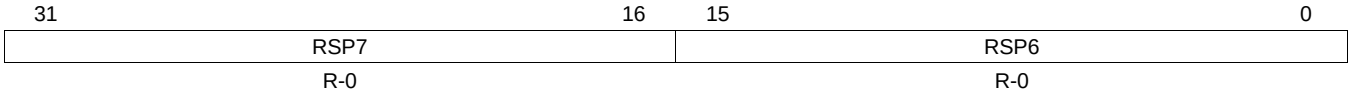
**Table 18-29. Command Response[95:64] Register (MMCHS_RSP54) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | RSP5 | 0-FFFFh | Command Response [95:80] |
| 15-0 | RSP4 | 0-FFFFh | Command Response [79:64] |

### 18.4.17 Command Response[127:96] Register (MMCHS_RSP76)

This 32-bit register holds bits positions [127:96] of command response type R2. The command response[127:96] register (MMCHS_RSP76) is shown in Figure 18-51 and described in Table 18-30.

**Figure 18-51. Command Response[127:96] Register (MMCHS_RSP76)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-30. Command Response[127:96] Register (MMCHS_RSP76) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | RSP7 | 0-FFFFh | Command Response [127:112] |
| 15-0 | RSP6 | 0-FFFFh | Command Response [111:96] |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.18 Data Register (MMCHS_DATA)

This register is the 32-bit entry point of the buffer for read or write data transfers. The buffer size is 32bitsx256(1024 bytes). Bytes within a word are stored and read in little endian format. This buffer can be used as two 512 byte buffers to transfer data efficiently without reducing the throughput.

Sequential and contiguous access is necessary to increment the pointer correctly. Random or skipped access is not allowed. In little endian, if the local host accesses this register byte-wise or 16bit-wise, the least significant byte (bits [7:0]) must always be written/read first. The update of the buffer address is done on the most significant byte write for full 32-bit DATA register or on the most significant byte of the last word of block transfer.

Example 1Byte or 16-bit access

- Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=1100 (2-bytes) OK
- Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=0100 (1-byte) OK
- Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=1000 (1-byte) Bad

The data register (MMCHS_DATA) is shown in Figure 18-52 and described in Table 18-31.

**Figure 18-52. Data Register (MMCHS_DATA)**

| 31 | 0 |
|---|---|
| DATA | |
| R/W-0 | |

LEGEND: R/W = Read/Write; *-n* = value after reset

**Table 18-31. Data Register (MMCHS_DATA) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | DATA | 0-FFFF FFFFh | Data register [31:0]. In functional mode (MMCHS_CON[4] MODE bit set to the default value 0):<br>A read access to this register is allowed only when the buffer read enable status is set to 1 (MMCHS_PSTATE[11] BRE bit), otherwise a bad access (MMCHS_STAT[29] BADA bit) is signaled.<br>A write access to this register is allowed only when the buffer write enable status is set to 1 (MMCHS_PSTATE[10] BWE bit), otherwise a bad access (MMCHS_STAT[29] BADA bit) is signaled and the data is not written. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.19 Present State Register (MMCHS_PSTATE)

The Host can get the status of the Host controller from this 32-bit read only register. The present state register (MMCHS_PSTATE) is shown in Figure 18-53 and described in Table 18-32.

**Figure 18-53. Present State Register (MMCHS_PSTATE)**

| 31 | 25 | 24 | 23 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | CLEV | DLEV | | WP | CDPL | CSS | CINS |
| R-0 | | R-x | R-x | | R-0 | R-0 | R-0 | R-0 |

| 15 | 12 | 11 | 10 | 9 | 8 | 7 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | BRE | BWE | RTA | WTA | Reserved | | DLA | DATI | CMDI |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 | | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 18-32. Present State Register (MMCHS_PSTATE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-25 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 24 | CLEV | | mmchs_cmd line signal level. This status is used to check the mmchs_cmd line level to recover from errors, and for debugging. The value of this register after reset depends on the mmchs_cmd line level at that time. |
| | | Read 0 | The mmchs_cmd line level is 0. |
| | | Read 1 | The mmchs_cmd line level is 1. |
| 23-20 | DLEV | | mmc_dat[3:0] line signal level<br>• mmc_dat3 => bit 23<br>• mmc_dat2 => bit 22<br>• mmc_dat1 => bit 21<br>• mmc_dat0 => bit 20<br>This status is used to check mmc_dat line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from mmc_dat0 . The value of these registers after reset depends on the mmc_dat lines level at that time. |
| 19 | WP | | Write Protect Status. MMC/SD/SDIO1 only. SDIO cards only. This bit reflects the write protect input pin (SDWP) level. The value of this register after reset depends one the protect input pin (SDWP) level at that time. |
| | | Read 0 | The card is write protected. |
| | | Read 1 | The card is not write protected. |
| 18 | CDPL | | Card Detect Pin Level. This bit reflects the inverse value of the card detect input pin (SDCD). Debouncing is not performed on this bit and is valid only when Card State is stable. (MMCHS_PSTATE[17] is set to 1). This bit must be debounced by software. The value of this register after reset depends on the card detect input pin (SDCD) level at that time. |
| | | Read 0 | The value of the card detect input pin (SDCD) is 1. |
| | | Read 1 | The value of the card detect input pin (SDCD) is 0. |
| 17 | CSS | | Card State Stable. This bit is used for testing. It is set to 1 only when Card Detect Pin Level is stable (MMCHS_PSTATE[18] CPDL). Debouncing is performed on the card detect input pin (SDCD) to detect card stability. This bit is not affected by software reset. |
| | | Read 0 | Reset or Debouncing. |
| | | Read 1 | No card or card inserted. |
| 16 | CINS | | Card inserted. This bit is the debounced value of the card detect input pin (SDCD). An inactive to active transition of the card detect input pin (SDCD) will generate a card insertion interrupt (MMCHS_STAT[CINS]). A active to inactive transition of the card detect input pin (SDCD) will generate a card removal interrupt (MMCHS_STAT[CREM]). This bit is not affected by a software reset. |
| | | Read 0 | If MMCHS_CON[CDP] is cleared to 0 (default), no card is detected. The card may have been removed from the card slot. If MMCHS_CON[CDP] is set to 1, the card has been inserted. |
| | | Read 1 | If MMCHS_CON[CDP] is cleared to 0 (default), the card has been inserted from the card slot. If MMCHS_CON[CDP] is set to 1, no card is detected. The card may have been removed from the card slot. |
| 15-12 | Reserved | 0 | Reserved bit field. Do not write any value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-32. Present State Register (MMCHS_PSTATE) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 11 | BRE | | Buffer read enable. This bit is used for non-DMA read transfers. It indicates that a complete block specified by MMCHS_BLK[10:0] BLEN bits has been written in the buffer and is ready to be read. It is cleared to 0 when the entire block is read from the buffer. It is set to 1 when a block data is ready in the buffer and generates the Buffer read ready status of interrupt (MMCHS_STAT[5] BRR bit). |
| | | Read 0 | Read BLEN bytes disable |
| | | Read 1 | Read BLEN bytes enable. Readable data exists in the buffer. |
| 10 | BWE | | Buffer Write enable. This status is used for non-DMA write transfers. It indicates if space is available for write data. |
| | | Read 0 | There is no room left in the buffer to write BLEN bytes of data. |
| | | Read 1 | There is enough space in the buffer to write BLEN bytes of data. |
| 9 | RTA | | Read transfer active. This status is used for detecting completion of a read transfer. It is set to 1 after the end bit of read command or by activating a continue request (MMCHS_HCTL[17] CR bit) following a stop at block gap request. This bit is cleared to 0 when all data have been read by the local host after last block or after a stop at block gap request. |
| | | Read 0 | No valid data on the mmc_dat lines. |
| | | Read 1 | Read data transfer on going. |
| 8 | WTA | | Write transfer active. This status indicates a write transfer active. It is set to 1 after the end bit of write command or by activating a continue request (MMCHS_HCTL[17] CR bit) following a stop at block gap request. This bit is cleared to 0 when CRC status has been received after last block or after a stop at block gap request. |
| | | Read 0 | No valid data on the mmc_dat lines. |
| | | Read 1 | Write data transfer on going. |
| 7-3 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 2 | DLA | | mmc_dat line active. This status bit indicates whether one of the mmc_dat lines is in use.<br>In the case of read transactions (card to host)This bit is set to 1 after the end bit of read command or by activating continue request MMCHS_HCTL[17] CR bit. This bit is cleared to 0 when the host controller received the end bit of the last data block or at the beginning of the read wait mode.<br>In the case of write transactions (host to card)This bit is set to 1 after the end bit of write command or by activating continue request MMCHS_HCTL[17] CR bit.<br>This bit is cleared to 0 on the end of busy event for the last block; host controller must wait 8 clock cycles with line not busy to really consider not "busy state" or after the busy block as a result of a stop at gap request. |
| | | Read 0 | mmc_dat line inactive |
| | | Read 1 | mmc_dat line active |
| 1 | DATI | | Command inhibit (mmc_dat). This status bit is generated if either mmc_dat line is active (MMCHS_PSTATE[2] DLA bit) or Read transfer is active (MMCHS_PSTATE[9] RTA bit) or when a command with busy is issued. This bit prevents the local host to issue a command.<br>A change of this bit from 1 to 0 generates a transfer complete interrupt (MMCHS_STAT[1] TC bit). |
| | | Read 0 | Issuing of command using the mmc_dat lines is allowed |
| | | Read 1 | Issuing of command using mmc_dat lines is not allowed |
| 0 | CMDI | | Command inhibit(mmchs_cmd). This status bit indicates that the mmchs_cmd line is in use. This bit is cleared to 0 when the most significant byte is written into the command register. This bit is not set when Auto CMD12 is transmitted. This bit is cleared to 0 in either the following cases:<br>• After the end bit of the command response, excepted if there is a command conflict error (MMCHS_STAT[17] CCRC bit or MMCHS_STAT[18] CEB bit set to 1) or a Auto CMD12 is not executed (MMCHS_AC12[0] ACNE bit).<br>• After the end bit of the command without response (MMCHS_CMD[17:16] RSP_TYPE bits set to "00"). In case of a command data error is detected (MMCHS_STAT[19] CTO bit set to 1), this register is not automatically cleared. |
| | | Read 0 | Issuing of command using mmchs_cmd line is allowed |
| | | Read 1 | Issuing of command using mmchs_cmd line is not allowed |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.20 Control Register (MMCHS_HCTL)

This register defines the host controls to set power, wake-up and transfer parameters.

- MMCHS_HCTL[31:24] = Wake-up control
- MMCHS_HCTL[23:16] = Block gap control
- MMCHS_HCTL[15:8] = Power control
- MMCHS_HCTL[7:0] = Host control

**NOTE:** If your device does not support MMC cards, then those bits in this register which are meant for MMC card use should be assumed to be reserved.

The control register (MMCHS_HCTL) is shown in Figure 18-54 and described in Table 18-33.

**Figure 18-54. Control Register (MMCHS_HCTL)**

| 31 | 28 | 27 | 26 | 25 | 24 | 23 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | OBWE | REM | INS | IWE | Reserved | | IBG | RWC | CR | SBGR |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 12 | 11 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | SDVS | | SDBP | CDSS | CDTL | Rsvd | DMAS | | HSPE | DTW | Rsvd |
| R-0 | | R/W-0 | | R/W-0 | R/W-0 | R/W-0 | R-0 | R/W-2h | | R/W-0 | R/W-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 18-33. Control Register (MMCHS_HCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 27 | OBWE | | Wake-up event enable for 'out-of-band' Interrupt. This bit enables wake-up events for 'out-of-band' assertion. Wake-up is generated if the wake-up feature is enabled (MMCHS_SYSCONFIG[2] ENAWAKEUP bit). The write to this register is ignored when MMCHS_CON[14] OBIE bit is not set. |
| | | 0 | Disable wake-up on 'out-of-band' Interrupt |
| | | 1 | Enable wake-up on 'out-of-band' Interrupt |
| 26 | REM | | Wake-up event enable on SD card removal. This bit enables wake-up events for card removal assertion. Wake-up is generated if the wake-up feature is enabled (MMCHS_SYSCONFIG[2] ENAWAKEUP bit). |
| | | 0 | Disable wake-up on card removal |
| | | 1 | Enable wake-up on card removal |
| 25 | INS | | Wake-up event enable on SD card insertion This bit enables wake-up events for card insertion assertion. Wake-up is generated if the wake-up feature is enabled (MMCHS_SYSCONFIG[2] ENAWAKEUP bit). |
| | | 0 | Disable wake-up on card insertion |
| | | 1 | Enable wake-up on card insertion |
| 24 | IWE | | Wake-up event enable on SD card interrupt. This bit enables wake-up events for card interrupt assertion. Wake-up is generated if the wake-up feature is enabled (MMCHS_SYSCONFIG[2] ENAWAKEUP bit) and enable status bit is set (MMCHS_IE[8] CIRQ_ENABLE bit). |
| | | 0 | Disable wake-up on card interrupt |
| | | 1 | Enable wake-up on card interrupt |
| 23-20 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 19 | IBG | | Interrupt block at gap. This bit is valid only in 4-bit mode of SDIO card to enable interrupt detection in the interrupt cycle at block gap for a multiple block transfer. For MMC cards and for SD card this bit should be cleared to 0. |
| | | 0 | Disable interrupt detection at the block gap in 4-bit mode |
| | | 1 | Enable interrupt detection at the block gap in 4-bit mode |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-33. Control Register (MMCHS_HCTL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 18 | RWC | | Read wait control. The read wait function is optional only for SDIO cards. If the card supports read wait, this bit must be enabled, then requesting a stop at block gap (MMCHS_HCTL[16] SBGR bit) generates a read wait period after the current end of block. Be careful, if read wait is not supported it may cause a conflict on mmc_dat line. |
| | | 0 | Disable read wait control. Suspend/resume cannot be supported. |
| | | 1 | Enable read wait control |
| 17 | CR | | Continue request. This bit is used to restart a transaction that was stopped by requesting a stop at block gap (MMCHS_HCTL[16] SBGR bit). Set this bit to 1 restarts the transfer. The bit is automatically cleared to 0 by the host controller when transfer has restarted, that is, mmc_dat line is active (MMCHS_PSTATE[2] DLA bit) or transferring data (MMCHS_PSTATE[8] WTA bit).<br>The Stop at block gap request must be disabled (MMCHS_HCTL[16] SBGR bit =0) before setting this bit. |
| | | 0 | No affect |
| | | 1 | Transfer restart |
| 16 | SBGR | | Stop at block gap request. This bit is used to stop executing a transaction at the next block gap. The transfer can restart with a continue request (MMCHS_HCTL[17] CR bit) or during a suspend/resume sequence. In case of read transfer, the card must support read wait control. In case of write transfer, the host driver shall set this bit after all block data written. Until the transfer completion (MMCHS_STAT[1] TC bit set to 1), the host driver shall leave this bit set to 1.If this bit is set, the local host shall not write to the data register (MMCHS_DATA). |
| | | 0 | Transfer mode |
| | | 1 | Stop at block gap |
| 15-12 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 11-9 | SDVS | | SD bus voltage select (All cards). The host driver should set these bits to select the voltage level for the card according to the voltage supported by the system (MMCHS_CAPA[26] VS18 bit, MMCHS_CAPA[25] VS30 bit, MMCHS_CAPA[24] VS33 bit) before starting a transfer. |
| | | | MMCHS2. This field must be set to 5h. |
| | | | MMCHS3. This field must be set to 5h. |
| | | 5h | 1.8 V (Typical) |
| | | 6h | 3.0 V (Typical) |
| | | 7h | 3.3 V (Typical) |
| 8 | SDBP | | SD bus power. Before setting this bit, the host driver shall select the SD bus voltage (MMCHS_HCTL[11:9] SDVS bits). If the host controller detects the No card state, this bit is automatically cleared to 0. If the module is power off, a write in the command register (MMCHS_CMD) will not start the transfer. A write to this bit has no effect if the selected SD bus voltage is not supported according to capability register (MMCHS_CAPA[VS*]). |
| | | 0 | Power off |
| | | 1 | Power on |
| 7 | CDSS | | Card Detect Signal Selection. This bit selects source for the card detection. When the source for the card detection is switched, the interrupt should be disabled during the switching period by clearing the Interrupt Status/Signal Enable register in order to mask unexpected interrupt being caused by the glitch. The Interrupt Status/Signal Enable should be disabled during over the period of debouncing. |
| | | 0 | SDCD# is selected (for normal use). |
| | | 1 | The Card Detect Test Level is selected (for test purposes). |
| 6 | CDTL | | Card Detect Test Level. This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not. |
| | | 0 | No card |
| | | 1 | Card inserted. |
| 5 | Reserved | 0 | Reserved bit field. Do not write any value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-33. Control Register (MMCHS_HCTL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 4-3 | DMAS | | DMA Select. One of the supported DMA modes can be selected. The host driver shall check support of DMA modes by referencing the Capabilities register. Use of selected DMA is determined by DMA Enable of the Transfer Mode register.<br>This register is only meaningful when MADMA_EN is set to 1. When MADMA_EN is cleared to 0 the bit field is read only and returned value is 0. |
| | | 0-1h | Reserved |
| | | 2h | 32-bit Address ADMA2 is selected. |
| | | 3h | Reserved |
| 2 | HSPE | | High Speed Enable. Before setting this bit, the Host Driver shall check the High Speed Support in the Capabilities register. If this bit is cleared to 0 (default), the Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock.<br>If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock. This bit shall not be set when dual data rate mode is activated in MMCHS_CON[DDR]. |
| | | 0 | Normal speed mode |
| | | 1 | High speed mode |
| 1 | DTW | | Data transfer width. This bit must be set following a valid SET_BUS_WIDTH command (ACMD6) with the value written in bit 1 of the argument. Prior to this command, the SD card configuration register (SCR) must be verified for the supported bus width by the SD card. |
| | | 0 | 1-bit Data width (mmc_dat0 used) |
| | | 1 | 4-bit Data width (mmc_dat[3:0] used) |
| 0 | Reserved | 0 | Reserved bit field. Do not write any value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.21 SD System Control Register (MMCHS_SYSCTL)

This register defines the system controls to set software resets, clock frequency management and data timeout.

- MMCHS_SYSCTL[31:24] = Software resets
- MMCHS_SYSCTL[23:16] = Timeout control
- MMCHS_SYSCTL[15:0] = Clock control

The SD system control register (MMCHS_SYSCTL) is shown in Figure 18-55 and described in Table 18-34.

**Figure 18-55. SD System Control Register (MMCHS_SYSCTL)**

| 31 | 27 | 26 | 25 | 24 | 23 | 20 | 19 | 16 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | SRD | SRC | SRA | Reserved | | DTO | |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R-0 | | R/W-0 | |

| 15 | 6 | 5 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| CLKD | | Reserved | | CEN | ICS | ICE |
| R/W-0 | | R-0 | | R/W-0 | R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-34. SD System Control Register (MMCHS_SYSCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-27 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 26 | SRD | | Software reset for mmc_dat line. This bit is set to 1 for reset and released to 0 when completed. mmc_dat finite state machine in both clock domain are also reset.<br>These registers are cleared by the MMCHS_SYSCTL[26] SRD bit:<br>• MMCHS_DATA<br>• MMCHS_PSTATEBRE, BWE, RTA, WTA, DLA and DATI<br>• MMCHS_HCTLSBGR and CR<br>• MMCHS_STATBRR, BWR, BGE and TC Interconnect and MMC buffer data management is reinitialized.<br>**Note**If a soft reset is issued when an interrupt is asserted, data may be lost. |
| | | 0 | Reset completed |
| | | 1 | Software reset for mmc_dat line |
| 25 | SRC | | Software reset for mmchs_cmd line. This bit is set to 1 for reset and released to 0 when completed. mmchs_cmd finite state machine in both clock domain are also reset.<br>These registers are cleared by the MMCHS_SYSCTL[25] SRC bit:<br>• MMCHS_PSTATECMDI<br>• MMCHS_STATCC Interconnect and MMC command status management is reinitialized.<br>**Note**If a soft reset is issued when an interrupt is asserted, data may be lost. |
| | | 0 | Reset completed |
| | | 1 | Software reset for mmchs_cmd line |
| 24 | SRA | | Software reset for all. This bit is set to 1 for reset , and released to 0 when completed. This reset affects the entire host controller except for the card detection circuit and capabilities registers. |
| | | 0 | Reset completed |
| | | 1 | Software reset for all the design |
| 23-20 | Reserved | 0 | Reserved bit field. Do not write any value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-34. SD System Control Register (MMCHS_SYSCTL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 19-16 | DTO | | Data timeout counter value and busy timeout. This value determines the interval by which mmc_dat lines timeouts are detected.<br>The host driver needs to set this bit field based on:<br>• The maximum read access time (NAC) (Refer to the SD Specification Part1 Physical Layer)<br>• The data read access time values (TAAC and NSAC) in the card specific data register (CSD) of the card<br>• The timeout clock base frequency (MMCHS_CAPA[5:0] TCF bits)<br>If the card does not respond within the specified number of cycles, a data timeout error occurs (MMCHS_STAT[20] DTO bit). The MMCHS_SYSCTL[19,16] DTO bit field is also used to check busy duration, to generate busy timeout for commands with busy response or for busy programming during a write command. Timeout on CRC status is generated if no CRC token is present after a block write. |
| | | 0 | TCF x 2^13 |
| | | 1h | TCF x 2^14 |
| | | Eh | TCF x 2^27 |
| | | Fh | Reserved |
| 15-6 | CLKD | | Clock frequency select. These bits define the ratio between a reference clock frequency (system dependant) and the output clock frequency on the mmc_clk pin of either the memory card (MMC, SD, or SDIO). |
| | | 0 | Clock Ref bypass |
| | | 1h | Clock Ref bypass |
| | | 2h | Clock Ref / 2 |
| | | 3h | Clock Ref / 3 |
| | | 3FFh | Clock Ref / 1023 |
| 5-3 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 2 | CEN | | Clock enable. This bit controls if the clock is provided to the card or not. |
| | | 0 | The clock is not provided to the card . Clock frequency can be changed . |
| | | 1 | The clock is provided to the card and can be automatically gated when MMCHS_SYSCONFIG[0] AUTOIDLE bit is set to 1 (default value) .<br>The host driver shall wait to set this bit to 1 until the Internal clock is stable (MMCHS_SYSCTL[1] ICS bit). |
| 1 | ICS | | Internal clock stable (status)This bit indicates either the internal clock is stable or not. |
| | | 0 | The internal clock is not stable. |
| | | 1 | The internal clock is stable after enabling the clock (MMCHS_SYSCTL[0] ICE bit) or after changing the clock ratio (MMCHS_SYSCTL[15:6] CLKD bits). |
| 0 | ICE | | Internal clock enable. This register controls the internal clock activity. In very low power state, the internal clock is stopped. NoteThe activity of the debounce clock (used for wake-up events) and the interface clock (used for reads and writes to the module register map) are not affected by this register. |
| | | 0 | The internal clock is stopped (very low power state). |
| | | 1 | The internal clock oscillates and can be automatically gated when MMCHS_SYSCONFIG[0] AUTOIDLE bit is set to 1 (default value). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.22 Interrupt Status Register (MMCHS_STAT)

The interrupt status regroups all the status of the module internal events that can generate an interrupt.

- MMCHS_STAT[31:16] = Error Interrupt Status
- MMCHS_STAT[15:0] = Normal Interrupt Status

The error bits are located in the upper 16 bits of the MMCHS_STAT register. All bits are cleared by writing a 1 to them. Additionally, bits 15 and 8 serve as special error bits. These cannot be cleared by writing a 1 to them. Bit 15 (ERRI) is automatically cleared when the error causing to ERRI to be set is handled. (that is, when bits 31:16 are cleared, bit 15 will be automatically cleared). Bit 8 (CIRQ) is cleared by writing a 0 to MMCHS_IE[8] (masking the interrupt) and servicing the interrupt.

The interrupt status register (MMCHS_STAT) is shown in Figure 18-56 and described in Table 18-35.

**Figure 18-56. Interrupt Status Register (MMCHS_STAT)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | BADA | CERR | Reserved | | ADMAE | ACE | Rsvd | DEB | DCRC | DTO | CIE | CEB | CCRC | CTO |
| R-0 | | R/W-0 | R/W-0 | R-0 | | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ERRI | Reserved | | BSR | OBI | CIRQ | CREM | CINS | BRR | BWR | DMA | BGE | TC | CC |
| R-0 | R-0 | | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-35. Interrupt Status Register (MMCHS_STAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Reserved bit field. Do not write any value |
| 29 | BADA | | Bad access to data space.<br>This bit is set automatically to indicate a bad access to buffer when not allowed:<br>• During a read access to the data register (MMCHS_DATA) while buffer reads are not allowed (MMCHS_PSTATE[11] BRE bit =0)<br>• During a write access to the data register (MMCHS_DATA) while buffer writes are not allowed (MMCHS_PSTATE[10] BWE bit=0) |
| | | Read 0 | No interrupt |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Bad access |
| | | Write 1 | Status is cleared. |
| 28 | CERR | | Card error. This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5 or R5b. Only bits referenced as type E (error) in status field in the response can set a card status error. An error bit in the response is flagged only if corresponding bit in card status response error MMCHS_CSRE in set.<br>There is no card error detection for autoCMD12 command. The host driver shall read MMCHS_RSP76 register to detect error bits in the command response. |
| | | Read 0 | No error |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Card error |
| | | Write 1 | Status is cleared. |
| 27-26 | Reserved | 0 | Reserved bit field. Do not write any value |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-35. Interrupt Status Register (MMCHS_STAT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 25 | ADMAE | | ADMA Error. This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA Error Status Register. In addition, the Host Controller generates this interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor. |
| | | Read 0 | No interrupt |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | ADMA error |
| | | Write 1 | Status is cleared. |
| 24 | ACE | | Auto CMD12 error. This bit is set automatically when one of the bits in Auto CMD12 Error status register has changed from 0 to 1. |
| | | Read 0 | No error |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | AutoCMD12 error |
| | | Write 1 | Status is cleared. |
| 23 | Reserved | 0 | Reserved bit field. Do not write any value |
| 22 | DEB | | Data End Bit error. This bit is set automatically when detecting a 0 at the end bit position of read data on mmc_dat line or at the end position of the CRC status in write mode. |
| | | Read 0 | No error |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Data end bit error |
| | | Write 1 | Status is cleared. |
| 21 | DCRC | | Data CRC Error. This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status different of a position "010" token during a block write command. |
| | | Read 0 | No error |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Data CRC error |
| | | Write 1 | Status is cleared. |
| 20 | DTO | | Data timeout error. This bit is set automatically according to the following conditions:<br>• Busy timeout for R1b, R5b response type<br>• Busy timeout after write CRC status<br>• Write CRC status timeout<br>• Read data timeout |
| | | Read 0 | No error |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Time out |
| | | Write 1 | Status is cleared. |
| 19 | CIE | | Command index error. This bit is set automatically when response index differs from corresponding command index previously emitted. It depends on the enable bit (MMCHS_CMD[20] CICE). |
| | | Read 0 | No error |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Command index error |
| | | Write 1 | Status is cleared. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-35. Interrupt Status Register (MMCHS_STAT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 18 | CEB | | Command end bit error. This bit is set automatically when detecting a 0 at the end bit position of a command response. |
| | | Read 0 | No error |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Command end bit error |
| | | Write 1 | Status is cleared. |
| 17 | CCRC | | Command CRC error. This bit is set automatically when there is a CRC7 error in the command response depending on the enable bit (MMCHS_CMD[19] CCCE). |
| | | Read 0 | No error |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Command CRC error |
| | | Write 1 | Status is cleared. |
| 16 | CTO | | Command timeout error. This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command.<br>For commands that reply within 5 clock cycles - the timeout is still detected at 64 clock cycles. |
| | | Read 0 | No error |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Time Out |
| | | Write 1 | Status is cleared. |
| 15 | ERRI | | Error interrupt. If any of the bits in the Error Interrupt Status register (MMCHS_STAT[31:16]) are set, then this bit is set to 1. Therefore the host driver can efficiently test for an error by checking this bit first.<br>Writes to this bit are ignored. |
| | | Read 0 | No interrupt |
| | | Read 1 | Error interrupt event(s) occurred |
| 14-11 | Reserved | 0 | Reserved bit field. Do not write any value |
| 10 | BSR | | Boot Status Received Interrupt. This bit is set automatically when MMCHS_CON[BOOT] is set 1 or 2 and a boot status is received on DAT[0] line. This interrupt is only useful for MMC card. |
| | | Read 0 | No interrupt |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Boot Status Received Interrupt occurred. |
| | | Write 1 | Status is cleared. |
| 9 | OBI | | Out-of-band interrupt (This interrupt is only useful for MMC card). This bit is set automatically when MMCHS_CON[14] OBIE bit is set and an out-of-band interrupt occurs on OBI pin.<br>The interrupt detection depends on polarity controlled by MMCHS_CON[13] OBIP bit.<br>The out-of-band interrupt signal is a system specific feature for future use, this signal is not required for existing specification implementation. |
| | | Read 0 | No out-of-band interrupt |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Interrupt out-of-band occurs |
| | | Write 1 | Status is cleared. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-35. Interrupt Status Register (MMCHS_STAT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 8 | CIRQ | | Card interrupt. This bit is only used for SD and SDIO cards.<br>In 1-bit mode, interrupt source is asynchronous (can be a source of asynchronous wake-up).<br>In 4-bit mode, interrupt source is sampled during the interrupt cycle.<br>In CE-ATA mode, interrupt source is detected when the card drives mmchs_cmd line to zero during one cycle after data transmission end.<br>All modes above are fully exclusive.<br>The controller interrupt must be clear by setting MMCHS_IE[8] CIRQ_ENABLE to 0, then the host driver must start the interrupt service with card (clearing card interrupt status) to remove card interrupt source. Otherwise the Controller interrupt will be reasserted as soon as MMCHS_IE[8] CIRQ_ENABLE is set to 1.<br>Writes to this bit are ignored. |
| | | Read 0 | No card interrupt |
| | | Read 1 | Generate card interrupt |
| 7 | CREM | | Card Removal. This bit is set automatically when MMCHS_PSTATE[CINS] changes from 1 to 0. A clear of this bit doesn't affect Card inserted present state (MMCHS_PSTATE[CINS]). |
| | | Read 0 | Card State stable or debouncing |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Card Removed |
| | | Write 1 | Status is cleared |
| 6 | CINS | | Card Insertion. This bit is set automatically when MMCHS_PSTATE[CINS] changes from 0 to 1. A clear of this bit doesn't affect Card inserted present state (MMCHS_PSTATE[CINS]). |
| | | Read 0 | Card State stable or debouncing |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Card inserted |
| | | Write 1 | Status is cleared. |
| 5 | BRR | | Buffer read ready. This bit is set automatically during a read operation to the card (see class 2 - block oriented read commands) when one block specified by the MMCHS_BLK[10:0] BLEN bit field is completely written in the buffer. It indicates that the memory card has filled out the buffer and that the local host needs to empty the buffer by reading it.<br>NoteIf the DMA receive-mode is enabled, this bit is never set; instead a DMA receive request to the main DMA controller of the system is generated. |
| | | Read 0 | Not ready to read buffer |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Ready to read buffer |
| | | Write 1 | Status is cleared. |
| 4 | BWR | | Buffer write ready. This bit is set automatically during a write operation to the card (see class 4 - block oriented write command) when the host can write a complete block as specified by MMCHS_BLK[10:0] BLEN. It indicates that the memory card has emptied one block from the buffer and that the local host is able to write one block of data into the buffer.<br>NoteIf the DMA transmit mode is enabled, this bit is never set; instead, a DMA transmit request to the main DMA controller of the system is generated. |
| | | Read 0 | Not ready to write buffer |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Ready to write buffer |
| | | Write 1 | Status is cleared. |
| 3 | DMA | | DMA Interrupt. This status is set when an interrupt is required in the ADMA instruction and after the data transfer completion. |
| | | Read 0 | DMA Interrupt detected |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | No DMA Interrupt |
| | | Write 1 | Status is cleared. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-35. Interrupt Status Register (MMCHS_STAT) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 2 | BGE | | Block gap event. When a stop at block gap is requested (MMCHS_HCTL[16] SBGR bit), this bit is automatically set when transaction is stopped at the block gap during a read or write operation.<br>This event does not occur when the stop at block gap is requested on the last block.<br>In read mode, a 1-to-0 transition of the mmc_dat line active status (MMCHS_PSTATE[2] DLA bit) between data blocks generates a Block gap event interrupt. |
| | | Read 0 | No block gap event |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Transaction stopped at block gap |
| | | Write 1 | Status is cleared |
| 1 | TC | | Transfer completed. This bit is always set when a read/write transfer is completed or between two blocks when the transfer is stopped due to a stop at block gap request (MMCHS_HCTL[16] SBGR bit).<br>This bit is also set when exiting a command in a busy state (if the command has a busy notification capability).<br>In Read modeThis bit is automatically set on completion of a read transfer (MMCHS_PSTATE[9] RTA bit).<br>In write modeThis bit is set automatically on completion of the mmc_dat line use (MMCHS_PSTATE[2] DLA bit). |
| | | Read 0 | No transfer complete |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Data transfer complete |
| | | Write 1 | Status is cleared |
| 0 | CC | | Command complete. This bit is set when a 1-to-0 transition occurs in the register command inhibit (MMCHS_PSTATE[0] CMDI bit)<br>If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command. A command timeout error (MMCHS_STAT[16] CTO bit) has higher priority than command complete (MMCHS_STAT[0] CC bit).<br>If a response is expected but none is received, then a command timeout error is detected and signaled instead of the command complete interrupt. |
| | | Read 0 | No command complete |
| | | Write 0 | Status bit unchanged |
| | | Read 1 | Command complete |
| | | Write 1 | Status is cleared |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.23 Interrupt SD Enable Register (MMCHS_IE)

This register allows to enable/disable the module to set status bits, on an event-by-event basis.

- MMCHS_IE[31:16] = Error Interrupt Status Enable
- MMCHS_IE[15:0] = Normal Interrupt Status Enable

The interrupt SD enable register (MMCHS_IE) is shown in Figure 18-57 and described in Table 18-36.

**Figure 18-57. Interrupt SD Enable Register (MMCHS_IE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | BADA_ENABLE | CERR_ENABLE | Reserved | | ADMA_ENABLE | ACE_ENABLE |
| R-0 | | R/W-0 | R/W-0 | R-0 | | R/W-0 | R/W-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | DEB_ENABLE | DCRC_ENABLE | DTO_ENABLE | CIE_ENABLE | CEB_ENABLE | CCRC_ENABLE | CTO_ENABLE |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 15 | 14 | | | 11 | 10 | 9 | 8 |
| NULL | Reserved | | | | BSR_ENABLE | OBI_ENABLE | CIRQ_ENABLE |
| R-0 | R-0 | | | | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CREM_ENABLE | CINS_ENABLE | BRR_ENABLE | BWR_ENABLE | DMA_ENABLE | BGE_ENABLE | TC_ENABLE | CC_ENABLE |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-36. Interrupt SD Enable Register (MMCHS_IE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 29 | BADA_ENABLE | | Bad access to data space interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 28 | CERR_ENABLE | | Card error interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 27-26 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 25 | ADMA_ENABLE | | ADMA error Interrupt Enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 24 | ACE_ENABLE | | Auto CMD12 error interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 23 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 22 | DEB_ENABLE | | Data end bit error interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 21 | DCRC_ENABLE | | Data CRC error interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 20 | DTO_ENABLE | | Data timeout error interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-36. Interrupt SD Enable Register (MMCHS_IE) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 19 | CIE_ENABLE | | Command index error interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 18 | CEB_ENABLE | | Command end bit error interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 17 | CCRC_ENABLE | | Command CRC error interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 16 | CTO_ENABLE | | Command timeout error interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 15 | NULL | 0 | Fixed to 0. The host driver shall control error interrupts using the Error Interrupt Signal Enable register. Writes to this bit are ignored. |
| 14-11 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 10 | BSR_ENABLE | | Boot Status Interrupt Enable A write to this register when MMCHS_CON[BOOT] is cleared to 0 is ignored. |
| | | 0 | Masked |
| | | 1 | Enabled |
| 9 | OBI_ENABLE | | Out-of-band interrupt enable A write to this register when MMCHS_CON[14] OBIE is cleared to 0 is ignored. |
| | | 0 | Masked |
| | | 1 | Enabled |
| 8 | CIRQ_ENABLE | | Card interrupt enable. A clear of this bit also clears the corresponding status bit. During 1-bit mode, if the interrupt routine does not remove the source of a card interrupt in the SDIO card, the status bit is reasserted when this bit is set to 1. This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. |
| | | 0 | Masked |
| | | 1 | Enabled |
| 7 | CREM_ENABLE | | Card Removal interrupt Enable This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. |
| | | 0 | Masked |
| | | 1 | Enabled |
| 6 | CINS_ENABLE | | Card Insertion interrupt Enable This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. |
| | | 0 | Masked |
| | | 1 | Enabled |
| 5 | BRR_ENABLE | | Buffer read ready interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 4 | BWR_ENABLE | | Buffer write ready interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 3 | DMA_ENABLE | | DMA interrupt enable |
| | | 0 | Masked |
| | | 1 | Enable |
| 2 | BGE_ENABLE | | Block gap event interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-36. Interrupt SD Enable Register (MMCHS_IE) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | TC_ENABLE | | Transfer completed interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 0 | CC_ENABLE | | Command completed interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.24 Interrupt Signal Enable Register (MMCHS_ISE)

This register allows you to enable/disable the module to set status bits, on an event-by-event basis.

- MMCHS_ISE[31:16] = Error Interrupt Signal Enable
- MMCHS_ISE[15:0] = Normal Interrupt Signal Enable

The Interrupt Signal Enable Register (MMCHS_ISE) is shown in Figure 18-58 and described in Table 18-37.

**Figure 18-58. Interrupt Signal Enable Register (MMCHS_ISE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | BADA_SIGEN | CERR_SIGEN | Reserved | | ADMA_SIGEN | ACE_SIGEN |
| R-0 | | R/W-0 | R/W-0 | R-0 | | R/W-0 | R/W-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | DEB_SIGEN | DCRC_SIGEN | DTO_SIGEN | CIE_SIGEN | CEB_SIGEN | CCRC_SIGEN | CTO_SIGEN |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 15 | 14 | | | 11 | 10 | 9 | 8 |
| NULL | Reserved | | | | BSR_SIGEN | OBI_SIGEN | CIRQ_SIGEN |
| R-0 | R-0 | | | | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CREM_SIGEN | CINS_SIGEN | BRR_SIGEN | BWR_SIGEN | DMA_SIGEN | BGE_SIGEN | TC_SIGEN | CC_SIGEN |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-37. Interrupt Signal Enable Register (MMCHS_ISE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 29 | BADA_SIGEN | | Bad access to data space interrupt enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 28 | CERR_SIGEN | | Card error interrupt signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 27-26 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 25 | ADMA_SIGEN | | ADMA error signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 24 | ACE_SIGEN | | Auto CMD12 error signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 23 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 22 | DEB_SIGEN | | Data end bit error signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 21 | DCRC_SIGEN | | Data CRC error signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 20 | DTO_SIGEN | | Data timeout error signal status enable |
| | | 0 | Masked<br>The host controller provides the clock to the card until the card sends the data or the transfer is aborted. |
| | | 1 | Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-37. Interrupt Signal Enable Register (MMCHS_ISE) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 19 | CIE_SIGEN | | Command index error signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 18 | CEB_SIGEN | | Command end bit error signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 17 | CCRC_SIGEN | | Command CRC error signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 16 | CTO_SIGEN | | Command timeout error signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 15 | NULL | 0 | Fixed to 0. The host driver shall control error interrupts using the error interrupt signal enable register. Writes to this bit are ignored. |
| 14-11 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 10 | BSR_SIGEN | | Boot Status signal status enable. A write to this register when MMCHS_CON[BOOT] is cleared to 0 is ignored |
| | | 0 | Masked |
| | | 1 | Enabled |
| 9 | OBI_SIGEN | | Out-of-band interrupt signal status enable. A write to this register when MMCHS_CON[14] OBIE is cleared to 0 is ignored. |
| | | 0 | Masked |
| | | 1 | Enabled |
| 8 | CIRQ_SIGEN | | Card interrupt signal status enable. A clear of this bit also clears the corresponding status bit.<br>During 1-bit mode, if the interrupt routine does not remove the source of a card interrupt in the SDIO card, the status bit is reasserted when this bit is set to 1.<br>This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. |
| | | 0 | Masked |
| | | 1 | Enabled |
| 7 | CREM_SIGEN | | Card Removal signal status enable This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. |
| | | 0 | Masked |
| | | 1 | Enabled |
| 6 | CINS_SIGEN | | Card Insertion signal status enable. This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source. |
| | | 0 | Masked |
| | | 1 | Enabled |
| 5 | BRR_SIGEN | | Buffer read ready signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 4 | BWR_SIGEN | | Buffer write ready signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 3 | DMA_SIGEN | | DMA signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 2 | BGE_SIGEN | | Block gap event signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-37. Interrupt Signal Enable Register (MMCHS_ISE) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | TC_SIGEN | | Transfer completed signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |
| 0 | CC_SIGEN | | Command completed signal status enable |
| | | 0 | Masked |
| | | 1 | Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.25 Auto CMD12 Error Status Register (MMCHS_AC12)

The host driver may determine which of the errors cases related to Auto CMD12 has occurred by checking this MMCHS_AC12 register when an auto CMD12 error interrupt occurs.

This register is valid only when auto CMD12 is enabled (MMCHS_CMD[2] ACEN bit) and auto CMD12Error (MMCHS_STAT[24] ACE bit) is set to 1.

**NOTE:** These bits are automatically reset when starting a new adtc command with data.

The auto CMD12 error status register (MMCHS_AC12) is shown in Figure 18-59 and described in Table 18-38.

**Figure 18-59. Auto CMD12 Error Status Register (MMCHS_AC12)**

| 31 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | CNI | Reserved | | ACIE | ACEB | ACCE | ACTO | ACNE |
| R-0 | | R-0 | R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-38. Auto CMD12 Error Status Register (MMCHS_AC12) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 7 | CNI | | Command not issue by auto CMD12 error. If this bit is set to 1, it means that pending command is not executed due to auto CMD12 error ACEB, ACCE, ACTO, or ACNE. |
| | | Read 0 | Not error |
| | | Read 1 | Command not issued |
| 6-5 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 4 | ACIE | | Auto CMD12 index error. This bit is a set to 1 when response index differs from corresponding command auto CMD12 index previously emitted.<br>This bit depends on the command index check enable (MMCHS_CMD[20] CICE bit). |
| | | Read 0 | No error |
| | | Read 1 | Auto CMD12 index error |
| 3 | ACEB | | Auto CMD12 end bit error. This bit is set to 1 when detecting a 0 at the end bit position of auto CMD12 command response. |
| | | Read 0 | No error |
| | | Read 1 | AutoCMD12 end bit error |
| 2 | ACCE | | Auto CMD12 CRC error. This bit is automatically set to 1 when a CRC7 error is detected in the auto CMD12 command response depending on the enable in the MMCHS_CMD[19] CCCE bit. |
| | | Read 0 | No error |
| | | Read 1 | Auto CMD12 CRC error |
| 1 | ACTO | | Auto CMD12 timeout error. This bit is set to 1 if no response is received within 64 clock cycles from the end bit of the auto CMD12 command. |
| | | Read 0 | No error |
| | | Read 1 | Auto CMD12 time out |
| 0 | ACNE | | Auto CMD12 not executed. This bit is set to 1 if multiple block data transfer command has started and if an error occurs in command before auto CMD12 starts. |
| | | Read 0 | Auto CMD12 executed |
| | | Read 1 | Auto CMD12 not executed |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.26 Capabilities Register (MMCHS_CAPA)

This register lists the capabilities of the MMC/SD/SDIO host controller. The capabilities register (MMCHS_CAPA) is shown in Figure 18-60 and described in Table 18-39.

**Figure 18-60. Capabilities Register (MMCHS_CAPA)**

| 31 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | 64BIT | Rsvd | VS18 | VS30 | VS33 | SRS | DS | HSS | Rsvd | AD2S | Rsvd | MBL | |
| R-0 | | R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R-1 | R-1 | R-1 | R-0 | R-1 | R-0 | R-1h | |

| 15 | 14 | 13 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | BCF | | TCU | Rsvd | TCF | |
| R-0 | | R-0 | | R-1 | R-0 | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-39. Capabilities Register (MMCHS_CAPA) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-29 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 28 | 64BIT | | 64 Bit System Bus Support. Setting 1 to this bit indicates that the Host Controller supports 64-bit address descriptor mode and is connected to 64-bit address system bus. |
| | | Read 0 | 32 bit System bus address |
| | | Read 1 | 64 bit System bus address |
| 27 | Reserved | 0 | Reserved bit. Any write to this bit results in 0. |
| 26 | VS18 | | Voltage support 1.8 V. Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via mmc_RESET signal).<br>Note. The device supports both 1.8v and 3.3v cards. However, MMC/SD/SDIO0 and MMC/SD/SDIO1 must be configured to the same voltage. You cannot configure MMC/SD/SDIO0, for example, to be 1.8v and MMC/SD/SDIO1 to be 3.3v. MMC/SD/SDIO2 is independent of the other two and can be set to either voltage regardless of what the other two are set to. |
| | | Read 0 | 1.8 V not supported |
| | | Write 0 | 1.8 V not supported |
| | | Read 1 | 1.8 V supported |
| | | Write 1 | 1.8 V supported |
| 25 | VS30 | | Voltage support 3.0V. Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via mmc_RESET signal)<br>Note. The device supports both 1.8v and 3.3v cards, however MMC/SD/SDIO0 and MMC/SD/SDIO1 must be configured to the same voltage. You cannot configure MMC/SD/SDIO0, for example, to be 1.8v and MMC/SD/SDIO1 to be 3.3v. MMC/SD/SDIO2 is independent of the other two and can be set to either voltage regardless of what the other two are set to. |
| | | Read 0 | 3.0 V not supported |
| | | Write 0 | 3.0 V not supported |
| | | Read 1 | 3.0 V supported |
| | | Write 1 | 3.0 V supported |
| 24 | VS33 | | Voltage support 3.3V. Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via mmc_RESET signal) |
| | | Read 0 | 3.3 V not supported |
| | | Write 0 | 3.3 V not supported |
| | | Read 1 | 3.3 V supported |
| | | Write 1 | 3.3 V supported |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-39. Capabilities Register (MMCHS_CAPA) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 23 | SRS | | Suspend/resume support (SDIO cards only). This bit indicates whether the host controller supports Suspend/Resume functionality. |
| | | Read 0 | The Host controller does not suspend/resume functionality. |
| | | Read 1 | The Host controller supports suspend/resume functionality. |
| 22 | DS | | DMA support. This bit indicates that the Host controller is able to use DMA to transfer data between system memory and the Host controller directly. |
| | | Read 0 | DMA not supported |
| | | Read 1 | DMA supported |
| 21 | HSS | | High-speed support. This bit indicates that the host controller supports high speed operations and can supply an up-to-52 MHz clock to the card. |
| | | Read 0 | High-speed not supported |
| | | Read 1 | High-speed supported |
| 20 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 19 | AD2S | | This bit indicates whether the Host Controller is capable of using ADMA2. It depends on setting of generic parameter MADMA_EN. |
| | | Read 0 | ADMA2 not supported |
| | | Read 1 | ADMA2 supported |
| 18 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 17-16 | MBL | | Maximum block length. This value indicates the maximum block size that the host driver can read and write to the buffer in the host controller. |
| | | | The host controller supports 512 bytes and 1024 bytes block transfers. |
| | | Read 0 | 512 bytes |
| | | Read 1 | 1024 bytes |
| | | Read 2 | 2048 bytes |
| 15-14 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 13-8 | BCF | | Base clock frequency for clock provided to the card. |
| | | Read 0 | The value indicating the base (maximum) frequency for the output clock provided to the card is system dependent and is not available in this register. Get the information via another method. See the *Power, Reset, and Clock Management* chapter for more information on the value of FUNC_96M_CLK clock signal. |
| 7 | TCU | | Timeout clock unit. This bit shows the unit of base clock frequency used to detect Data Timeout Error (MMCHS_STAT[20] DTO bit). |
| | | Read 0 | kHz |
| | | Read 1 | MHz |
| 6 | Reserved | 0 | Reserved. This bit is initialized to zero, and writes to it are ignored. |
| 5-0 | TCF | | Timeout clock frequency. The timeout clock frequency is used to detect Data Timeout Error (MMCHS_STAT[20] DTO bit). |
| | | Read 0 | The timeout clock frequency depends on the frequency of the clock provided to the card. The value of the timeout clock frequency is not available in this register. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.27 Force Event Register for Error Interrupt Status (MMCHS_FE)

The Force Event register is not a physically implemented register. Rather, it is an address at which the Error Interrupt Status register can be written. The effect of a write to this address will be reflected in the Error Interrupt Status Register, if corresponding bit of the Error Interrupt Status Enable Register is set.

The force event register (MMCHS_FE) is shown in Figure 18-61 and described in Table 18-40.

**Figure 18-61. Interrupt Signal Enable Register (MMCHS_FE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | FE_BADA | FE_CERR | Reserved | | FE_ADMAE | FE_ACE |
| R-0 | | W-0 | W-0 | R-0 | | W-0 | W-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | FE_DEB | FE_DCRC | FE_DTO | FE_CIE | FE_CEB | FE_CCRC | FE_CTO |
| R-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 | W-0 |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FE_CNI | Reserved | | FE_ACIE | FE_ACEB | FE_ACCE | FE_ACTO | FE_ACNE |
| W-0 | R-0 | | W-0 | W-0 | W-0 | W-0 | W-0 |

LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 18-40. Force Event Register (MMCHS_FE) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Reserved bit field. Any writes to these bits result in 0. |
| 29 | FE_BADA | | Force Event Bad access to data space. |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 28 | FE_CERR | | Force Event Card error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 27-26 | Reserved | 0 | Reserved bit field. Any writes to these bits result in 0. |
| 25 | FE_ADMAE | | Force Event ADMA error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 24 | FE_ACE | | Force Event Auto CMD12 error. |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 23 | Reserved | 0 | Reserved bit field. Any writes to this bit result in 0. |
| 22 | FE_DEB | | Force Event Data End Bit error. |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 21 | FE_DCRC | | Force Event Data CRC error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 20 | FE_DTO | | Force Event Data timeout error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 18-40. Force Event Register (MMCHS_FE) Field Descriptions (continued)**

| Bits | Field | Value | Description |
|---|---|---|---|
| 19 | FE_CIE | | Force Event Command index error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 18 | FE_CEB | | Force Event Command end bit error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 17 | FE_CCRC | | Force Event Command CRC error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 16 | FE_CTO | | Force Event Command Timeout error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 15-8 | Reserved | 0 | Reserved bit field. Any writes to these bits result in 0. |
| 7 | FE_CNI | | Force Event Command not issue by Auto CMD12 error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 6-5 | Reserved | 0 | Reserved bit field. Any writes to these bits result in 0. |
| 4 | FE_ACIE | | Force Event Auto CMD12 index error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 3 | FE_ACEB | | Force Event Auto CMD12 end bit error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 2 | FE_ACCE | | Force Event Auto CMD12 CRC error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 1 | FE_ACTO | | Force Event Auto CMD12 timeout error |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |
| 0 | FE_ACNE | | Force Event Auto CMD12 not executed. |
| | | Write 0 | No effect; no interrupt |
| | | Write 1 | Interrupt forced. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.28 ADMA Error Status Register (MMCHS_ADMAES)

When an ADMA Error Interrupt has occurred, the ADMA Error States field in this register holds the ADMA state and the ADMA System Address Register holds the address around the error descriptor. For recovering the error, the Host Driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address.
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address.
- ST_CADR: This sate is never set because do not generate ADMA error in this state.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address.

In the case of a write operation, the Host Driver should use ACMD22 to get the number of written block rather than using this information, since unwritten data may exist in the Host Controller. The Host Controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid = 0) at the ST_FDS state. In this case, ADMA Error State indicates that an error occurs at ST_FDS state. The Host Driver may find that the Valid bit is not set in the error descriptor.

The ADMA error status register (MMCHS_BLK) is shown in Figure 18-62 and described in Table 18-41.

**Figure 18-62. ADMA Error Status Register (MMCHS_ADMAES)**

| 31 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | LME | AES | |
| R-0 | | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 18-41. ADMA Error Status Register (MMCHS_ADMAES) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 2 | LME | | ADMA Length Mismatch Error:<br>• While Block Count Enable is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length.<br>• Total data length cannot be divided by the block length. |
| | | 0 | No error |
| | | 1 | Error |
| 1-0 | AES | | ADMA Error State. This field indicates the state of ADMA when an error occurred during an ADMA data transfer. This field never indicates "10" because ADMA never stops in this state. |
| | | Read 0 | ST_STOP (Stop DMA). Contents of the SYS_SDR register |
| | | Write 1h | ST_STOP (Stop DMA). Points to the error descriptor. |
| | | Read 2h | Never set this state. (Not used) |
| | | Write 3h | ST_TFR (Transfer Data). Points to the 'next' of the error descriptor. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.29 ADMA System Address Low Bits Register (MMCHS_ADMASAL)

This register holds the byte address of the executing command of the Descriptor table. The 32-bit Address Descriptor uses the lower 32 bits of this register. At the start of ADMA, the Host Driver shall set the start address of the Descriptor table.

The ADMA System address low bits register (MMCHS_ADMASAL) is shown in Figure 18-63 and described in Table 18-42.

**Figure 18-63. ADMA System Address Low Bits (MMCHS_ADMASAL)**

| 31 | 0 |
|---|---|
| ADMA_A32B | |
| R/W-0 | |

LEGEND: R/W = Read/Write; *-n* = value after reset

**Table 18-42. ADMA System Address Low Bits (MMCHS_ADMASAL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ADMA_A32B | 0-FFFF FFFFh | The ADMA increments this register address, which points to the next line, whenever fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register holds the valid Descriptor address depending on the ADMA state. The Host Driver shall program the Descriptor Table on a 32-bit boundary and set the 32-bit boundary address to this register. ADMA2 ignores the lower 2 bits of this register and assumes it to be 00b. |

### 18.4.30 ADMA System Address High Bits Register (MMCHS_ADMASAH)

The ADMA System address high bits register (MMCHS_ADMASAH) is shown in Figure 18-63 and described in Table 18-42.

**Figure 18-64. ADMA System Address High Bits Register (MMCHS_ADMASAH)**

| 31 | 0 |
|---|---|
| ADMA_A32B | |
| R/W-0 | |

LEGEND: R/W = Read/Write; *-n* = value after reset

**Table 18-43. ADMA System Address High Bits Register (MMCHS_ADMASAH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ADMA_A32B | 0 | This register is only used on devices with greater than 32-b address space. This register is not used on this device. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 18.4.31 Versions Register (MMCHS_REV)

This register contains the hard coded RTL vendor revision number, the version number of SD specification compliancy and a slot status bit.

- MMCHS_REV[31:16] = Host Controller Version
- MMCHS_REV[15:0] = Slot Interrupt Status

The versions register (MMCHS_REV) is shown in Figure 18-65 and described in Table 18-44.

**Figure 18-65. Versions Register (MMCHS_REV)**

| 31 24 | 23 16 | 15 1 | 0 |
|---|---|---|---|
| VREV | SREV | Reserved | SIS |
| R-x | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 18-44. Versions Register (MMCHS_REV) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | VREV | 0 | Vendor Version Number. Bits [7:4] is the major revision, bits [3:0] is the minor revision. Examples: 10h for 1.0 21h for 2.1 |
| 23-16 | SREV | Read 0 | Specification Version Number. This status indicates the Standard SD Host Controller Specification Version. The upper and lower 4-bits indicate the version. SD Host Specification Version 1.0 |
| 15-1 | Reserved | 0 | Reserved bit field. Do not write any value. |
| 0 | SIS | 0-1 | Slot Interrupt Status. This status bit indicates the inverted state of interrupt signal for the module. By a power on reset or by setting a software reset for all (MMCHS_SYSCTL[24] SRA), the interrupt signal shall be deasserted and this status shall read 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

# Peripheral Component Interconnect Express (PCIe) Module

This chapter describes the Peripheral Component Interconnect Express (PCIe) module. The PCIe provides a high speed glueless serial interconnect to peripherals utilizing high bandwidth applications.

**Topic** **Page**

**19.1 Introduction / Feature Overview** ..... **3255**
**19.2 Peripheral Architecture** ..... **3259**
**19.3 Supported Use Case** ..... **3277**
**19.4 Encoding of LTSSM State in DEBUG registers** ..... **3281**
**19.5 PCIESS Memory Map** ..... **3281**



Copyright © 2011–2016, Texas Instruments Incorporated

## 19.1 Introduction / Feature Overview

The Peripheral Component Interconnect Express (PCIe) module is a single-lane I/O interconnect that provides low pin-count, high reliability, and high-speed data transfer at line rates of up to 5.0 Gbps per lane per direction, for serial links on backplanes and printed wiring boards. It is a 3rd Generation I/O interconnect technology succeeding the PCI and ISA bus, that is designed to be used as a general-purpose serial I/O interconnect in multiple market segments, including desktop, mobile, server, storage and embedded communications. It is also used as a bridge to other interconnects like SATA, USB2/3.0, GbE MAC, and more.

The PCIESS contains the Synopsys DesignWare Core (DWC) PCIe Dual Mode core and Texas Instruments SERDES PHY. The dual mode (DM) core operates as either endpoint (EP) mode or root complex (RC) port mode. The core supports a single in-port and a single out-port. The operating mode of the device; that is, the role that the PCIe core assumes, is set to either EP or RC based on the value sampled from the BOOTMODE[4:0] pins and can be changed by a secondary bootloader or by application software configuring PCIE_CFG[PCIE_DEVTYPE] register. The DM core can be switched between modes at runtime by applying a power-on reset.

### 19.1.1 Purpose of the Peripheral

The incumbent design, PCI, is a parallel bus architecture that is increasingly difficult to scale-up in bandwidth, which is usually performed by increasing the number of data signal lines. More signal lines result in difficult clock-to-data skew management, creating complex PCB layout rules that make cost-effective implementations in the FR4 technology difficult. Increasing the number of signal lines also increases the power dissipation. The PCIe architecture was developed to help minimize I/O bus bottlenecks within systems and to provide the necessary bandwidth for high-speed, chip-to-chip, and board-to-board communications within a system. It is designed to replace the PCI-based shared, parallel-bus-signaling technology that is approaching its practical performance limits while simplifying the interface design. It includes cost, performance, and scalability advantages, ensuring a long life span for applications, system designs, and investments.

PCIe is a serial-based technology which uses low-voltage differential data signaling/lines (LVDS) to reduce the number of data signal lines and high-frequency clock signals in a point-to-point interconnect arrangement between two devices. It also serves to eliminate multiple host presences on the same bus.

### 19.1.2 Terminology Used in This Document

**Table 19-1. Terminology Used**

| Term | Definition |
|---|---|
| ADPLL | All digital phase locked loop |
| ASPM | Active state power management |
| AXI | AMBA AXI bus protocol |
| DBI | Direct bus interface |
| EP | Endpoint |
| LVDS | Low voltage differential signaling |
| MMR | Memory mapped register |
| OCP | Open core protocol |
| PCI | Peripheral component interconnect |
| PCIe | PCI Express |
| PCIESS | PCI Express subsystem |
| PCISIG | PCI special interest group |
| PHY | Physical interface device |
| PIPE | Physical interface for PCI Express - Programmable Image Processing Element |
| PME | Power management event |
| RC | Root complex |
| TC | Traffic class |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-1. Terminology Used (continued)**

| Term | Definition |
|---|---|
| TLP | Transaction layer packet |
| VC | Virtual channel |
| VCXO | Voltage controlled crystal oscillator |
| VPD | Vital Product data |

### 19.1.3 Features Supported

The PCIESS on the device supports an interface width of x1 lane. The following features are included:

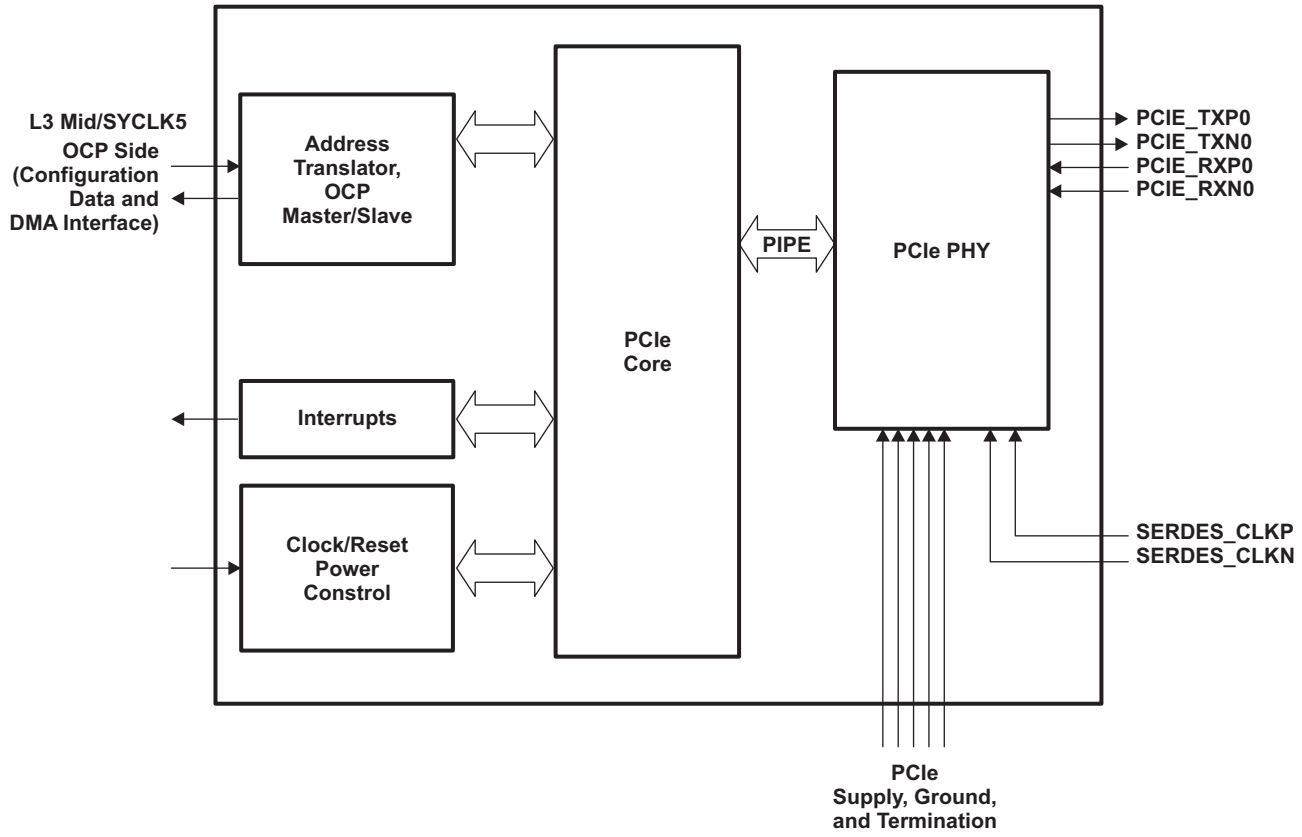
- 250 MHz functional clock frequency operation (PIPE clock frequency)
- Supports a single bi-directional link interface (for example, a single Ingress and a single Egress ports) single channel lane width (x1)
- Operates at a raw speed of 2.5Gbps and 5.0 Gbps per direction
- Maximum outbound payload size of 128 bytes
- Maximum inbound payload size of 256 bytes
- Maximum remote read request size of 256 bytes
- Ultra-low transmit and receive latency
- Supports dynamic-width conversion
- Automatic lane reversal
- Polarity inversion on receive
- Single virtual channel (VC)
- Single traffic class (TC)
- Automatic credit management
- Single function in endpoint (EP) mode
- ECRC generation and checking
- PCI device power management with the exception of D3 cold with Vaux
- PCI Express active state power management (ASPM) state L0s and L1
- PCI Express link power management states except L2 state
- PCI Express advanced error reporting
- PCI Express messages for both transmit and receive
- Filtering for posted, non-posted, and completion traffic
- Configurable BAR filtering, I/O filtering, configuration filtering and completion lookup/timeout
- Access to configuration space registers and external application memory mapped registers through BAR0 and through configuration access
- Legacy interrupts reception (Root Complex (RC)) and generation (EP)
- MSI generation and reception
- PHY loopback in RC mode

### 19.1.4 Functional Block Diagram

The PCIESS is shown in Figure 19-1.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-1. PCIe Subsystem (PCIESS) Block Diagram**




### 19.1.4.1 PCIe Core

The PCI Express core contains the transaction layer, data link layer, and MAC part of the PHY. The PCIe core is a dual mode core that can operate as either an RC or EP. As an EP, it can operate as a legacy endpoint or native PCIe endpoint. The role it assumes is based on the state of the PCIe configuration register (within the control module) bit field PCIE_DEVTYPE. PCIE_CFG.DEVTYPE = 00b/ 01b/ 10b = EP/ Legacy Endpoint/ RC, respectively. The user software or BOOTROM is required to initialize this field. When the PCIe is used as a boot device, the state of bootmode pins will determine the boot type used (in this case it will be PCIe) and the address size to be used which is either 32-bit or 64-bit addressing. The bootloader software configures the PCIe core to operate in EP mode. No RC bootmode is supported.

### 19.1.4.2 PCIe PHY

The PCI PHY (SERDES) contains the analog portion of the PHY which is the transmission line channel that is used to transmit and receive data. It contains a phase-locked loop, analog transceiver, phase interpolator-based clock/data recovery, parallel to serial converter, serial to parallel converter, scrambler, configuration, and test logic.

### 19.1.4.3 Configuration and DMA Access Interface

CPU side access, configuration registers, data access, and remote EP DMA access is performed through the configuration and DMA access interface. The maximum amount of data burst access made through this port is 128 bytes.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 19.1.4.4 Clock, Reset, Power Control Logic

Several clock domains exist within PCIESS; these are functional clocks used by the PCIe controller and interface bridges, as well as receive and transmit clocks used to clock data in and out. The clocks required for clocking data and PHY functional clocks are generated by the PHY through the supplied external input 100 MHz differential clock that complies to the requirements listed in the SERDES_CLKNP Input Clock section of the data manual. The PCIe controller functional clock frequency is generated from the internal PLL and should be programmed to generate a clock frequency of 250 MHz. The input clock source for the controller is derived from the main device input clock source and the frequency of this source should be a 20 MHz clock source.

The PCIESS supports the conventional reset mechanism that is specified within the PCI Express Specification. The reset shown on the block diagram pertains to a hardware reset (cold or warm reset).

In addition to automatic power down mode exercised and entered by the hardware (active state power management) when no activity is present, PCIESS supports higher level power down modes controlled (activated and deactivated) by application software.

Note that link state L2 is not supported.

#### 19.1.4.5 Interrupts

The PCIESS is capable of generating multiple interrupts events via the four interrupt lines (INTA/B/C/D – legacy interrupts combined, MSI, error, and PM/reset) connected to the interrupt controller. The user software is required to acknowledge the serviced interrupt by writing to the corresponding vector onto the EOI register.

#### 19.1.4.6 PCIe Power/Ground/Termination

Several supplies, grounds, and termination exist to properly power up the PHY.

#### 19.1.4.7 Differential Data Lines

A pair of differential data lines exists, for both transmit and receive paths, for each lane.

### *19.1.5 Supported Use Case Statement*

The PCIe subsystem has only one interface link with a x1 lane arrangement that connects to a single device. It also means that it can not be used as a switch.

### *19.1.6 Industry Standard(s) Compliance Statement*

The PCIESS complies with the following standards.

- Revision 2.0 of the PCI Express base specification.
- Synopsys DWC PCIe Dual Core Version 3.60a
- TI SERDES PHY (Mercury)

### *19.1.7 Features Not Supported*

The following features are not supported by the PCIESS.

- Multiple VCs
- Multiple TCs
- Outbound transactions involving less than 4 bytes
- Function level reset
- PCI Express beacon for in-band wake
- Built-in hardware support for hot plug
- Vendor messaging
- IO access in inbound direction
- Addressing modes other than incremental for burst transactions


Copyright © 2011–2016, Texas Instruments Incorporated

- Auxiliary power to maintain controller state to come out D3cold state
- Llink state L2 support

## 19.2 Peripheral Architecture

This section discusses the architecture of the PCIESS.

### *19.2.1 Clock Control*

The PCIESS uses multiple clock domains. At the top level, there are only two functional clocks of relevance – a reference clock to the PCIE PHY and a clock for OCP interfaces of the subsystem.

The OCP interface functional clock, SYSCLK4, is derived from the PLL_L3 which is generated from an off-chip/external 20 MHz VCXO.

The other clock input to the PCIESS is the differential 100MHz clock with a maximum frequency of deviation value of 300ppm. This differential clock is used by the PHY PLL to generate the necessary PHY functional, PIPE clock and Tx and Rex bit clocks required by the PHY.

The PCIESS operation is completely dependent upon availability of clock from the PLL that is inside the PHY. All registers in PCIESS are located in the clock domain that is dependent upon PLL to be in lock and properly configured. This implies that transactions initiated before PLL lock will not be processed or could result with undefined condition. The PLL registers reside at the SoC level (for example, external to the PCIe register file space) and these registers are used to control and insure/verify lock status of the PLL.

### *19.2.2 Supported PCIe Transactions*

All of the PCIe transactions defined, posted and non-posted, are supported except the locked memory read request transaction and its subsequent completion locked response transaction. In-bound I/O read/write transactions are also not supported.

**Table 19-2. PCIe Transaction Layer Packets Supported**

| Transaction Packet Types | Posted/Non-posted |
|---|---|
| Memory Read | Non-posted |
| Memory Write | Posted |
| IO read | Non-posted |
| IO Write (Outbound only is supported) | Non-posted |
| Configuration Read (Type 0 and Type 1) | Non-posted |
| Configuration Write (Type 0 and Type 1) | Non-posted |
| Message Request without Data | Posted |
| Message Request with Data | Posted |
| Completion without Data | - |
| Completion with Data | - |

### *19.2.3 Address Translations*

PCI Express TLP transactions that use address routing, use PCIe addresses. A built-in hardware address translation mechanisms accommodates the mapping requirement that exists between a PCIe address and a local OCP/internal address. That is, if the “type” field of an outgoing or received TLP indicates the use of address routing, then an Outbound or Inbound address translation is required. This translation is performed accordingly, to map an OCP/internal address to a PCIe address or vice-versa, using hardware address translators. Address translations for Outbound and Inbound transactions are discussed below.


Copyright © 2011–2016, Texas Instruments Incorporated

PCIe recognizes four address spaces: memory, I/O, configuration, and message. Messages do not consume any memory or I/O resource and no type of translation is required. I/O addressing is used by PCIe that supports legacy device operation capabilities; this is not a mandatory feature by PCIe and when supported, the I/O space is memory-mapped to system memory. This implies that there exists a need to map only two types of device address spaces:, configuration and memory address spaces. The address translators apply to translating configuration and memory addresses between external, PCIe address and OCP/internal address.

The address space of the physical device is divided into two spaces (Range 0 and Range 1). Address space (Range 0) that is used for PCI configuration task, is referred to as configuration space and a second address space (Range 1) that is used for accessing memory (non-configuration related), is referred to as memory space.

**Note:** All I/O access use address routing. There is no-support for I/O access in Inbound direction. That is, when it comes to a transaction that performs I/O access, only TLPs with outbound transactions are supported.

#### 19.2.3.1 Outbound Address Translations

Outgoing transaction layer packets (TLP) associated with address routing would require the use of the outbound address translator that creates a mapping between the device OCP/internal memory address and external PCIe address.

The PCIE Subsystem allows mapping of physical device address to PCIe address on outgoing TLPs. This is accomplished by using outbound address translation logic. For each outbound read/write request, the address translation module within PCIESS can convert an OCP address (OCP/internal address) to a PCIE address (PCIe address) of memory read/write type. The address translation logic uses information in address translation registers to perform the mapping. The registers, OB_SIZE, OB_OFFSET_INDEXn, OB_OFFSETn_HI are used in conjunction with outbound address translator.

The physical memory range that PCIESS occupies in the devices internal address range is divided into 32 equally sized translation regions (Regions 0 to 31). These equally divided regions can be programmed to have a size of 1, 2, 4, or 8 MB and this size value is communicated with the outbound address translator via the OB_SIZE register. Each such region (OCP/internal address) can be remapped to a PCIE address range the same size as the size of translation region itself. The address translation logic identifies and extracts the 5 bits (32 regions) from the OCP/internal address and uses this value as an index to one of the 32 regions. The bit address position for these 5 bits depends on the range size. Once the region is identified, the address translation logic then generates PCIe base address, from the values provided within the corresponding configuration registers for that region (the registers OB_OFFSET_INDEXn and OB_OFFSETn_HI [n=0-31]). If 32-bit addressing is used, OB_OFFSETn_HI will always be programmed with zero.

Once the PCIe base address has been identified, the offset that is to be added to this base address is derived from the lower bit fields of the OCP/internal address. The bit fields that make up this offset correspond to the size of the regions.

Application software is required to identify the desired physical memory that is to be accessible by the PCIe module and initialize the corresponding registers prior to enabling PCIe transactions. For outbound translation of the registers in use, initialization, and usage is discussed below.

- Outbound Size Register (OB_SIZE) — Application software initializes this register with the size value that applies to all 32 regions. OB_SIZE=0,1,2, and 3 correspond to region sizes of 1MB, 2MB, 4MB, and 8MB and the corresponding indexed regions for these sizes are identified from the OCP/internal address. Note that a single OB_SIZE register exist and the size of region entered/ programed pertains to all regions, for example, all regions are made of equal size. Bits[24:20], bits [25:21], bits[26:22], and bits[27:23] of the OCP/internal address holds the Index value used to identify one of the 32 regions for the 1MB, 2MB, 4MB and 8MB regions respectively. Size of regions also affect the meaningful bit fields within OB_OFFSET_INDEXn [n=0-31] register. The lower bit fields corresponding to the size of the region are masked and not used in the mapping.
- Outbound Translation Region n Offset Low and Index Register (OB_OFFSET_INDEXn [n=0-31]) — Application software initializes this register with the 32-bit PCIe address. The useful address bits used by the outbound translator from this register depends upon the size of the region. If each region is 1MB, 2MB, 4MB, or 8MB, then the relevant address field bits used to create PCIe address are the


Copyright © 2011–2016, Texas Instruments Incorporated

higher address bits, specifically bits[31:20], bits[31:21], bits[31:22], and bits[31:23] respectively.

- Outbound Address High Offset (OB_OFFSETn_HI [n=0-31]) – Application software initializes this register if using 64-bit addressing with the higher 32-bit address. This register is required to be programmed with a Zero value if using 32-bit addressing.

The following example demonstrates the mapping of a given OCP/internal address to a PCIe address.

**Example 1:**

For a given OCP/internal device address of 0x9D3A_1234; what would be the corresponding PCIe address that would be used on an outgoing TLP Header (assume a 2MB region partition)?

For this example, further assume the following: 64 bit addressing is to be used and for region 9, application software has initialized OB_OFFSET9_HI with 0x3344_5566 and OB_OFFSET9 with 0x56Ex_xxxx (where 'x' here is a don't care hexadecimal value). Note that bits[31:21] of OB_OFFSET9 register is the part of this register content that is meaningful for this example, since the size of the regions is assumed to be 2MB.

The procedure that is used to convert this OCP/internal address to a PCIe address is as follows:

- Extract the five bits index and offset from given input OCP/internal address of 0x9D31_1234: Since region size is given to be 2MB, bits [25:21] of address 0x9D3A_1234 is extracted to be used as an Index and this comes out to be 01001b, which is 9. Since size of all regions is 2MB, the offset address corresponds to bits[20:00] of the OCP/internal address. Note that the offset of the OCP/internal address, bits[20:00], is directly mapped to the PCIe address; that is, bits[20:00]. For this example that translates to 0xXX1A_1234. Since index value for this example corresponds to bits[25:21] of 0x9D3A_1234 which is 01001b or 9 decimal, translation registers contents for region 9 will be used.

Index is now matched to Region 9 and offset is 0x001A1234 which is bits [20:00] of 0x9D3A_1234.

- Generate PCIe base address: Using the Index value identified above, 9, generate base address from register OB_OFFSET9_HI (bits[63:32]) for 64-bit addressing and register OB_OFFSET9 (bits[31:21]). The PCIe base address for this example would result with: 0x3344_5566__0x56E0_0000.
- Compute translated address, PCIe address: Using the generated base address and offset, the PCIe address corresponding to an OCP/internal address of 0x9D3A_1234 is 0x3344_5566__0x56FA_1234.

Note: For this example, 2MB size region will allow unique address mapping for OCP/internal address values 0x0000_0000 to 0x03FF_FFFF. This allows a total of 2MB * 32 = 64MB unique OCP/internal address location to be mapped onto a corresponding unique PCIe address. Any OCP/internal address outside this 64MB space will not be uniquely mapped and the OCP/internal address will be truncated to fall in within the 64MB space, for this example. In other words, bits[31:26] of the given example input OCP/internal address is masked and does not contribute to the PCIe address generation.

For this example, OCP/internal addresses 0x9D3A_1234, 0x913A_1234, 0x953A_1234, 0x993A_1234, and more, are few of other physical values that will all map to the same PCIe address 0x3344_5566__0x56FA_1234.

#### 19.2.3.2 Inbound Address Translations

Incoming error free accepted Transaction Layer Packets (TLP) associated with address routing would require the use of the inbound address translator to map received PCIe address to OCP/internal address.

The PCIe subsystem allows mapping of accepted PCI Express addresses to a physical device address internal to the device using the Inbound address translation logic. For each Inbound read/write request, that results with incoming TLP, the address translation module within PCIESS can convert a PCIe address to OCP (internal/physical) address for memory or configuration read/write type transactions.

PCIESS is aware of two OCP address spaces (internal/physical memory spaces). The first address space, Address Space Zero (also known as region 0), is fixed and is dedicated for local application registers, local configuration accesses, remote configuration accesses and remote IO accesses (RC only). In other words, Address Space Zero is mapped to a fixed unique dedicated location and can not be moved or re-mapped. The second, address space, Address Space One (also known as region 1), is dedicated for data transfer and there exists no fixed or unique mapping for this address space. Details on this are captured on later sections below.


Copyright © 2011–2016, Texas Instruments Incorporated

Address Space Zero occupies a contiguous 16K Bytes of location where 4KB of 16KB is the configuration space. Address Space One is used for data buffering and is large in size and not necessarily contiguous. To perform a mapping of a PCIe addresses that would land within Address Space One, four dedicated Regions exist, (Regions [0-3]), for the use of the Inbound Address Translator. Note that Outbound translation has 32 regions while Inbound translation has four regions.

This means that any Accepted TLP that has found a match with one of the BARs, BAR[0-5] will be mapped to one of the two address spaces: Address Space Zero or Address Space One. BAR0 is dedicated to Address Space Zero and implicit mapping is done and no region association is required. However BARs[1-5] are dedicated to Address Space One and association with one of the four regions, Regions[0-3], is required. Note that this is valid in the context of an accepted TLP with 32-bit Addressing. If 64-bit Addressing is used, then the association of the six BARs with the address spaces change to a total of three since a pair of adjacent BARs concatenated is required to hold the 64-bit Address. This implies that BAR0 and BAR1 will hold 64-bit Address with BAR0 holding the low 32-bit PCIe address to match while BAR1 holds the high 32-bit PCIe address to match (associated with Address Space Zero). The same holds for Address Space One association. BARs[2-3] and BARs[4-5] will hold the accepted 64-bit address that will be associated to Address Space One where mapping takes place via the use of Regions[0-3].

Address Translation for address space zero does not exist since the internal location is unique and is contiguous. All is needed is the PCIe Address within the received TLP address matching BAR0, for 32-bit addressing, and BARs[0:1] for 64-bit addressing.

However, Address Translation for Address Space One requires the use of one of the four Regions (Regions[0-3]) to map accepted TLPs to internal/physical memory. Four region specific memory mapped registers exist and used by the Inbound Address Translator.

Association between regions (Regions[0-3]) and BARs is made via the corresponding IB_BARn [n=0-3] register. IB_STARTn_HI [n=0-3] and IB_STARTn_LO [n=0-3] are used to hold the start address of a 64-bit PCIe Address to match, and IB_OFFSETn [n=0-3] is used to hold the Offset for that particular Region (note that what is referred here as offset could be viewed as the physical Memory Base Address). Note that for 32-bit addressing, IB_STARTn_HI[0-3] should be programmed with ZERO.

The procedure used by the In-Bound Address Translator to perform the mapping between the PCIe Address and Address Space One is as follows:

1st. Extract the Offset: PCIe Address – (IB_STARTn_HI : IB_STARTn_LO)

2nd. Compute absolute OCP/internal address: Add Base Address IB_OFFSETn to the Offset extracted on step 1.

Note: When using TLPs with 64-bit addressing a pair of adjacent BARs are concatenated to hold the 64-bit Address. This implies that BAR0 and BAR1 will hold 64-bit Address with BAR0 holding the high 32-bit PCIe address to match and BAR1 holding the low 32-bit PCIe address to match. Same argument holds for BAR3, BAR4 and BAR5, BAR6 where BAR3 and BAR5 hold the high 32-bit of the 64-bit addresses.

Example 2:

For a given 64-bit PCIe Address of 0x1234_5678_ABC5_0000, that qualifies for acceptance; what would be the corresponding mapped internal (physical) device address? (Assume that Region 1 registers are programmed to match BAR2, BAR3 RC programmed addresses)?

Further assume that application software has programmed the set of registers corresponding to Region 1 with the values shown below.

IB_BAR1 = 2. This assignment associates Region 1 with BAR2, BAR3 for 64-bit Addressing. The value programmed 2 here associate the match between Region 1 (IB_BAR1) and Configuration Register BAR2.

IB_START1_HI = 0x1234_5678

IB_START1_LO = 0xABC0_0000

IB_OFFSET1 = 0x3340_0000

The OCP/internal address is computed by aubtracting the PCIe Base Address and extracting the offset from the PCIe Address (Address within the TLP Header) and adding the difference to the start Address of the OCP/internal address.


Copyright © 2011–2016, Texas Instruments Incorporated

Extract the Offset from the PCIe Address.

Extracted Offset = PCIe address – (IB_STARTn_HI : IB_STARTn_LO)

- – Extracted Offset = 0x1234_5678_ABC5_0000 – 0x1234_5678_0xABC0_0000
  = 0x0005_0000

Compute the absolute OCP/internal address.

Add the extracted Offset to the internal Base Address

OCP/internal address = Extracted Offset + IB_OFFSET

- – internal/hysical absolute address = 0x3340_0000 + 0x0005_0000
  = 0x3345_0000

In example 2, PCIe Address of 0x1234_5678_ABC5_0000 is translated or mapped to OCP/internal address, 0x3345_0000.

#### *19.2.3.2.1 BAR0 Exception for In-Bound Address Translation*

The memory space covered by BAR0 in inbound direction is completely dedicated to accessing the application registers, address space zero. It implies that the BAR0 cannot be remapped to any other location but to application registers. Any remote inbound access matching the BAR0 region will automatically be routed to these registers. It allows RC devices to control EP devices in absence of dedicated software running on EP. For RC and EP, this mapping of BAR0 to registers allows the message signaled interrupts to work. There is no support to disable BAR0 accesses from reaching the application registers.

#### *19.2.3.2.2 Mapping Multiple Non-contiguous Memory Ranges to One Region*

A single inbound TLP address translation region (matching RC programmed PCIe BAR Address) can be used to map accesses to multiple non-contiguous internal address ranges, within Address Space One, by ensuring that the start addresses of these regions are programmed in ascending order in the inbound start address registers.

Example 3:

In this example, two BARs (BAR1 and BAR2) are remapped to four separate locations (4 regions) that are non-contiguous using 32-bit addressing. The first two regions (Region 0 and Region 1) remap accesses landing in BAR1 space (IB_BAR0 = 1 and IB_BAR1 = 1 associates two Regions 0 & 1 with the same BAR, BAR1), and the following two regions (Region 2 and Region 3) remap accesses landing in BAR2 space. Each BAR is treated as a 32-bit BAR.

Assumed RC programmed Value of Base Address Registers in Configuration Space:

BAR1: 0x11110000

BAR2: 0x22220000

**Table 19-3.**

| Region (IB_BARn) | BAR | IB_STARTn_HI | IB_OFFSETn | TLPAddress | TranslatedAddress |
|---|---|---|---|---|---|
| 0 | 1 | 0x11110000 | 0x33330000 | 0x11110080 | [0x11110080 - 0x11110000] +0x33330000 = 0x33330080 |
| 1 | 1 | 0x11118000 | 0x44440000 | 0x11119000 | [0x11119000 – 0x11118000] +0x44440000 = 0x44441000 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-3. (continued)**

| | | | | | |
|---|---|---|---|---|---|
| 2 | 2 | 0x22220000 | 0x55550000 | 0x22220400 | [0x22220400 – 0x22220000] +0x55550000 = 0x55550400 |
| 3 | 2 | 0x22220800 | 0x66660000 | 0x22231000 | [0x22231000 – 0x22220800] + 0x66660000 = 0x66670800 |

Note: In case there are more than one matching BAR, then the highest start address that is greater than the TLP address is the one that is considered for translation.

#### *19.2.3.2.3 Using BAR1 Value as Start Address*

The inbound address translation for BAR1 has additional capability of using the value of BAR1 register (from PCIe Configuration Space) as the start address for inbound address translation. This feature can be activated by leaving the start address of the corresponding inbound translation region (one of the four Regions associated to BAR1) programmed with zero. When an incoming read/write access matches BAR1 and the inbound region's BAR match (IB_BARn) is set to one with start address programmed to zero, the BAR1 value is used to compute the translated address. Note that if this feature is used, only one inbound translation window is available and it will be relative to BAR1 programmed value. No other BARs or inbound translation windows can be used if BAR1 is designated as the reference for inbound address translation.

#### *19.2.3.2.4 BAR Mask Registers*

The PCI Configuration register BARs in end point devices are programmed by RC during the PCI Configuration process; that is, RC performing PCIe enumeration or PCIe device discovery process.. Since the PCIESS is a dual mode (DM) core, prior to the start of PCI configuration, the PCIESS EP device has the opportunity to modify the behavior of BAR registers prior to RC starting the enumeration process and assignment of PCIe Base Address. Using the BAR Mask registers which are overlaid on the BAR registers, the software on the EP device can configure the size of address space that the EP will request from the RC during configuration for each of its BARs. The software can also modify the BAR types and can enable/disable the BARs. That is, the read-only bit fields (BAR bits[3:0]) of the Configuration Register BARs are user programmable and are required to be programmed prior to enumeration or device discovery process begins.

Note that the BAR Mask registers are accessible, for configuration purposes of the BARs, only when operating as EP and at the same time only when CMD_STATUS[DBI_CS2] is set/enabled state. Firmware needs to always read back the data written after modification to ensure that the write has completed since the confirmation read will not happen until the write completion insuring the register update is made prior to use. Firmware needs to clear CMD_STATUS[DBI_CS2] after initial configuration prior to RC starting enumeration.

It is important to not attempt modification of BAR Mask registers from serial link side as unpredictable behavior may occur and the system could become unstable and even worse unusable.

### 19.2.3.3 Address Alignment Requirements

As per PCIe standard, no transaction can cross a 4 KB aligned address boundary.

The limit on transaction size on the internal bus interface (OCP) is 128 bytes. So, the transactions must be 128 byte or smaller on the internal bus interface. If a transaction received in inbound direction crosses a 128 byte aligned address, then the PCIESS master interface can split such transaction into multiple transactions at the 128 byte boundary.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 19.2.3.4 Byte Strobe Usage Requirements

For any type of write transactions, the byte enables can only have a single unbroken string of 1s. In other words, in a transaction, if a byte's write strobe is set, then all following bytes must have write strobe set until the last byte with write enabled. "Holes" or "Zeros" in between the byte enables are not allowed.

Since the internal bus width is greater that 32-bit, the TLP size will not be 1 (PCIe counts in 32-bit units) and therefore, it is through the FBE/LBE (First/Last Byte Enable) that the actual data transfer size is controlled.

#### 19.2.3.5 Zero-Length Read/Write Transactions

Read transactions that request zero bytes are not supported by PCIESS. PCIESS will issue a read transaction with FBE field of PCIe TLP set to 0xF. Writes of zero bytes are supported.

#### 19.2.3.6 Transactions Crossing 4KB boundary

Any single transaction that reads/writes data on locations crossing a 4KB aligned address are invalid as per of PCI Express Base Specification Revision 2.0. If such a read is issued to PCIESS Slave, it will lead to multiple transactions on the serial link so that PCIe protocol is not violated. But if the completions do not return in order from the remote PCIe link partner, then it may result in violation.

**Outbound Transaction Involving Less Than 4 Bytes**

The PCIESS imposes a limitation on the minimum size of data to be accessed in Outbound transactions. The minimum data size is 4 bytes wide. It is not possible to access less than 4 bytes. This could potentially be a problem when accessing non-prefetchable memory resulting in accesses to undesired memory location.

#### 19.2.3.7 Transaction Address Alignment

The PCIESS imposes a limitation of a maximum of 128 byte outbound read/write command. However, if the starting address is not aligned to an 8 byte boundary, then the maximum transaction size is reduced to 120 bytes. Unspecified behavior will occur if misaligned transactions in outbound direction are not limited to a maximum of 120 bytes.

#### 19.2.3.8 Read Interleaving

Read interleaving refers to the process of returning split read responses from multiple transactions. This implies that read data is not guaranteed to be sent in sequential order (data for one transaction to be sent completely before the next). PCIE core is guaranteed to not interleave read responses if the outbound read command/transaction size does not exceed the max transaction size configured in the PCIE core. Currently this is configured as 128 bytes.

#### 19.2.3.9 Endian Mode

The PCIESS is configured to operate in little endian mode. Most of the peripherals within the device are also configured in Little Endian mode. If user system is configured in big endian mode, then a conversion is required to transform data format.

### *19.2.4 Address Spaces*

Internal device addressable resources, from PCIESS perspective, are categorized in to two separate spaces. That is, PCIESS has two OCP address spaces or regions. The first is dedicated for local application registers, local configuration accesses, remote configuration accesses and remote IO accesses (RC only). This space is referred as address space zero. The second is dedicated for data transfer and it is referred as Address Space One.

#### 19.2.4.1 Address Space Zero

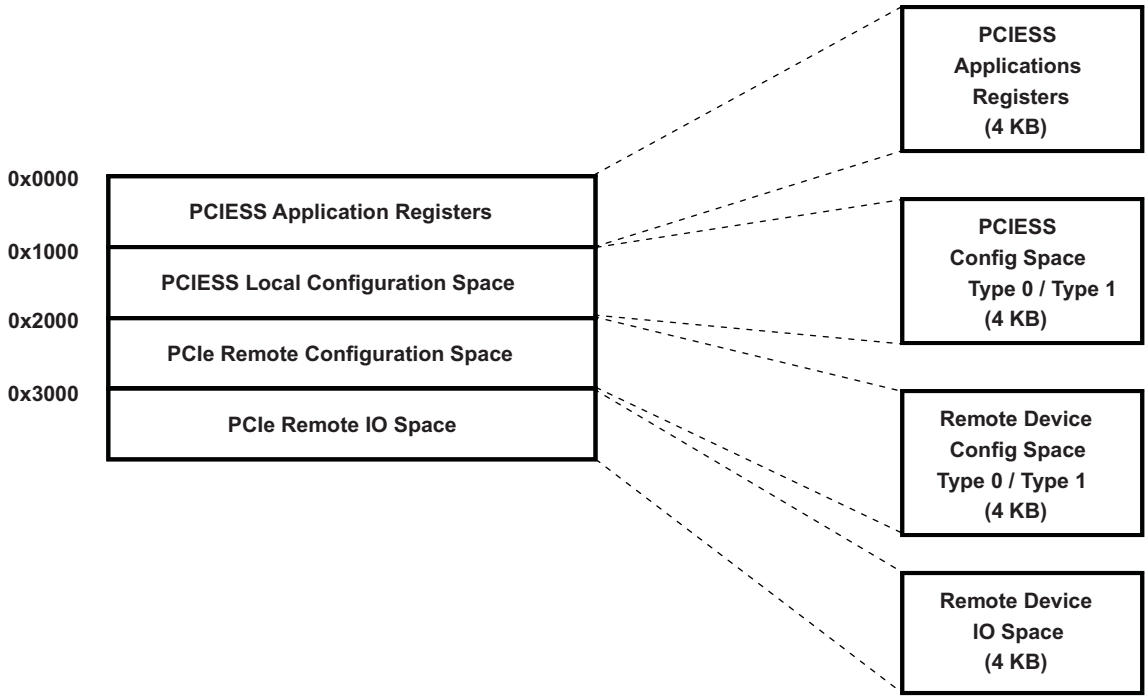
Address Space Zero is made of a contiguous 16 Kbytes of memory partitioned into four sections/regions with equal sizes (each 4 Kbytes long). These regions are:


Copyright © 2011–2016, Texas Instruments Incorporated

- PCIESS application registers
- PCIe local configuration registers
- PCIe remote configuration registers
- PCIe IO access window

Figure 19-2 illustrates the relationship of the various address regions within the address space zero.

**Figure 19-2. Address Space Zero Regions**




Each of the four address space zero regions is meant for accessing a set of registers.

- PCIESS Application Registers – These registers are used to configure and monitor various settings within the PCIESS. These registers are specific to the application needs and are not related to the PCIE Configuration registers. Note that all PCIESS Application registers should be accessed in 32-bit mode.
- PCIe Local Configuration Registers – PCIESS local configuration registers are used to read the settings of configuration registers of the local PCIE Device. Prior to PCI Express configuration is complete, these registers can also be written to. Depending upon whether PCIESS is configured as RC or EP, the layout of these registers is either Config space Type 0 or Type 1. Note that all accesses on PCIe Local Configuration registers must be made in 32-bit mode.
- PCIe Remote Configuration Registers – Remote PCIe configuration registers are accessed by programming the bus number, device number and function number of the remote PCIE device in one of the application register and then accessing the PCI Remote configuration registers as if it were the PCIE Config space of a single PCIE function. The layout of the remote configuration registers varies based upon whether the device is an end point or a PCIe switch.
- PCIe IO Access Window – A 4KB region is dedicated for remote IO accesses when PCIESS is in RC mode. Any access made on this space becomes an IO access. The actual address in the TLP gets its base address from the IO_BASE register value and an offset that is directly derived from the address of the OCP/VBUM access in this 4KB space.

None of the four address ranges described above support burst transactions. So, only single 32-bit transactions should be issued to these addresses. These addresses should also be configured as non-cacheable address space.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 19.2.4.1.1 Remote Configuration and I/O Requests (Caution)

Since the remote configuration and IO transaction windows are directly mapped to internal bus space, the software must take caution not to access these spaces when there is no operational PCIe link. No response may be generated for such transactions. It is recommended that checks be built into software to avoid remote accesses in the absence of an operational link.

### 19.2.4.2 Address Space One

The second address space/region is used for data transfer. The BAR values setup within the PCIe local configuration registers define where within the memory map of the CPU on root complex side are the end points located. All locations other than what is setup in BAR registers of the end points are on the root complex side.

The Address Space One is mapped to multiple devices in RC mode. Each remote device could be allocated a portion of this memory space and any transaction that is targeted to this address space gets converted to a PCIe transaction targeted to the appropriate remote PCIe device.

#### 19.2.4.2.1 Organization of Configuration Registers

As illustrated in Table 19-4, the registers for various PCIe capabilities are linked to each other via address offsets specified in the registers.

**Table 19-4. Register Blocks That Make up the PCIe Configuration Registers**

| Offset FROM START OF Configuration Space | REGISTER BLOCK |
|---|---|
| 0x000 | Configuration Registers Common to Type 0 and Type 1 Headers |
| 0x040 | Power Management Capabilities Registers |
| 0x050 | Message Signaled Capabilities Registers |
| 0x070 | PCI Express Capabilities Registers |
| 0x100 | PCI Express Extended Capabilities Registers |
| 0x700 | Port Logic Registers |

## 19.2.5 Bus Mastering

An end point that is capable of becoming a bus master and initiate transactions in upstream direction must have its bus master enable bit set in configuration space registers (STATUS_COMMAND[BUS_MASTER]). If transactions are initiated (while assuming the role of a RC or EP) before enabling bus mastering capability, then transactions will not start until the bus master enable is set. There is no timeout or error response generated when transactions do not go out because of the bus master bit not being set.

## 19.2.6 PCIe Loopback

The PCIe specifications provides loopback support in two ways – through PIPE interface (link layer) and second through PHY loopback capability (PHY layer).

The PIPE interface loopback requires for two PCIe devices/components to attached to each other in a loopback master and a loopback slave configuration. A loopback master is the component requesting loopback. A loopback slave is the component looping back the data. Note that, regardless with the PCIESS role it assumes (RC or EP), it can assume a role of a loopback master or a loopback slave.

The loopback, loopback at the PHY level, does not require another component.

### 19.2.6.1 PIPE Loopback

The procedure depends upon whether the device is operating in RC or EP mode. In either case, the PCIESS can be loopback master or loopback slave as outlined in PCIe specifications. Note that this loopback mode cannot be used for looping back transactions. These modes are to be used with PCIe test equipment only for symbol level loopback.


Copyright © 2011–2016, Texas Instruments Incorporated

**The Loopback entry procedure when PCIESS is a Loopback Master**

RC Mode

- Set Loopback Enable in the Port Link Control Register in Port Logic Register space.
- The link retraining sequence must be initiated by writing to Link Retrain field in the Link Control Register in PCI Express Capabilities structure in the device's PCI configuration space.

EP Mode

- Set Loopback Enable in the Port Link Control register.
- Force the LTSSM to be in recovery state via the Port Force Link register of the Port Logic registers.
- Set Force Link bit to high in Port Force Link register in Port Logic registers.

Once this is done, devices at the end of a PCIe link enter PCIe LTSSM Loopback state. The initiator of the loopback state is the Loopback Master and the other device is Loopback Slave. Note that it is not possible to send TLPs in this mode and return them via the loopback state of the other device.

**The Loopback entry procedure when PCIESS is a Loopback Slave**

If the PCIESS is a Loopback slave, then the incoming serial data is routed back to the originating device from the PIPE interface as per PCIe loopback requirements. Typically, PCIe test equipment will be used as loopback master and it will transition PCIESS into loopback slave state following which the inbound transactions will be Loopback to the test equipment. There is no programming required on PCIESS to enter Loopback in slave mode. PHY support is not required to use this loopback mode.

#### 19.2.6.2 PHY Loopback

The PHY loopback is accomplished by switching the PHY to loopback where the transmitted data is looped back to the receive path at the PHY level. This mode can be used to perform TLP loopback even if there is no link partner. It is, however, only possible to set up when PCIESS is used in RC mode. This limitation is because link training cannot occur between two upstream ports; at least one port must be a downstream port.

The procedure is similar to the one for PIPE (link layer) interface but the PHY programming is used instead. This mode is entered by configuring the SERDES configuration registers. Both the transmit and receive paths must be set in loopback mode to enable PHY loopback mode.

Note that there are several other requirements for this to correctly work:

- The PHY should be configured to operate in loopback mode before any transaction is sent out. The recommended approach is to set loopback before link training. Otherwise, any transactions that were not looped back will cause sequence numbers to increment on transmitter but not on receiver and all following transactions will be dropped because of sequence number mismatch.
- The BAR0 and BAR1 values (if programmed) should not match the address for the transaction that is issued on slave interface. Otherwise, it will not get to the master port but to internal registers.
- The memory base/limit registers should not be configured such that the address of the transaction lies in the range specified by the memory base/limit registers. Otherwise, the transaction will be discarded as a misrouted packet.
- It is not possible to use configuration type transactions in this mode as RC cannot be a target of configuration transactions. Such transactions will be invalid and loopback will not work.

### *19.2.7 L3 Memory Map*

The L3 address block for PCIESS is 256MB wide with starting address of 0x2000_0000 and ending address of 0x2FFF_FFFF. This is also referred to as Address Space One.

**NOTE:** Some processors within the device may re-map the PCIESS to a different address through use of a MMU. Also, not all masters have access to PCIESS as defined in the L3 Master/Slave Connectivity table in the data manual.


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.2.8 Signal Descriptions

This section describes each of the signals on the external interface and their functions.

### 19.2.9 Reset Considerations

The PCIESS supports the conventional reset mechanism that is specified within the PCI Express specification. Both hardware and software reset are supported. No support for the functional level reset is needed since the PCISS supports a single function.

#### 19.2.9.1 Software Reset Considerations

Software reset is issued via the transmission of TS1 Ordered Sets.

#### 19.2.9.2 Hardware Reset Considerations

Hardware reset is caused while power is being sourced to the device. The SoC level power-on-reset acts as the power on reset for PCIESS.

#### 19.2.9.3 PCIe Boot Capability

The device supports PCIe boot capability as an endpoint. No support for RC bootmode. See bootmode configuration details for the applicable settings of the bootmode pins.

#### 19.2.9.4 PCIe System Module Registers

Several configuration registers are used for configuring PCIe PHY and PLL for operation as well as test mode generation. These registers exist within the configuration module space. Consult the system module for more information.

The base address for the control module is at 0x4814_0000 for Cortex-A8 and L3 Masters.

**Table 19-5. PCIe System Module Registers**

| | | |
|---|---|---|
| 0x480 | PCIE_CFG | PCIE Configuration Register |
| 0x6D8 | PCIE_PLLCFG0 | PCIE SerDes PLL Configuration Register 0 |
| 0x6DC | PCIE_PLLCFG1 | PCIE SerDes PLL Configuration Register 1 |
| 0x6E0 | PCIE_PLLCFG2 | PCIE SerDes PLL Configuration Register 2 |
| 0x6E4 | PCIE_PLLCFG3 | PCIE SerDes PLL Configuration Register 3 |
| 0x6E8 | PCIE_PLLCFG4 | PCIE SerDes PLL Configuration Register 4 |
| 0x6EC | PCIE_PLLSTATUS | PCIE SerDes PLL Status Register |
| 0x6F0 | PCIE_RXSTATUS | PCIE SerDes Receive Status Register |
| 0x6F4 | PCIE_TXSTATUS | PCIE SerDes Transmit Status Register |
| 0xE24 | SERDES_REFCLK_CTL | SerDes Refclk Powerdown Control |

### 19.2.10 Interrupt Support

The PCIESS provides a total of four interrupts to the device interrupt controller, for example, it generates four interrupt events aggregated. In other words, each of these interrupts has more than one underlying event that the interrupt represents. Once all pending interrupts have been cleared, the hardware EOI for the interrupt generic will be generated.


Copyright © 2011–2016, Texas Instruments Incorporated

Both message signaled interrupt (MSI) and legacy interrupt are handled by the PCIESS. When operating as an EP, the PCIESS is capable of generating MSI or legacy interrupt, depending on the role it is assuming. Note that one PCIe component can not generate both type of interrupts. It is either one or the other. The interrupt type an EP generates is configured during configuration time. When operating as RC, the PCIESS is capable of handling both MSI and legacy interrupts. This is because when operating as RC it should be able to service both PCIe endpoints as well as legacy endpoints.

#### 19.2.10.1 Interrupt Events and Requests

A total of four interrupt events are supported by the PCIESS. Some of the events are meaningful based on the role the PCISS assumes (RC or EP). Table 19-6 captures the interrupt events supported for both RC and EP modes.

**Table 19-6. PCISS Interrupt Events**

| Interrupt | Interrupt Description |
|---|---|
| 0 | PCIE Express Legacy Interrupt Mode (RC mode only) |
| | [0] INTA |
| | [1] INTB |
| | [2] INTC |
| | [3] INTD |
| 1 | MSI Interrupts 0 through 31 in PCI Express Mode (EP/RC Modes) |
| | [0] MSI Interrupt 0 |
| | [1..30] MSI Interrupt 1 and so on |
| | [31] MSI Interrupt 31 |
| 2 | Error Interrupts (EP, RC) |
| | [0] System Error (OR of Fatal, Nonfatal, Correctable errors) |
| | [1] PCIe Fatal Error |
| | [2] PCIe Non-fatal Error |
| | [3] PCIe Correctable Error |
| | [4] AXI Error due to fatal condition in AXI bridge |
| | [5] PCIe Advanced Error |
| 3 | Power Management and Reset Event Interrupts (EP, RC) |
| | [0] Power Management Turn-off Message Interrupt (EP only) |
| | [1] Power Management Ack Message Interrupt |
| | [2] Power Management Event Interrupt |
| | [3] Link Request Reset Interrupt (Hot reset or Link down) |

#### 19.2.10.2 Interrupt Generation

Since the PCIESS is capable of assuming a role of RC or EP, interrupt generation capability is dependent upon the role it is assuming.

##### *19.2.10.2.1 Interrupt Generation in RC Mode*

As per PCI Express base specifications, root complex ports only receive interrupts. There is no mechanism to generate interrupts from RC port to EP mode as per PCIe specifications. However, PCIESS does support generation of interrupts from RC to EP. The behavior is similar to generation and reception of MSI interrupts in RC mode except for the fact that this functionality is enabled in EP mode as well.

The RC device can perform a memory write into the MSI IRQ register over the PCIe link to generate one of 32 EP interrupts. Note that the PCIESS will follow PCIe MSI rules and will not necessarily accumulate multiple writes to the same MSI vector. Only one of such writes is guaranteed to be processed and subsequent writes on the same vector arriving before the interrupt status is cleared may be lost.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 19.2.10.2.2 Interrupt Generation in EP Mode

When PCIESS is operating as an EP, either the legacy interrupts or the MSI interrupts can be triggered to the upstream ports (eventually leading to an interrupt in RC device). As per PCIe specifications, each PCIe function may generate only one of the legacy or MSI interrupt types as decided during configuration period.

- Legacy interrupt generation in EP mode

The endpoint can trigger generation of a PCI legacy interrupt at the root complex via an in-band Assert_INTx / Deassert_INTx PCIe Message. The actual interrupt that is generated on RC port is based on the configuration of the EP that generates the interrupt and it could be one of INTA, INTB, INTC or INTD. See Interrupt related registers in configuration space registers.

To generate an interrupt, following steps are required:

- Write 0x1 to EP_IRQ_SET register to enable the legacy interrupt
- A ASSERT INTA/B/C/D message is automatically sent.
- Write 0x1 to EP_IRQ_CLR register to disable the legacy interrupt by sending a DEASSERT INT A/B/C/D message.

Once an assert message has been generated, it cannot be generated again until a de-assert message is generated. Thus, only one interrupt can be pending at a time. The pending status can be checked in the EP_IRQ_STATUS register.

Note that the interrupt messaging mechanism makes it unfeasible to guarantee a time of delivery of the interrupt unlike in conventional designs where the interrupt line is often electrically connected to the final destination.

There is no hardware input port provided that will allow generation of legacy interrupts on the EP port.

- MSI interrupt generation in EP mode

MSI Interrupts are generated by a PCIe write transaction that performs a 32-bit memory write to a pre-determined address with a pre-determined data. The PCIe system software configures the address and the data that is to be used in the memory write transaction at the time of initialization of the EP device. The MSI scheme supports multiple interrupts and each device can request up to 32 interrupt vectors even though the allotted interrupts may be less than the requested number.

To generate MSI interrupts, following steps need to be taken:

- Ensure that the MSI support has been enabled in the device
- Read the value of MSI Address register in the local PCIe configuration space
- Read the value of MSI Data register in the local PCIe configuration space
- Determine the number of MSI vectors allocated (and the number requested) to the device
- Depending upon the number of MSI interrupts allocated, issue a Memory Write transaction with the address same as MSI Address register and data same as MSI Data register. In the data, the LSBs can be modified to reflect appropriate MSI event that needs to be notified to Root Complex
- The Memory Write transaction can also be optionally routed through the outbound address translation interface if the destination PCIe address is not directly accessible.

For more details about how MSI interrupts are expected to behave, refer to PCIe Standard Specifications.

### 19.2.10.3 Interrupt Reception

Since the PCIESS is capable of assuming a role of RC or EP, interrupt reception capability is dependant upon the role it is assuming.

#### 19.2.10.3.1 Interrupt Reception in EP Mode

The PCIe specification does not have provision for end points to receive legacy interrupts. As a result, only events other than PCIe related can cause interrupts. The MSI interrupts are not supported on EP devices as per PCIe Specification but PCIESS does support these interrupts. The MSI interrupt is generated as a result of one of the 32 events that are triggered by a write to MSI_IRQ register by the RC.


Copyright © 2011–2016, Texas Instruments Incorporated

These interrupts, delivered via register writes over the serial link, could also be coming from another end point that performs a write to the appropriate interrupt registers in the EP's BAR0 space. It is up to software designers to implement a way to determine the actual source of the interrupt.

#### 19.2.10.3.1.1 Host Reset Request Interrupt Reception in EP Mode

When the link is down, the upstream port may request reset of the end point. This request is terminated as an interrupt to theend point host software. The PCIESS automatically disables LTTSSM by de-asserting app_ltssm_enable bit in CMD_STATUS register and suspends LTSSM in DETECT QUIET state. All outstanding transactions are errored out on slave port and further transactions are no generated on master port. Once the transactions are completely stopped through OCP disconnect protocol, the software should issue a local reset to PCIESS. The re-initialization process may then be started.

#### 19.2.10.3.2 Interrupt Reception in RC Mode

When PCIESS operates in RC mode, the endpoints on the PCI fabric can be a Switch, PCIe EP or legacy EP. For this reason, it should be able to handle both MSI and legacy interrupt.

#### 19.2.10.3.3 MSI Interrupt Reception in RC Mode

A total of 32 MSI interrupts can be generated from one or more downstream devices. These MSI interrupts represent the MSI interrupts that have been allocated to various downstream devices during PCIe configuration/enumeration procedure. Before the end point devices can issue MSI interrupts, the MSI address and data registers must be configured. Each end point can either use MSI interrupts or legacy interrupts and not both at the same time. MSI interrupt have the same race condition hazard as the legacy interrupts. Hence, software drivers should take precaution.

#### 19.2.10.3.4 Legacy Interrupt Reception in RC Mode

Any of the four legacy interrupts may be generated by the PCIESS. Each interrupt can originate from multiple end point devices. The software should service these by probing the interrupt registers in each downstream device's configuration space. When all devices have been serviced, the last device serviced will send a interrupt deassert message which will clear the interrupt.

The interrupt request signal at the PCIESS boundary is a pulse signal that is triggered each time a assert interrupt message is received. The interrupt pending signal is a level signal that is high as long as the interrupt has not be serviced and the interrupt status not cleared through a register write.

Note that the traditional EOI procedure, although implemented, may not operate as expected if the deassert message arrives after the EOI has been issued. This can not be corrected but only worked around by doing a read on the interrupt register downstream before issuing EOI. If the EOI register is written to before the deassert message has reached the interrupt logic, the interrupt pending status will not have cleared and the interrupt will retrigger.

In addition, the software drivers for downstream devices must ensure that the data transaction that is expected to complete before interrupt is triggered in RC device's CPU has completed. Since PCIe write transactions are posted, it is not necessary that a write from EP to system memory in RC has completed before a write to MSI interrupt generation register has completed. This could create a potential race condition.

It is optional to support Legacy Interrupts in PCI Express (non-legacy) devices.

### 19.2.11 DMA Support

The PCIESS has no built-in DMA and makes use of the EDMA to move data in and out of the PCIESS without the need of the CPU intervention. EDMA is only used when PCIESS is the initiator of a transaction and is accessing other PCIe components. When another PCIe component is accessing a resource within the device, the PCIESS in this case acts as a master and makes use of the master port to directly access the necessary resource requiring no need for the EDMA usage. The EDMA chapter provides more details.

See the L3 Master/Slave Connectivity table in the datasheet to determine the master/slave connectivity for PCIESS.


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** Master/Slave reference here does not mean RC/EP mode. As an RC or EP, the PCIESS can act as both Master and Slave on the L3 interconnect. Instead, Master/Slave refers to who is generating the request on the L3 interconnect. If the PCIESS is generating a request on the L3 interconnect, it means that an external PCIe component is the actual entity generating the request and the PCIESS is now regarded as a Master. If the PCIESS is accessing external PCIe component, the PCIESS is regarded as a slave on the L3 interconnect.

#### 19.2.11.1 DMA Support in RC Mode

When operating in Root Complex mode, the EDMA controller can perform DMA transfer between device internal resource and any remote device located on the PCI Express fabric. The memory address of such devices is available to the software via the PCI Express bus enumeration procedure. In addition, the PCIe subsystem has a provision to perform memory address translation on outbound requests. Thus, the software is able to map different memory regions in its memory map to correspond to different addresses (and different access types) on the PCI Express side.

There are bandwidth implications of using an external DMA. If the PCIe core has been programmed to establish a link in PCIe 2.5 Gbps rate, then the DMA controller that drives PCIESS slave port must be able to write/read data at about 85% of 2 Gbps bandwidth per PCIe link. For a PCIe link speed of 5.0 Gbps, the DMA controller must be able to provide bandwidth of about 85% of 4 Gbps per PCIe link.

In addition, the master port on PCIe port can issue read/write accesses that have been initiated by remote PCI Express device. The interconnect fabric should provide sufficient capacity to serve 85% of 2 Gbps (4 Gbps if operating in Gen2) per PCIe link in each direction.

#### 19.2.11.2 DMA Support in EP Mode

When operating as a PCIe end point, the device will be located in PCIe memory map at location programmed in the Base Address registers by the PCIe root device. In end point mode, the PCIESS provides address translation functionality. It is possible to map IO, config andmemory accesses originating on PCI Express side to memory accesses with different address on the OCP side. These address ranges are configurable through application registers.

The approximate data rate for each of these transactions will be about 85% of 2 Gbps (4Gbps in Gen2 mode) in each direction on each PCIe lane.

### *19.2.12 Power Management*

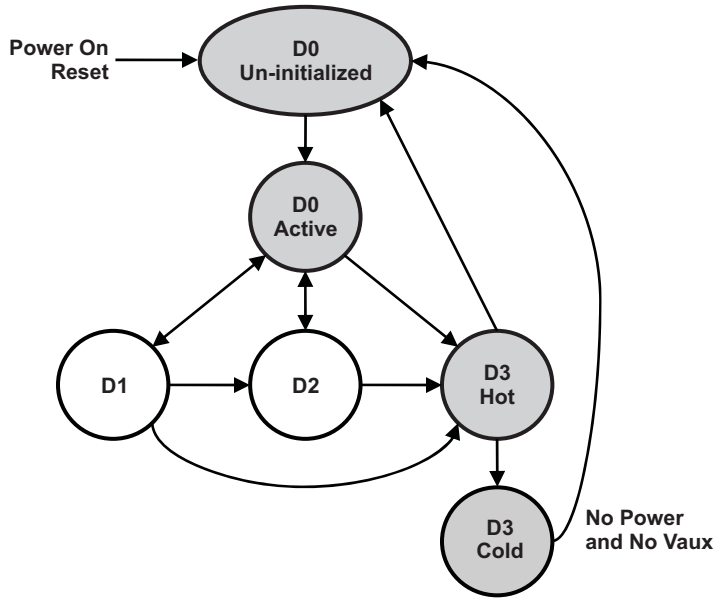
PCI Express has multiple power management protocols. Some of these are invoked by the hardware solely, Advanced State Power Management (ASPM) feature, while others are activated at higher levels via software.

#### 19.2.12.1 Device Power Management

The PCI Express protocol is compatible with all PCI power management functionality. As shown in Figure 19-3, the power states specified are D0, D1, D2, D3Hot and D3cold. All functions must support D0 and D3 states.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-3. PCI Express Power Management State Transitions**




#### 19.2.12.1.1 D0 State

Upon completion of a reset such a power-on or hot reset, the function is considered to be in D0 uninitialized state. Once the function is enumerated, configured and one or more of the memory space enable, IO space enable or bus master enable bits are set, it is considered to be in D0 active state. The D0 active state is the full-operation state of a PCI Express function.

#### 19.2.12.1.2 D1 State

Support for D1 state is optional and is primarily driven by software. It is considered to be a light sleep state that provides some power savings compared to D0 state while still allowing transition to D0 state. While in the D1 state, function must not initiate any Request TLPs on the Link except for a PME Message. Configuration and message requests are the only TLPs accepted by a Function in the D1state. All other received Requests must be handled as unsupported requests, and all received Completions may optionally be handled as unexpected completions. A function's software driver participates in the process of transitioning the Function from D0 to D1. It contributes to the process by saving any functional state (if necessary), and otherwise preparing the function for the transition to D1. As part of this quiescence process the function's software driver must ensure that any mid-transaction TLPs (for example, requests with outstanding completions), are terminated prior to handing control to the system configuration software that would then complete the transition to D1.

#### 19.2.12.1.3 D2 State

Support for D2 state is optional in PCI Express and is primarily driven by software. It provides significant power savings while still retaining capability to transition back to previous condition. In this state, the PCI function can only initiate power management events and it can only respond to configuration accesses.

#### 19.2.12.1.4 D3Hot and D3cold State

All PCI Express functions are required to support D3 state. In D3hot state, power is still present while in D3cold there is no power. Devices in D3Hot state may be transitioned back to D0 state while the devices in D3Hot state need re-initialization to be brought back to D0 state.


Copyright © 2011–2016, Texas Instruments Incorporated

Functions in D3hot respond to configuration space accesses as long as power and clock are supplied so that they can be returned to D0 by software. When programmed to D0, the function may return to the D0 Initialized or D0 Un-initialized state without PCIe reset being asserted. There is an option of either performing an internal reset or not performing an internal reset. If not performing an internal reset, upon completion of the D3hot to D0 Initialized state, no additional operating system intervention is required beyond writing the Power State bits. If the internal reset is performed, devices return to D0 Un-initialized and a full re-initialization is performed on the device. The full re-initialization sequence returns the device to D0 Initialized when normal operation may resume.

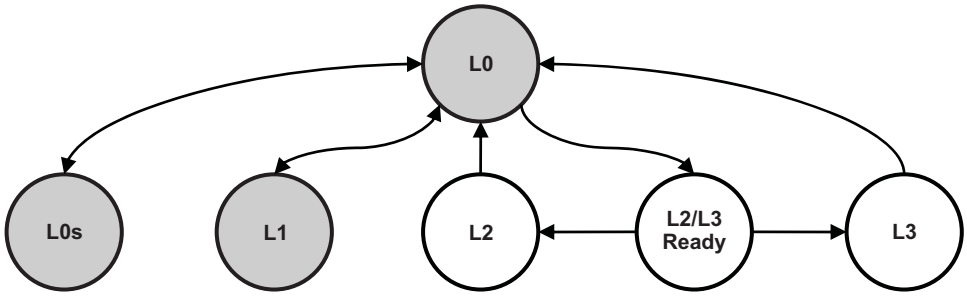
#### 19.2.12.2 Link State Power Management

There are multiple states for the physical layer – L0, L0s, L1, L2 and L3 states that provide incrementally higher power savings as the states transition from L0 towards L3. PCIESS supports L0, L0s and L1 states. L3 state is power-off state and is supported by default. PCIESS does not support L2 power down states.

Some of the link power states are autonomously managed by hardware. This scheme is referred to as Active State Power Management (ASPM). The following diagram illustrates the transitions between states L0, L0s and L1 that are managed by ASPM. The transitions between gray states are allowed in ASPM. These transitions can be enabled through registers in the PCI Express Configuration Space to allow either just L0s transition or both L0s and L1 transitions.

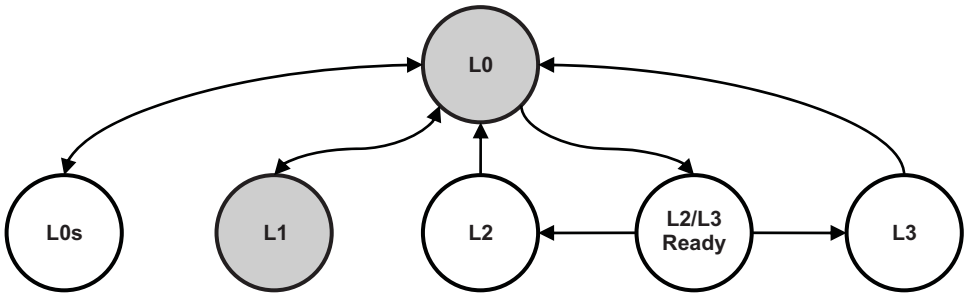
Figure 8:

**Figure 19-4. ASPM Link State Transitions Diagram**




The L1 power state can also be entered via software control. When the device power state is D1, D2 or D3hot, then the link state must transition to L1 state. The following diagram shows such state transition.

**Figure 19-5. Software driven Link Power State Transition**




##### *19.2.12.2.1 L0s State*

L0s is a lower power state enabled by Active State Power Management. Each device can control its L0s transition on its transmitter. Receiver side is controlled by the remote device.

##### *19.2.12.2.2 L1 State*

The L1 state is reached either through ASPM timer mechanism or via the software initiated transition to a D state other than the D0 state. In L1 state, the link is in idle state and both receiver and transmitter in devices on both ends of a link are able to conserve energy.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 19.2.12.2.3 L2/L3 Ready State

This state is a transitionary state reached from L1 where from the link must either transition to L2 or to L3 state. The L3 state is a full power off state. The L2 state is also a power off state except that the WAKE signal can be used by end point devices to request power and clock from the system.

#### 19.2.12.2.4 L2 State

The L2 state is appropriate when a device needs to monitor an external event while in deep power down mode. In L2 state, auxiliary power is supplied and a minimal amount of current is drawn from the power source. Almost all of the logic is devoid of power. To recover, the wake signal is used.

#### 19.2.12.2.5 L3 State

In this state, device is supplied no power from the PCIe fabric. There is no mechanism to communicate in L3 state. To recover, the system must re-establish power and reference clock followed by a fundamental reset.

### 19.2.12.3 Relationship Between Device and Link Power States

The device D-states are correlated to the link power states. For each D state, there are specific states that the PCIe link or interconnect can transition to. The table below shows the permissible state combinations.

**Table 19-7.**

| Downstream Device State | Permissible Upstream Device State | Permissible Link State |
|---|---|---|
| D0 | D0 | L0 (required), L0s (required), L1 ASPM (optional) |
| D1 | D0 – D1 | L1 |
| D2 | D0 – D2 | L1 |
| D3hot | D0 – D3hot | L1, L2/L3 Ready |
| D3cold | D0 – D3cold | L2aux, L3 |

## 19.2.13 OCP Power Management

The OCP bus power management is accomplished via the use of Idle and Standby Mode. The list of Idle/Standby states support by PCIESS are:

**Table 19-8. Supported Idle/Standby States**

| IDLE/STANDBY Mode | Remarks |
|---|---|
| NO IDLE | Supported. |
| SMART IDLE | Default State |
| SMART IDLE w/ WAKEUP | Supported |
| FORCE IDLE | Supported |
| NO STANDBY | Supported |
| SMART STANDBY | Default State |
| SMART STANDBY w/ WAKE | Not supported |
| FORCE STANDBY | Supported |

Note that if idle mode is reached in default state, the hardware will not request wakeup once in Idle mode. Software must ensure that it wakes up the hardware by exiting IDLE state in such situations.

The compatibilities of various states are tabulated below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-9. Compatibilities of Various States**

| STANDBY Mode | IDLE Mode(s) | Link States and Transitions |
|---|---|---|
| NO STANDBY | NO IDLE | States: L0 and L1Transitions: L0àL1, L1àL0 |
| SMART STANDBY | SMART IDLE WITH WAKE | States: L0, L1The valid transitions among these states are tested Transitions: L0àL1, L1àL0, L0àL2/L3 ReadyThe states L2 and L3 need fundamental reset to recover from. |
| SMART STANDBY | SMART IDLE | States: L0, L1, L2/L3 ReadyTransitions: L0àL1, L0àL2/L3 ReadyValid transitions among these states are tested. Note that the Remote partner cannot wake up PCIESS if OCP Wakeup is not enabled. A fundamental reset is always required to wake from L2 or L3 states. |
| SMART STANDBY | NO IDLE | States: L0, L1Transitions: L0àL1Only master port will enter standby. No clock gating is achievable in this configuration. |
| SMART STANDBY w/ WAKE | NA | Not supported |
| FORCE STANDBY | FORCE IDLE | States: Uninitialized Link, L0Transitions: NoneIt is not recommended to keep OCP in standby/idle and link in initialized state. This should be done only when the remote side is aware that the device interconnect in not operational and therefore, no transactions should be initiated. |

## 19.3 Supported Use Case

### *19.3.1 Software Configuration*

#### 19.3.1.1 PCIe Root Complex

When the PCIESS is desired to operate as a Root Complex (RC), the below initialization sequence are recommended to be followed.

##### *19.3.1.1.1 Initialization Sequence of PCIe Root Complex*

The initialization sequence is as follows:

- Configure PCIe mode of operation to RC mode by programming PCIE_CFG.PCIE_DEVTYPE with a value of 2.
- Bring PCIESS out of reset through the device level reset controller.
- Configure the PCIe PHY PLL registers in Control Module to enable the PHY bit clock . (These registers have to be configured in sequence)
  - SERDES_REFCLK_CTL=0x2; //PowerDown
  - PCIE_PLLCFG0 = 0; //SERDES CFG0
  - PCIE_PLLCFG1 = 0x00640000; //SERDES CFG1
  - PCIE_PLLCFG2 = 0x00000000; //cfgpll2 //SERDES CFG2
  - PCIE_PLLCFG3 = 0x004008E0; //cfgpll3 //SERDES CFG3
  - PCIE_PLLCFG4 = 0x0000609C; //cfgpll4 //SERDES CFG4
    wait a bit (about 50 µs)
  - PCIE_PLLCFG0 = 0x00000004; //Config PLL CFG0 bit [2] - ENBGSC_REF
    wait a bit (about 50 µs)
  - PCIE_PLLCFG0 = 0x00000014; //Config PLL CFG0 bit [4] - DIGLDO
    wait a bit (about 50 µs)


Copyright © 2011–2016, Texas Instruments Incorporated

  - PCIE_PLLCFG0 = 0x00000016; //Config PLL CFG0 bit [1] - ENPLLLDO
    wait a bit (about 50 µs)
  - PCIE_PLLCFG0 = 0x30000016; // Configure proxy TXLDO and RXLDO enables
    wait a bit (about 50 µs)
  - PCIE_PLLCFG0 = 0x70007016; // Configure multiplier
    wait a bit (about 200 µs)
  - PCIE_PLLCFG0 = 0x70007017); // Enable PLL
- Ensure that the link is idle. Disable link training by de-asserting the *LTSSM Enable* bit in PCIESS Control Register. Upon reset, the LTSSM Enable is de-asserted automatically by hardware, CMD_STATUS.LTSSM_EN is zero.
- Configure core registers through OCP slave interface address space 0.
- Initiate link training can be initiated by asserting *LTSSM Enable* bit in PCIESS Control register, for example, by programming CMD_STATUS.LTSSM_EN with a 1.
- Insure link training completion and success by observing DEBUG0.LTSSM_STATE field change to 0x11.
- In conjunction with the system software, start bus enumeration and setup configuration space on downstream ports.
- Continue software handshake and initialization on the remote devices. This includes setting up DMA protocols, interrupt procedures and more.
- With the completion of software initialization, DMA accesses can be started on various end points.

#### *19.3.1.1.2 Configuration Accesses*

Configuration accesses are made by RC port to individual function in each downstream EP device to program the PCIe specific operating parameters. In particular, the configuration accesses are used to allocate memory ranges for each downstream device, configure those memory ranges as IO or Memory type, enable bus master capability on the device if necessary and also build a software database of PCIE attributes and capabilities of each downstream device. Each downstream device can be configured through the configuration access region of PCIESS. The PCIESS convert memory reads and writes on the configuration region of OCP interface into configuration access on the serial link.

#### *19.3.1.1.3 Memory Accesses*

There are two types of memory accesses – the outbound memory accesses that are initiated by the DMA on the PCIESS Slave port and the inbound memory accesses that are initiated by the PCIESS Master port targeted to internal memory regions within the OCP interconnect.

The outbound PCIe memory read and write accesses are made through the PCIESS slave port through the device memory range that is dedicated to data transfers. The read and write transactions on this region are directly mapped to PCI Express space by the PCIESS in conjunction with the outbound address translation mechanism. The completions to the bus transactions are generated by PCIESS when it receives completions from remote devices. In case of errors or timeouts, an error response is provided. For reads, the error responses span as many phases as there would be data phases if the error had not occurred.

The inbound memory accesses received by PCIESS on the serial link are initiated by remote PCIe end points that are capable of bus mastership. Such accesses are converted to bus transactions on PCIESS master port and once a Response/Completion is received, the PCIESS sends data/response back to the PCIe bus master over the serial links. Typically, the inbound accesses will be targeted to memory space resident on the bus side of the RC port. The locations to which inbound memory accesses map are determined by software. The software must inform the remote devices about what protocol is to be followed so that the remote device will access the relevant memory regions to read or write data/control information. Not all end points have the capability to initiate inbound accesses.

An inbound access cannot cross a 4 KB boundary as per PCIe specifications. In addition, any access that spans a 128 byte boundary in the device memory map can get split into two transactions at the 128 byte boundary.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 19.3.1.1.4 I/O Accesses

I/O accesses are optional in PCI Express. In PCIESS, these accesses can be made through a 4KB memory space and a programmable register. All accesses made to the 4KB space become IO accesses and the IO Base register determines the target address for such accesses. The IO accesses cannot be for more than 32-bits of data aligned at 4 byte boundary.

### 19.3.1.2 PCIe End Point

When the PCIESS is desired to operate as an endpoint (EP), the below initialization sequence are recommended to be followed.

#### 19.3.1.2.1 Initialization Sequence of PCIe Endpoint

Upon de-assertion of reset, the PCIESS is configured as end point by chip level setting of PCIESS inputs. Before a Root Complex is allowed to access the configuration space of the end point, the following initialization sequence should be followed:

- Configure PCIe Mode of operation for EP Mode by programming PCIE_CFG.PCIE_DEVTYPE with a value of 0.
- Bring PCIESS out of reset through the device level reset controller.
- Configure the PCIe PHY PLL registers in the Control Module to enable the PHY bit clock . (These registers have to be configured in sequence)
  - SERDES_REFCLK_CTL=0x2; //PowerDown
  - PCIE_PLLCFG0 = 0; //SERDES CFG0
  - PCIE_PLLCFG1 = 0x00640000; //SERDES CFG1
  - PCIE_PLLCFG2 = 0x00000000; //cfgpll2 //SERDES CFG2
  - PCIE_PLLCFG3 = 0x004008E0; //cfgpll3 //SERDES CFG3
  - PCIE_PLLCFG4 = 0x0000609C; //cfgpll4 //SERDES CFG4
    wait a bit (about 50 µs)
  - PCIE_PLLCFG0 = 0x00000004; //Config PLL CFG0 bit [2] - ENBGSC_REF
    wait a bit (about 50 µs)
  - PCIE_PLLCFG0 = 0x00000014; //Config PLL CFG0 bit [4] - DIGLDO
    wait a bit (about 50 µs)
  - PCIE_PLLCFG0 = 0x00000016; //Config PLL CFG0 bit [1] - ENPLLLDO
    wait a bit (about 50 µs)
  - PCIE_PLLCFG0 = 0x30000016; // Configure proxy TXLDO and RXLDO enables
    wait a bit (about 50 µs)
  - PCIE_PLLCFG0 = 0x70007016; // Configure multiplier
    wait a bit (about 200 µs)
  - PCIE_PLLCFG0 = 0x70007017); // Enable PLL
- Ensure that the link is idle. Disable link training by de-asserting the *LTSSM Enable* bit in PCIESS Control Register. Upon reset, the LTSSM Enable is de-asserted automatically by hardware, CMD_STATUS.LTSSM_EN is zero.
- Program the configuration registers in the PCIESS to desired values.
- Initiate link training can be initiated by asserting *LTSSM Enable* bit in PCIESS Control Register, for example, by programming CMD_STATUS.LTSSM_EN with a 1.
- Insure link training completion and success by observing DEBUG0.LTSSM_STATE field change to 0x11.
- If further configuration register initialization is required, the *Application Request Retry* bit should be set. This will lead to incoming accesses to be responded with the retry response. This feature allows slow devices extra time before the Root Port assumes the devices to be inactive. Once programming is complete, de-assert *Application Request Retry* field to allow transactions from the Root Complex


Copyright © 2011–2016, Texas Instruments Incorporated

- Once configuration setup is complete, DMA transactions can begin.
- Inbound PCIe transactions will arrive at the master port of the PCIESS end point. These will be responded to by the target slave devices and the PCIESS will relay the response back to the PCIE device that initiated the transaction.
- Outbound PCIe transactions will be targeted to the slave port of the PCIESS end point. These transactions will be serviced only if the PCIESS end point has been given Bus Master Capability by the Root Complex.

#### 19.3.1.2.2 Configuration Accesses

As an endpoint, the PCIESS can only be a target of configuration accesses from upstream. Until the link is established and APP_RETRY_EN is disabled, the PCIESS will not respond to configuration accesses. When enabled, the PCIESS will respond to configuration accesses automatically and these accesses do not get relayed to the master interface on the device interconnect side.

End point is not capable to accessing configuration space of devices other than its own. The system software can initialize the read only fields in PCIESS Configuration Space via the slave port before the link training has been initiated. Once the configuration is complete by the PCI Express Root Complex, the system software should not modify the PCIESS configuration parameters. There is no explicit prevention mechanism in hardware to disallow such accesses though.

#### 19.3.1.2.3 Memory Accesses

There are two types of Memory Accesses – inbound and outbound memory accesses.

An inbound access is typically initiated by a Root Complex port or by another end point that is reaching the PCIESS via a PCI Express switch that supports peer-to-peer access. In either case, the incoming PCIe transaction results in an access on the PCIESS master port. The response to the such accesses is relayed back to the originating PCIe device.

In an outbound access, a DMA module or a host CPU initiates a read/write access on the PCIESS Slave port. This request is converted into a PCIe memory read/write transaction over the PCIE link. Once the PCIESS receives completion from the remote device, it generates a completion on the OCP slave port. The software/hardware that initiates the request on the slave port must perform it in a memory region that has been determined previously through software protocols. For example, the software may get information about applicable memory regions from the software that is running on the Root Complex device.

#### 19.3.1.2.4 I/O Accesses

IO accesses are not supported in PCIESS operating as a PCIe end point.

### 19.3.2 Accessing Read-only Registers in Configuration Space

Some of the register fields provided in the configuration space can be written to prior to bus enumeration via the slave interface of PCIESS. Note that the hardware does not prevent modification of read only field after bus enumeration but it is strongly advised that read only fields not be written once the bus enumeration is complete.

In addition, the BAR Mask registers can also be programmed by first enabling the DBI_CS2 bit and then performing writes on the BAR registers. The BAR mask registers are overlaid on BAR registers. For the BAR mask registers to be writable, the respective BAR must first be enabled.

### 19.3.3 Accessing EP Application Registers from PCIe RC

The application registers are also mapped at 2K and above address, in the configuration space. The RC software can access these registers over PCIe link provided the registers are programmed to some default values by the BOOT code in the PCIESS host device that enables link training and TLP exchange.


Copyright © 2011–2016, Texas Instruments Incorporated

## 19.4 Encoding of LTSSM State in DEBUG registers

The LTSSM state value that is read from DEBUG0 Register bits[4:0] is an encoded value. The literal names of the LTSSM states corresponding to the encoded values are tabulated below.

**Table 19-10. LTSSM States**

| CODE | LTSSM STATE |
|---|---|
| 0x00 | DETECT_QUIET |
| 0x01 | DETECT_ACT |
| 0x02 | POLL_ACTIVE |
| 0x03 | POLL_COMPLIANCE |
| 0x04 | POLL_CONFIG |
| 0x05 | PRE_DETECT_QUIET |
| 0x06 | DETECT_WAIT |
| 0x07 | CFG_LINKWD_START |
| 0x08 | CFG_LINKWD_ACEPT |
| 0x09 | CFG_LANENUM_WAIT |
| 0x0A | CFG_LANENUM_ACEPT |
| 0x0B | CFG_COMPLETE |
| 0x0C | CFG_IDLE |
| 0x0D | RCVRY_LOCK |
| 0x0E | RCVRY_SPEED |
| 0x0F | RCVRY_RCVRCFG |
| 0x10 | RCVRY_IDLE |
| 0x11 | L0 |
| 0x12 | L0S |
| 0x13 | L123_SEND_EIDLE |
| 0x14 | L1_IDLE |
| 0x15 | L2_IDLE |
| 0x16 | L2_WAKE |
| 0x17 | DISABLED_ENTRY |
| 0x18 | DISABLED_IDLE |
| 0x19 | DISABLED |
| 0x1A | LPBK_ENTRY |
| 0x1B | LPBK_ACTIVE |
| 0x1C | LPBK_EXIT |
| 0x1D | LPBK_EXIT_TIMEOUT |
| 0x1E | HOT_RESET_ENTRY |
| 0x1F | HOT_RESET |

## 19.5 PCIESS Memory Map

### *19.5.1 Application Registers*

The Application registers are accessible in multiple ways depending on the point of origin of such accesses. When accessed from the OCP side (on board CPU or EDMA performing the access), these registers are accessed via the 4KB space in Address Space or Address Region Zero. When accessed from the PCIe serial link side, the application registers are mapped to BAR0 of an EP as well as RC. In other words, if a PCIe component desires access to the application registers, based on the address type used (memory or configuration) the right block of memory is accessed automatically. In addition, an RC can access these registers in a PCIESS EP via the upper 1 KB space of the PCIe configuration space. The offsets of the registers remain the same regardless with the access method used.


Copyright © 2011–2016, Texas Instruments Incorporated

The starting base address of the application registers (also referred as PCIe Configuration registers) is 0x5100_0000.

#### 19.5.1.1 Application Registers Summary

The applications registers summary is shown below.

**Table 19-11. Applications Registers Summary**

| Name | Reset | Offset |
|---|---|---|
| PID | NA | 0x000 |
| CMD_STATUS | NA | 0x004 |
| CFG_SETUP | NA | 0x008 |
| IOBASE | NA | 0x00C |
| TLPCFG | NA | 0x010 |
| RSTCMD | NA | 0x014 |
| PMCMD | NA | 0x020 |
| PMCFG | NA | 0x024 |
| ACT_STATUS | NA | 0x028 |
| OB_SIZE | NA | 0x030 |
| DIAG_CTRL | NA | 0x034 |
| ENDIAN | NA | 0x038 |
| PRIORITY | NA | 0x03C |
| IRQ_EOI | NA | 0x050 |
| MSI_IRQ | NA | 0x054 |
| EP_IRQ_SET | NA | 0x064 |
| EP_IRQ_CLR | NA | 0x068 |
| EP_IRQ_STATUS | NA | 0x06C |
| GPRO | NA | 0x070 |
| GPR1 | NA | 0x074 |
| GPR2 | NA | 0x078 |
| GPR3 | NA | 0x07C |
| MSI0_IRQ_STATUS_RAW | NA | 0x100 |
| MSI0_IRQ_STATUS | NA | 0x104 |
| MSI0_IRQ_ENABLE_SET | NA | 0x108 |
| MSI0_IRQ_ENABLE_CLR | NA | 0x10C |
| IRQ_STATUS_RAW | NA | 0x180 |
| IRQ_STATUS | NA | 0x184 |
| IRQ_ENABLE_SET | NA | 0x188 |
| IRQ_ENABLE_CLR | NA | 0x18C |
| ERR_IRQ_STATUS_RAW | NA | 0x1C0 |
| ERR_IRQ_STATUS | NA | 0x1C4 |
| ERR_IRQ_ENABLE_SET | NA | 0x1C8 |
| ERR_IRQ_ENABLE_CLR | NA | 0x1CC |
| PMRST_IRQ_STATUS_RAW | NA | 0x1D0 |
| PMRST_IRQ_STATUS | NA | 0x1D4 |
| PMRST_ENABLE_SET | NA | 0x1D8 |
| PMRST_ENABLE_CLR | NA | 0x1DC |
| OB_OFFSET_INDEXn | NA | 0x200 |
| OB_OFFSETn_HI | NA | 0x204 |
| IB_BAR0 | NA | 0x300 |
| IB_START0_LO | NA | 0x304 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-11. Applications Registers Summary (continued)**

| Name | Reset | Offset |
|---|---|---|
| IB_START0_HI | NA | 0x308 |
| IB_OFFSET0 | NA | 0x30C |
| IB_BAR1 | NA | 0x310 |
| IB_START1_LO | NA | 0x314 |
| IB_START1_HI | NA | 0x318 |
| IB_OFFSET1 | NA | 0x31C |
| IB_BAR2 | NA | 0x320 |
| IB_START2_LO | NA | 0x324 |
| IB_START2_HI | NA | 0x328 |
| IB_OFFSET2 | NA | 0x32C |
| IB_BAR3 | NA | 0x330 |
| IB_START3_LO | NA | 0x334 |
| IB_START3_HI | NA | 0x338 |
| IB_OFFSET3 | NA | 0x33C |
| PCS_CFG0 | NA | 0x380 |
| PCS_CFG1 | NA | 0x384 |
| PCS_STATUS | NA | 0x388 |
| SERDES_STATUS | NA | 0x38C |
| SERDES_RXCFG0 | NA | 0x390 |
| SERDES_RXCFG1 | NA | 0x394 |
| SERDES_RXCFG2 | NA | 0x398 |
| SERDES_RXCFG3 | NA | 0x39C |
| SERDES_RXCFG4 | NA | 0x3A0 |
| SERDES_TXCFG0 | NA | 0x3A4 |
| SERDES_TXCFG1 | NA | 0x3A8 |
| SERDES_TXCFG2 | NA | 0x3AC |
| SERDES_TXCFG3 | NA | 0x3B0 |
| SERDES_TXCFG4 | NA | 0x3B4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.1.2 PID Register

The peripheral version and ID register (PID) is described in the figure and table below.

**Figure 19-6. PID Register**

| 31 | 30 | 29 | 28 | 27 | 16 |
|---|---|---|---|---|---|
| SCHEME | | Reserved | | FUNC | |
| R-1 | | R-0 | | R-E31h | |

| 15 | 11 | 10 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| RTL | | MAJOR | | CUSTOM | | MINOR | |
| R-1 | | R-1 | | R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-12. PID Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | SCHEME | 0-3h | PID Register Format Scheme |
| 29-28 | Reserved | 0 | Reserved |
| 27-16 | FUNC | 0-FFFh | Function code of the peripheral |
| 15-11 | RTL | 0-1Fh | RTL version number (R) |
| 10-8 | MAJOR | 0-7h | Major revision code (X) |
| 7-6 | CUSTOM | 0-3h | Custom code |
| 5-0 | MINOR | 0-3Fh | Minor revision code (Y) |

### 19.5.1.3 CMD_STATUS Register

The command status register (CMD_STATUS) is described in the figure and table below.

**Figure 19-7. CMD_STATUS Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|
| Reserved | | OCP_STANDBY | | OCP_IDLE | |
| R-0 | | R/W-2h | | R/W-2h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | DBI_CS2 | APP_RETRY_EN | POSTED_WR_EN | IB_XLT_EN | OB_XLT_EN | LTSSM_EN |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-13. CMD_STATUS Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | Reserved | 0 | Reserved |
| 11-10 | OCP_STANDBY | 0-3h | The **OCP standby mode bitfield** defines the local clock-management behavior of the PCIe module based on Standby (master) protocol with the device PRCM. **For more information on Standby states relations to the PCIe protocol link power states, refer to** Section 19.2.13, *OCP Power Management* . |
| | | | **0h : Force standby** / The module unconditionally asserts the standby request to the PRCM module, regardless of its internal operations. The PRCM module may gate the functional and interface clocks to the module. This mode must be used carefully **(should NOT be used in normal usage but for debug purposes only)** because it does not prevent loss of data at the time the clocks are gated. / |
| | | | **1h : No standby** / The module never asserts the standby request to the PRCM module. This mode is safe from a module point of view because it ensures that the clocks remain active; however, it is not efficient from a power-saving perspective because it never allows the PRCM module output clocks to be gated. / |
| | | | **2h : Smart standby** / The module asserts the standby request based on its internal activity status. The standby signal is asserted only when all ongoing transactions are complete and the module is idled. The PRCM module can then gate the clocks to the module. Note that, this mode is NOT wake-up capable, i.e. the module can NOT generate (IRQ- or DMA request-related) wake-up events when in IDLE state. / |
| | | | **3h :** Reserved |
| 9-8 | OCP_IDLE | 0-3h | The **OCP idle mode bitfield** defines the local clock-management behavior of the PCIe module based on Idle (Slave) protocol with the device PRCM. **For more information on Idle states relations to the PCIe protocol link power states, refer to** Section 19.2.13, *OCP Power Management*. |
| | | | **0h : Force Idle** / The module unconditionally acknowledges an IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully **(should NOT be used in normal usage but for debug purposes only)**, because it does not prevent loss of data at the time the clock is switched off. / |
| | | | **1h : No Idle** / The module never acknowledges any IDLE request from the PRCM module. This mode is safe from a module point of view because it ensures the clocks remain active; however, it is not efficient from a power-saving perspective because it does not allow the PRCM module output clock to be shut off, and thus the power domain to be set to a lower power state. / |
| | | | **2h : Smart Idle** / The module acknowledges an IDLE request, basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or DMA requests are processed. This is the best approach to efficient system power management. Note that, this mode is NOT wake-up capable, i.e. the module can NOT generate (IRQ- or DMA request- related) wake-up events when in IDLE state. / |
| | | | **3h : Smart Idle (wake-up capable)** / The module acknowledges the IDLE request, basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or DMA requests are processed. This is the best approach to efficient system power management. The module may generate (IRQ- or DMA request- related) wake-up events when in IDLE state. / |
| 7-6 | Reserved | 0 | Reserved |
| 5 | DBI_CS2 | 0 | Set to enable writing to BAR mask registers that are overlaid on BAR registers. |
| 4 | APP_RETRY_EN | 0 | Application Request Retry Enable—Setting this bit will enable all incoming PCIe transactions to be returned with a retry response. This feature can be used if initialization can take longer than PCIe stipulated time frame. |
| 3 | POSTED_WR_EN | 0 | Posted Write Enable—Setting this bit will cause the OCP master to use posted write commands. Default is zero with all OCP Master writes defaulting to non-posted. |
| 2 | IB_XLT_EN | 0 | Inbound Address Translation Enable—Setting this bit will enable translation of inbound memory/io read/write requests into memory read/write requests. |
| 1 | OB_XLT_EN | 0 | Outbound Address Translation Enable—Setting this bit will enable translation of outbound memory read/write requests into memory/io/cfg read/write requests. |
| 0 | LTSSM_EN | 0 | Link Transitioning Enable—Setting this bit will enable LTSSM in PCI Express Core and link negotiation with link partner will begin. |

### 19.5.1.4 CFG_SETUP Register

The config transaction setup register (CFG_SETUP) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-8. CFG_SETUP Register**

| 31 | 25 | 24 | 23 | 16 |
|---|---|---|---|---|
| Reserved | | CFG_TYPE | CFG_BUS | |
| R-0 | | R/W-0 | R/W-0 | |

| 15 | 13 | 12 | 8 | 7 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | CFG_DEVICE | | Reserved | | CFG_FUNC | |
| R-0 | | R/W-0 | | R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-14. CFG_SETUP Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-25 | Reserved | 0 | Reserved |
| 24 | CFG_TYPE | 0 | Configuration Type for outbound configuration accesses. Set for Type 1 access and clear for Type 0 access. |
| 23-16 | CFG_BUS | 0-FFh | PCIe Bus number for outbound configuration accesses |
| 15-13 | Reserved | 0 | Reserved |
| 12-8 | CFG_DEVICE | 0-1Fh | PCIe Device number for outbound configuration accesses |
| 7-3 | Reserved | 0 | Reserved |
| 2-0 | CFG_FUNC | 0-7h | PCIe Function number for outbound configuration accesses |

### 19.5.1.5 IOBASE Register

The IO TLP base register (IOBASE) is described in the figure and table below.

**Figure 19-9. IOBASE Register**

| 31 | 16 |
|---|---|
| IOBASE | |
| R/W-0 | |

| 15 | 12 | 11 | 0 |
|---|---|---|---|
| IOBASE | | Reserved | |
| R/W-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-15. IOBASE Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | IOBASE | 0-F FFFFh | Bits 31-12 of outgoing IO TLP. RC mode only. |
| 11-0 | Reserved | 0 | Reserved |

### 19.5.1.6 TLPCFG Register

The TLP attribute configuration register (TLPCFG) is described in the figure and table below.

**Figure 19-10. TLPCFG Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | RELAXED | NO_SNOOP |
| R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-16. TLPCFG Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1 | RELAXED | 0 | Enable Relaxed Ordering for all outgoing TLPs. |
| 0 | NO_SNOOP | 0 | Enable No Snoop attribute on all outgoing TLPs. |

### 19.5.1.7 RSTCMD Register

The reset command and status register ( RSTCMD) is described in the table and figure below.

**Figure 19-11. RSTCMD Register**

| 31 | 17 | 16 |
|---|---|---|
| Reserved | | FLUSH_N |
| R-0 | | R-1 |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | INIT_RST |
| R-0 | | W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-17. RSTCMD Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved |
| 16 | FLUSH_N | 1 | Bridge Flush Status—Reads a zero when no transaction is pending. Used to ensure no pending transactions before issuing warm reset. Only applicable in PCIESS versions that use Designware core version 3.57 and newer. |
| 15-1 | Reserved | 0 | Reserved |
| 0 | INIT_RST | 0 | Write one to initiate a downstream hot reset sequence on downstream. |

### 19.5.1.8 PMCMD Register

The power management command register (PMCD) is described in the figure and table below.

**Figure 19-12. PMCMD Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | PM_XMT_TURNOFF | PM_XMT_PME |
| R-0 | | W1S-0 | W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-18. PMCMD Register Field Descriptions**

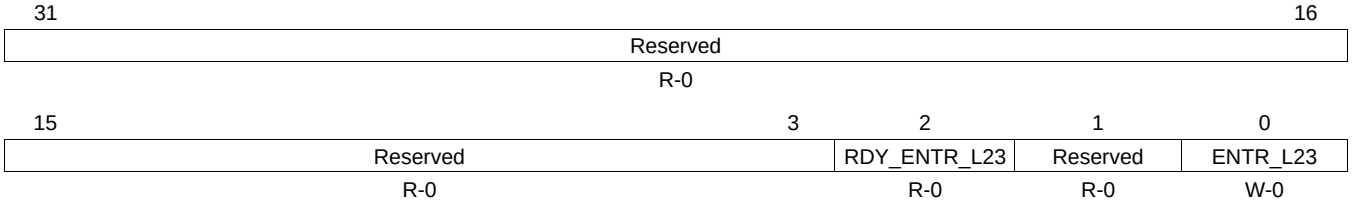
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1 | PM_XMT_TURNOFF | 0 | Write one to transmit a PM_TURNOFF message. Reads zero. Applicable in RC mode only. |
| 0 | PM_XMT_PME | 0 | Write one to transmit a PM_PME message. Reads zero. Applicable to EP mode only. |

### 19.5.1.9 PMCFG Register

The power management configuration register (PMCFG) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-13. PMCFG Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

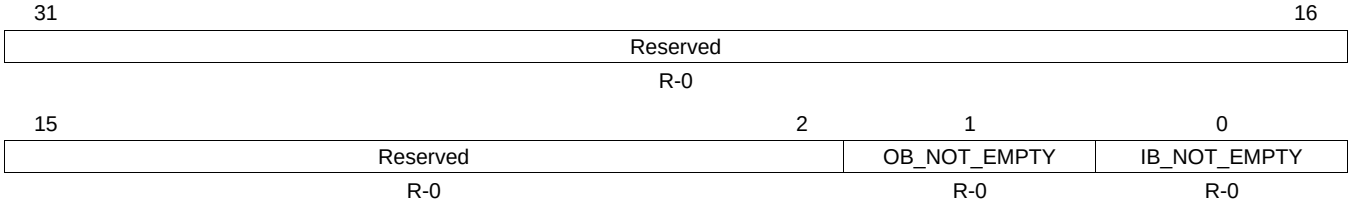
**Table 19-19. PMCFG Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2 | RDY_ENTR_L23 | 0 | Read L2/L3 entry readiness. Applicable to RC and EP. |
| 1 | Reserved | 0 | Reserved |
| 0 | ENTR_L23 | 0 | Write one to enable entry to L2/L3 ready state. Applicable to RC and EP. |

### 19.5.1.10 ACT_STATUS Register

The activity status register (ACT_STATUS) is described in the figure and table below.

**Figure 19-14. ACT_STATUS Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

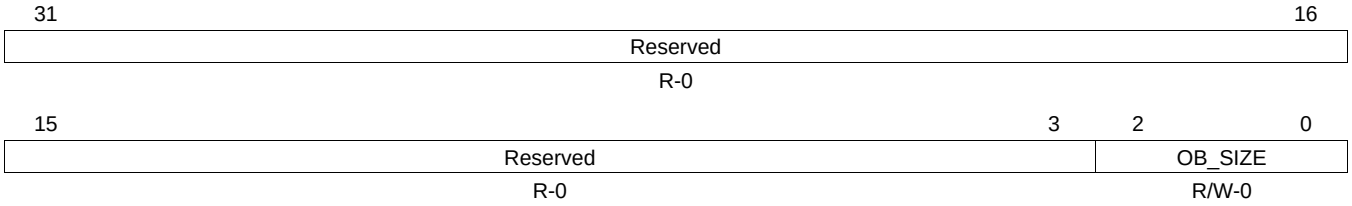
**Table 19-20. ACT_STATUS Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1 | OB_NOT_EMPTY | 0 | Outbound buffers are empty if this bit is read as 0. |
| 0 | IB_NOT_EMPTY | 0 | Inbound buffers are empty if this bit is read as 1. |

### 19.5.1.11 OB_SIZE Register

The outbound size register (OB_SIZE) is described in the figure and table below.

**Figure 19-15. OB_SIZE Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-21. OB_SIZE Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2-0 | OB_SIZE | 0-7h | Write 0, 1, 2, or 3 and so on to set each outbound translation window size to 1, 2, 4, or 8 MB region size. Applicable to RC and EP. |

### 19.5.1.12 DIAG_CTRL Register

The diagnostic control register (DIAG_CTRL) is described in the figure and table below.

**Figure 19-16. DIAG_CTRL Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | INV_ECRC | INV_LCRC |
| R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-22. DIAG_CTRL Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1 | INV_ECRC | 0 | Write 1 to force inversion of LSB of ECRC for next one packet. It is self-cleared when the ECRC error has been injected on one TLP. |
| 0 | INV_LCRC | 0 | Write 1 to force inversion of LSB of LCRC for next one packet. It is self-cleared when the ECRC error has been injected on one TLP. |

### 19.5.1.13 ENDIAN Register

The endian mode register (ENDIAN) is described in the figure and table below.

**Figure 19-17. ENDIAN Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | ENDIAN_MODE | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-23. ENDIAN Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1-0 | ENDIAN | 0-3h | Applicable in CBA version of PCIESS only. Not applicable to OCP bus. |

### 19.5.1.14 PRIORITY Register

The CBA transaction priority register (PRIORITY) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-18. PRIORITY Register**

| 31 | 17 | 16 |
|---|---|---|
| Reserved | | MST_PRIV |
| R-0 | | R/W-0 |

| 15 | 12 | 11 | 8 | 7 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | MST_PRIVID | | Reserved | | MST_PRIORITY | |
| R-0 | | R/W-0 | | R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-24. PRIORITY Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved |
| 16 | MST_PRIV | 0 | Value on master transactions |
| 15-12 | Reserved | 0 | Reserved |
| 11-8 | MST_PRIVID | 0-Fh | Value on master transactions |
| 7-2 | Reserved | 0 | Reserved |
| 1-0 | MST_PRIORITY | 0-3h | Priority level for each inbound transaction on the CBA master port. This field is not used for OCP interface. |

### 19.5.1.15 IRQ_EOI Register

The end of interrupt register (IRQ_EOI) is described in the figure and table below.

**Figure 19-19. IRQ_EOI Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | EOI | |
| R-0 | | W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-25. EOI Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1-0 | EOI | 0-3h | EOI for each interrupt. Write to indicate end-of-interrupt for the interrupt events. Write 0 to mark EOI for INTA/INTB/INTC/INTD, 1 to mark EOI for MSI interrupts and so on. |

### 19.5.1.16 MSI_IRQ Register

The MSI interrupt IRQ register (MSI_IRQ) is described in the figure and table below.

**Figure 19-20. MSI_IRQ Register**

| 31 | 0 |
|---|---|
| MSI_IRQ | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

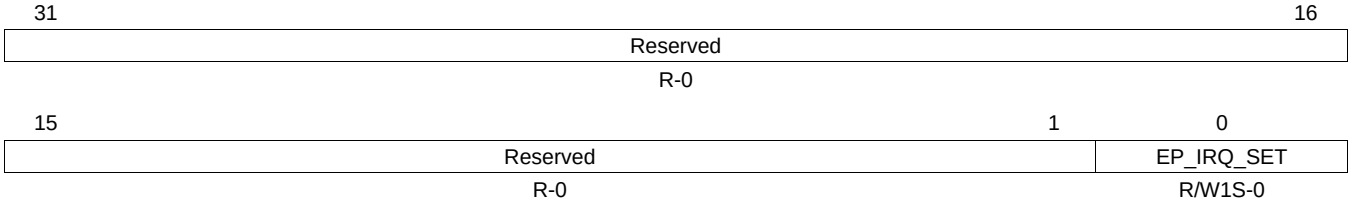
**Table 19-26. MSI_IRQ Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | MSI_IRQ | 0-FFFF FFFFh | This register is written to by the remote device. Writes initiated by an EP over PCIe link that target BAR0 of the RC land to this register if the offset matches. To generate MSI Interrupt 0, the EP should write 0000_0000h to this register. It will result in a pulse on bit 0 triggering the MSI interrupt from PCIESS to the external processor. |

### 19.5.1.17 EP_IRQ_SET Register

The endpoint interrupt request set register (EP_IRQ_SET) is described in the figure and table below.

**Figure 19-21. EP_IRQ_SET Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

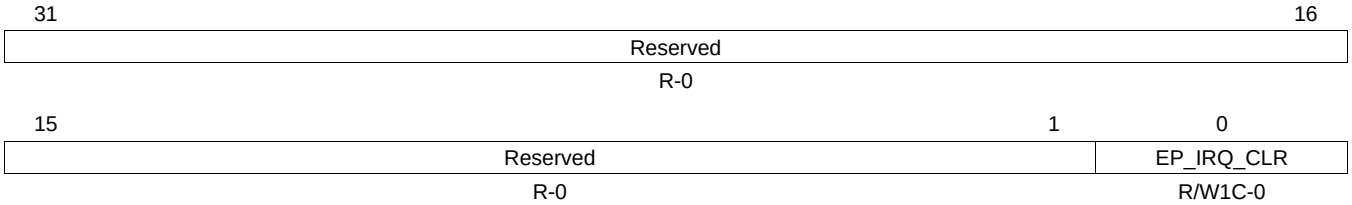
**Table 19-27. EP_IRQ_SET Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | EP_IRQ_SET | 0 | Write 1 to generate assert interrupt message. If MSI is disabled, legacy interrupt assert message will be generated. On read, a 1 indicates currently asserted interrupt. |

### 19.5.1.18 EP_IRQ_CLR Register

The endpoint interrupt request clear register (EP_IRQ_CLR) is described in the figure and table below.

**Figure 19-22. EP_IRQ_CLR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-28. EP_IRQ_CLR Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | EP_IRQ_CLR | 0 | Write 1 to generate deassert interrupt message. If MSI is disabled, legacy interrupt deassert message will be generated. On read, a 1 indicates currently asserted. |

### 19.5.1.19 EP_IRQ_STATUS Register

The endpoint interrupt status register (EP_IRQ_STATUS) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-23. EP_IRQ_STATUS Register**

| 31 | 1 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | EP_IRQ_STATUS |
| R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-29. EP_IRQ_STATUS Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved |
| 0 | EP_IRQ_STATUS | 0 | Indicates whether interrupt for function 0 is asserted or not. |

### 19.5.1.20 GPRO Register

The general purpose 0 register (GPR0) is described in the figure and table below.

**Figure 19-24. GPR0 Register**

| 31 | 0 |
|---|---|
| GENERIC0 | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-30. GPR0 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | GENERIC0 | 0-FFFF FFFFh | Generic Info field 0 |

### 19.5.1.21 GPR1 Register

The general purpose 1 register (GPR1) is described in the figure and table below.

**Figure 19-25. GPR1 Register**

| 31 | 0 |
|---|---|
| GENERIC1 | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-31. GPR1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | GENERIC1 | 0-FFFF FFFFh | Generic Info field 1 |

### 19.5.1.22 GPR2 Register

The general purpose 2 register (GPR2) is described in the figure and table below.

**Figure 19-26. GPR2 Register**

| 31 | 0 |
|---|---|
| GENERIC2 | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-32. GPR2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | GENERIC2 | 0-FFFF FFFFh | Generic Info field 2 |

### 19.5.1.23 GPR3 Register

The general purpose 3 register (GPR3) is described in the figure and table below.

**Figure 19-27. GPR3 Register**

| 31 | 0 |
|---|---|
| GENERIC3 | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-33. GPR3 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | GENERIC3 | 0-FFFF FFFFh | Generic Info field 3 |

### 19.5.1.24 MSI0_IRQ_STATUS_RAW Register

The MSI 0 interrupt raw status register (MSI0_IRQ_STATUS_RAW) is described in the figure and table below.

**Figure 19-28. MSI0_STATUS_RAW Register**

| 31 | 0 |
|---|---|
| MSI0_RAW_STATUS | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-34. MSI0_STATUS_RAW Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | MSI0_RAW_STATUS | 0-FFFF FFFFh | Each bit indicates raw status of MSI vector associated with the bit. Typically, writes to this register are only done for debug purposes. |

### 19.5.1.25 MSI0_IRQ_STATUS Register

The MSI 0 interrupt enabled status register (MSI0_IRQ_STATUS) is described in the figure and table below.

**Figure 19-29. MSI0_IRQ_STATUS Register**

| 31 | 0 |
|---|---|
| MSI0_IRQ_STATUS | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-35. MSI0_IRQ_STATUS Register Field Descriptions**

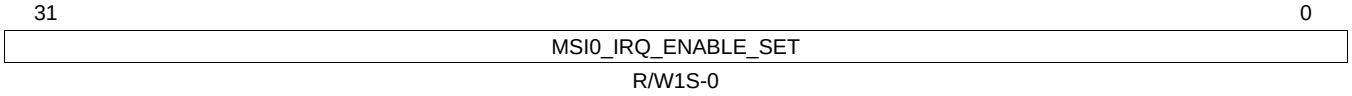
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | MSI0_IRQ_STATUS | 0-FFFF FFFFh | Each bit indicates raw status of MSI vector associated with the bit. Each of the bits can be written with a one to clear the respective interrupt status bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.1.26 MSI0_IRQ_ENABLE_SET Register

The MSI 0 interrupt enable set register (MSI0_IRQ_ENABLE_SET) is described in the figure and table below.

**Figure 19-30. MSI0_IRQ_ENABLE_SET Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

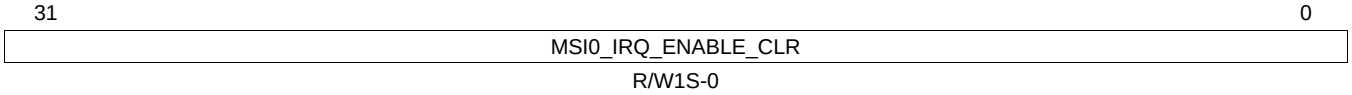
**Table 19-36. MSI0_IRQ_ENABLE_SET Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | MSI0_IRQ_ENABLE_SET | 0-FFFF FFFFh | Each bit, when written to, enables the MSI interrupt associated with the bit. |

### 19.5.1.27 MSI0_IRQ_ENABLE_CLR Register

The MSI0 interrupt enable clear register (MSI0_IRQ_ENABLE_SET) is described in the figure and table below.

**Figure 19-31. MSI0_IRQ_ENABLE_CLR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

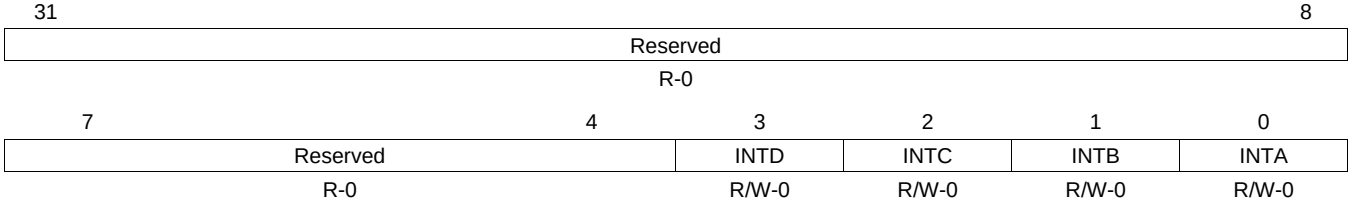
**Table 19-37. MSI0_IRQ_ENABLE_CLR Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | MSI0_IRQ_ENABLE_CLR | 0-FFFF FFFFh | Each bit, when written to, disables the MSI interrupt associated with the bit. |

### 19.5.1.28 IRQ_STATUS_RAW Register

The raw interrupt status register (IRQ_STATUS_RAW) is described in the figure and table below.

**Figure 19-32. IRQ_STATUS_RAW Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-38. IRQ_STATUS_RAW Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | INTD | 0 | Legacy Interrupt D raw status. RC mode only. |
| 2 | INTC | 0 | Legacy Interrupt C raw status. RC mode only. |
| 1 | INTB | 0 | Legacy Interrupt B raw status. RC mode only. |
| 0 | INTA | 0 | Legacy Interrupt A raw status. RC mode only. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.1.29 IRQ_STATUS Register

The interrupt enabled status register (IRQ_STATUS) is described in the figure and table below.

**Figure 19-33. IRQ_STATUS Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | INTD | INTC | INTB | INTA |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-39. IRQ_STATUS Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | INTD | 0 | Legacy Interrupt D status. Set when interrupt is active. Write one, '1', to clear the interrupt event. RC mode only. |
| 2 | INTC | 0 | Legacy Interrupt C status. Set when interrupt is active. Write one, '1', to clear the interrupt event. RC mode only. |
| 1 | INTB | 0 | Legacy Interrupt B status. Set when interrupt is active. Write one, '1', to clear the interrupt event. RC mode only. |
| 0 | INTA | 0 | Legacy Interrupt A status. Set when interrupt is active. Write one, '1' to clear the interrupt event. RC mode only. |

### 19.5.1.30 IRQ_ENABLE_SET Register

The interrupt enable set register (IRQ_ENABLE_SET) is described in the figure and table below.

**Figure 19-34. IRQ_ENABLE_SET Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | INTD | INTC | INTB | INTA |
| R-0 | | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-40. IRQ_ENABLE_SET Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | INTD | 0 | Legacy Interrupt D status. Set to enable the interrupt. On read, one/zero means interrupt is enabled/disabled. |
| 2 | INTC | 0 | Legacy Interrupt C status. Set to enable the interrupt. On read, one/zero means interrupt is enabled/disabled. |
| 1 | INTB | 0 | Legacy Interrupt B status. Set to enable the interrupt. On read, one/zero means interrupt is enabled/disabled. |
| 0 | INTA | 0 | Legacy Interrupt A status. Set to enable the interrupt. On read, one/zero means interrupt is enabled/disabled. |

### 19.5.1.31 IRQ_ENABLE_CLR Register

The interrupt enable clear register (IRQ_ENABLE_CLR) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-35. IRQ_ENABLE_CLR Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | INTD | INTC | INTB | INTA |
| R-0 | | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-41. IRQ_ENABLE_CLR Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | INTD | 0 | Legacy Interrupt D disable. Set to enable the interrupt. On read, one/zero means interrupt is enabled/disabled. |
| 2 | INTC | 0 | Legacy Interrupt C disable. Set to enable the interrupt. On read, one/zero means interrupt is enabled/disabled. |
| 1 | INTB | 0 | Legacy Interrupt B disable. Set to enable the interrupt. On read, one/zero means interrupt is enabled/disabled. |
| 0 | INTA | 0 | Legacy Interrupt A disable. Set to enable the interrupt. On read, one/zero means interrupt is enabled/disabled. |

### 19.5.1.32 ERR_IRQ_STATUS_RAW Register

The raw ERRinterrupt status register (ERR_IRQ_STATUS_RAW) is described in the figure and table below.

**Figure 19-36. ERR_IRQ_STATUS_RAW Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ERR_AER | ERR_AXI | ERR_CORR | ERR_NONFATAL | ERR_FATAL | ERR_SYS |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-42. ERR_IRQ_STATUS_RAW Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5 | ERR_AER | 0 | ECRC error raw status. |
| 4 | ERR_AXI | 0 | AXI tag lookup fatal error raw status. |
| 3 | ERR_CORR | 0 | Correctable error raw status. |
| 2 | ERR_NONFATAL | 0 | Nonfatal error raw status. |
| 1 | ERR_FATAL | 0 | Fatal error raw status. |
| 0 | ERR_SYS | 0 | System Error (FATAL, NONFATAL, or CORRECTABLE error) raw status. |

### 19.5.1.33 ERR_IRQ_STATUS Register

The ERR interrupt enabled status register (ERR_IRQ_STATUS) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-37. ERR_IRQ_STATUS Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ERR_AER | ERR_AXI | ERR_CORR | ERR_NONFATAL | ERR_FATAL | ERR_SYS |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-43. ERR_IRQ_STATUS Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5 | ERR_AER | 0 | ECRC error raw status. |
| 4 | ERR_AXI | 0 | AXI tag lookup fatal error raw status. |
| 3 | ERR_CORR | 0 | Correctable error raw status. |
| 2 | ERR_NONFATAL | 0 | Nonfatal error raw status. |
| 1 | ERR_FATAL | 0 | Fatal error raw status. |
| 0 | ERR_SYS | 0 | System Error (FATAL, NONFATAL, or CORRECTABLE error) raw status. |

### 19.5.1.34 ERR_IRQ_ENABLE_SET Register

The ERR interrupt enable set register (ERR_IRQ_ENABLE_SET) is described in the figure and table below.

**Figure 19-38. ERR_IRQ_ENABLE_SET Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ERR_AER | ERR_AXI | ERR_CORR | ERR_NONFATAL | ERR_FATAL | ERR_SYS |
| R-0 | | R/W-0 | R/W-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-44. ERR_IRQ_ENABLE_SET Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5 | ERR_AER | 0 | ECRC error interrupt enable. Set to enable. On read, one/zero means enabled/disabled respectively. |
| 4 | ERR_AXI | 0 | AXI tag lookup fatal error interrupt enable. Set to enable. On read, one/zero means enabled/disabled respectively. |
| 3 | ERR_CORR | 0 | Correctable error interrupt enable. Set to enable. On read, one/zero means enabled/disabled respectively. |
| 2 | ERR_NONFATAL | 0 | Nonfatal error interrupt enable. Set to enable. On read, one/zero means enabled/disabled respectively. |
| 1 | ERR_FATAL | 0 | Fatal error interrupt enable. Set to enable. On read, one/zero means enabled/disabled respectively. |
| 0 | ERR_SYS | 0 | System Error (FATAL, NONFATAL, or CORRECTABLE error) interrupt enable. Set to enable. On read, one/zero means enabled/disabled respectively. |

### 19.5.1.35 ERR_IRQ_ENABLE_CLR Register

The ERR interrupt enable clear register (ERR_IRQ_ENABLE_CLR) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-39. ERR_IRQ_ENABLE_CLR Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ERR_AER | ERR_AXI | ERR_CORR | ERR_NONFATAL | ERR_FATAL | ERR_SYS |
| R-0 | | R/W-0 | R/W-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-45. ERR_IRQ_ENABLE_CLR Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5 | ERR_AER | 0 | ECRC error interrupt disable. Set to disable. On read, one/zero means enabled/disabled respectively. |
| 4 | ERR_AXI | 0 | AXI tag lookup fatal error interrupt disable. Set to disable. On read, one/zero means enabled/disabled respectively. |
| 3 | ERR_CORR | 0 | Correctable error interrupt disable. Set to disable. On read, one/zero means enabled/disabled respectively. |
| 2 | ERR_NONFATAL | 0 | Nonfatal error interrupt disable. Set to disable. On read, one/zero means enabled/disabled respectively. |
| 1 | ERR_FATAL | 0 | Fatal error interrupt disable. Set to disable. On read, one/zero means enabled/disabled respectively. |
| 0 | ERR_SYS | 0 | System Error (FATAL, NONFATAL, or CORRECTABLE error) interrupt disable. Set to disable. On read, one/zero means enabled/disabled respectively. |

### 19.5.1.36 PMRST_IRQ_STATUS_RAW Register

The power management and reset interrupt status register (PMRST_IRQ_STATUS_RAW) is described in the figure and table below.

**Figure 19-40. PMRST_IRQ_STATUS_RAW Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | LINK_RST_REQ | PM_PME | PM_TO_ACK | PM_TURNOFF |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-46. PMRST_IRQ_STATUS_RAW Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | LINK_RST_REQ | 0 | Link Request Reset interrupt raw status |
| 2 | PM_PME | 0 | Power Management PME message received interrupt raw status |
| 1 | PM_TO_ACK | 0 | Power Management ACK received interrupt raw status |
| 0 | PM_TURNOFF | 0 | Power Management Turnoff message received raw status |

### 19.5.1.37 PMRST_IRQ_STATUS Register

The power management and reset interrupt enabled status register (PMRST_IRQ_STATUS) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-41. PMRST_IRQ_STATUS Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | LINK_RST_REQ | PM_PME | PM_TO_ACK | PM_TURNOFF |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-47. PMRST_IRQ_STATUS Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | LINK_RST_REQ | 0 | Link Request Reset interrupt raw status |
| 2 | PM_PME | 0 | Power Management PME message received interrupt raw status |
| 1 | PM_TO_ACK | 0 | Power Management ACK received interrupt raw status |
| 0 | PM_TURNOFF | 0 | Power Management Turnoff message received raw status |

### 19.5.1.38 PMRST_ENABLE_SET Register

The power management and reset interrupt enabled set register (PMRST_ENABLE_SET) is described in the figure and table below.

**Figure 19-42. PMRST_ENABLE_SET Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | LINK_RST_REQ | PM_PME | PM_TO_ACK | PM_TURNOFF |
| R-0 | | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-48. PMRST_ENABLE_SET Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | LINK_RST_REQ | 0 | Link Request Reset interrupt enable. Set to enable the interrupt. Read 1 means interrupt is enabled. |
| 2 | PM_PME | 0 | Power Management PME message received interrupt enable. Set to enable the interrupt. Read 1 means interrupt is enabled. |
| 1 | PM_TO_ACK | 0 | Power Management ACK received interrupt enable. Set to enable the interrupt. Read 1 means interrupt is enabled. |
| 0 | PM_TURNOFF | 0 | Power Management Turnoff message received enable. Set to enable the interrupt. Read 1 means interrupt is enabled. |

### 19.5.1.39 PMRST_ENABLE_CLR Register

The power management and reset interrupt enabled clear register (PMRST_ENABLE_CLR) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-43. PMRST_ENABLE_CLR Register**

| 31 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | LINK_RST_REQ | PM_PME | PM_TO_ACK | PM_TURNOFF |
| R-0 | | R/W1S-0 | R/W1S-0 | R/W1S-0 | R/W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-49. PMRST_ENABLE_CLR Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | LINK_RST_REQ | 0 | Link Request Reset interrupt disable. Set to disable the interrupt. Read 1 means interrupt is enabled. |
| 2 | PM_PME | 0 | Power Management PME message received interrupt disable. Set to disable the interrupt. Read 1 means interrupt is enabled. |
| 1 | PM_TO_ACK | 0 | Power Management ACK received interrupt disable. Set to disable the interrupt. Read 1 means interrupt is enabled. |
| 0 | PM_TURNOFF | 0 | Power Management Turnoff message received disable. Set to disable the interrupt. Read 1 means interrupt is enabled. |

### 19.5.1.40 OB_OFFSET_INDEXn (0x200 + N*8) Register

The outbound translation region N offset low and index register (OB_OFFSET_INDEXn) is described in the figure and table below.

**Figure 19-44. OB_OFFSET_INDEXn Register**

| 31 | 20 | 19 | 16 |
|---|---|---|---|
| OB_OFFSETn_LO | | Reserved | |
| R/W-0 | | R-0 | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | OB_ENABLEn |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-50. OB_OFFSET_INDEXn Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | OB_OFFSETn_LO | 0-FFFh | Offset bits 31-20 for translation region N (N=0-31) |
| 19-1 | Reserved | 0 | Reserved |
| 0 | OB_ENABLEn | 0 | Enable translation region N (N=0-31) |

### 19.5.1.41 OB_OFFSETn_HI Register (0x200 + N*8 + 0x4)

The outbound translation region N offset high register (OB_OFFSETn_HI ) is described in the figure and table below.

**Figure 19-45. OB_OFFSETn_HI Register**

| 31 | 0 |
|---|---|
| OB_OFFSETn_HI | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

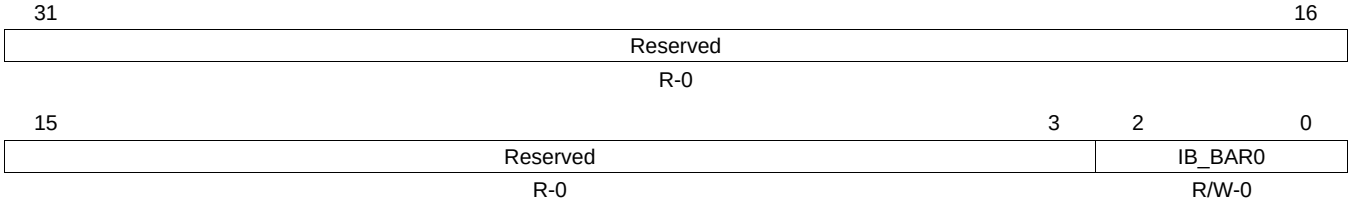
**Table 19-51. OB_OFFSETn_HI Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | OB_OFFSETn_HI | 0-FFFF FFFFh | Offset bits 31-0 for translation region N (N=0-31) |

### 19.5.1.42 IB_BAR0 Register

The inbound translation bar match 0 register (IB_BAR0 ) is described in the figure and table below.

**Figure 19-46. IB_BAR0 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

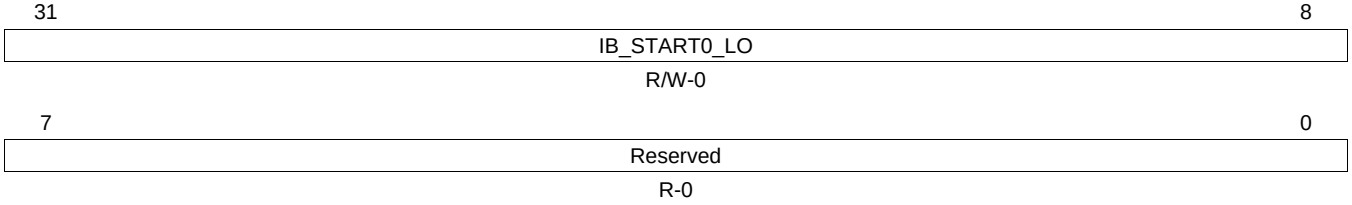
**Table 19-52. IB_BAR0 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2-0 | IB_BAR0 | 0-7h | BAR number to match for inbound translation region 0 |

### 19.5.1.43 IB_START0_LO Register

The inbound translation 0 start address low register (IB_START0_LO) is described in the figure and table below.

**Figure 19-47. IB_START0_LO Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-53. IB_START0_LO Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | IB_START0_LO | 0-FF FFFFh | Start address bits 31-8 for inbound translation region 0. |
| 7-0 | Reserved | 0 | Reserved |

### 19.5.1.44 IB_START0_HI Register

The inbound translation 0 start address high register (IB_START0_HI) is described in the figure and table below.

**Figure 19-48. IB_START0_HI Register**

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

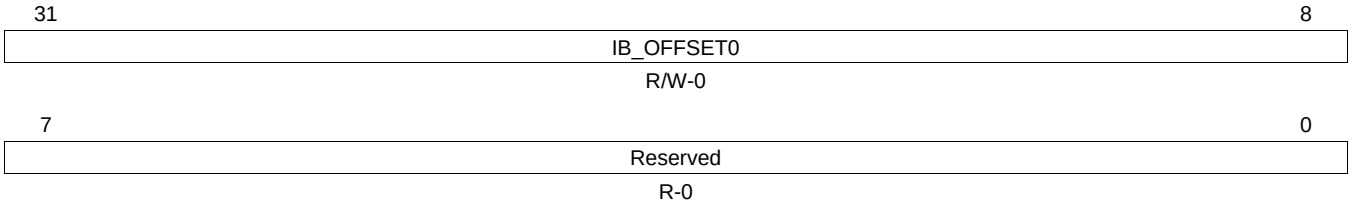
**Table 19-54. IB_START0_HI Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | IB_START0_HI | 0-FFFF FFFFh | Start address bits 63-0 for inbound translation region 0. |

### 19.5.1.45 IB_OFFSET0 Register

The inbound translation 0 address offset register (IB_OFFSET0) is described in the figure and table below.

**Figure 19-49. IB_OFFSET0 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

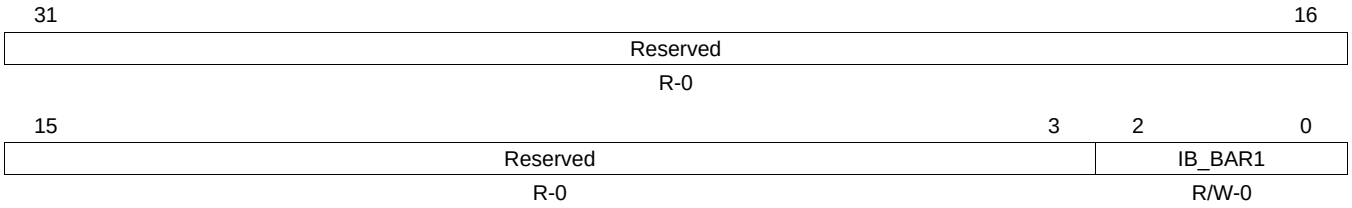
**Table 19-55. IB_OFFSET0 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | IB_OFFSET0 | 0-FF FFFFh | Offset address bits 31-8 for inbound translation region 0. |
| 7-0 | Reserved | 0 | Reserved |

### 19.5.1.46 IB_BAR1 Register

The inbound translation bar match 1 register (IB_BAR1) is described in the figure and table below.

**Figure 19-50. IB_BAR1 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-56. IB_BAR1 Register Field Descriptions**

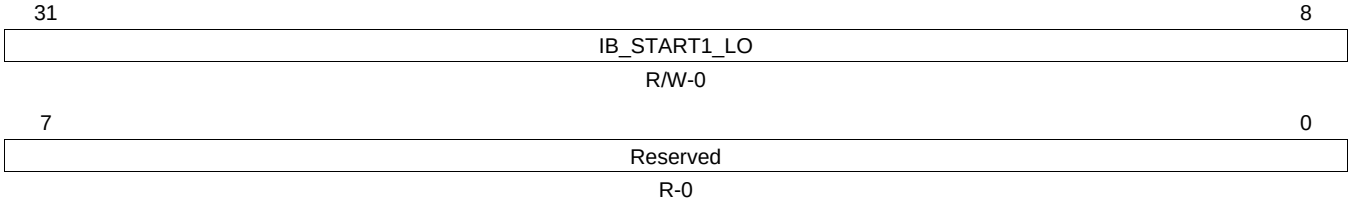
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2-0 | IB_BAR1 | 0-7h | BAR number to match for inbound translation region 1 |

### 19.5.1.47 IB_START1_LO Register

The inbound translation 1 start address low register (IB_START1_LO) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-51. IB_START1_LO Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

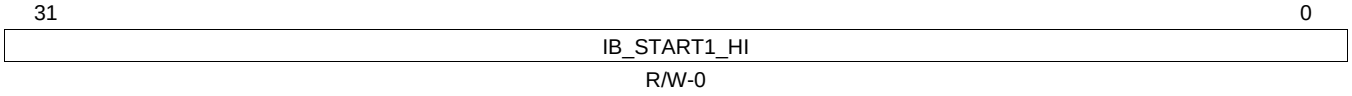
**Table 19-57. IB_START1_LO Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | IB_START1_LO | 0-FF FFFFh | Start address bits 31-8 for inbound translation region 1 |
| 7-0 | Reserved | 0 | Reserved |

### 19.5.1.48 IB_START1_HI Register

The inbound translation 1 start address high register (IB_START1_HI) is described in the figure and table below.

**Figure 19-52. IB_START1_HI Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

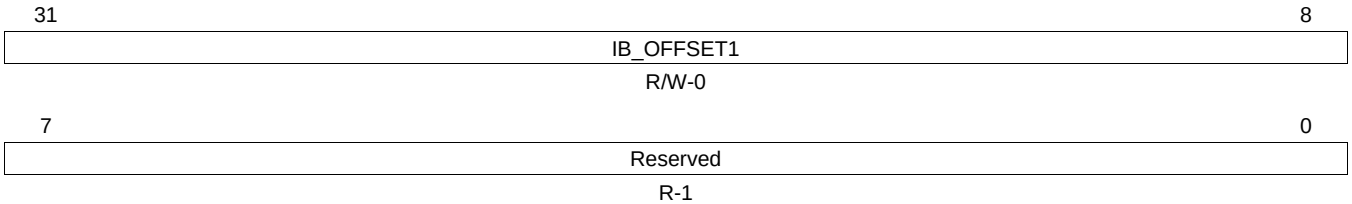
**Table 19-58. IB_START1_HI Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | IB_START1_HI | 0-FFFF FFFFh | Start address bits 63-0 for inbound translation region 1 |

### 19.5.1.49 IB_OFFSET1 Register

The inbound translation 1 address offset register (IB_OFFSET1) is described in the figure and table below.

**Figure 19-53. IB_OFFSET1 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-59. IB_OFFSET 1 Register Field Descriptions**

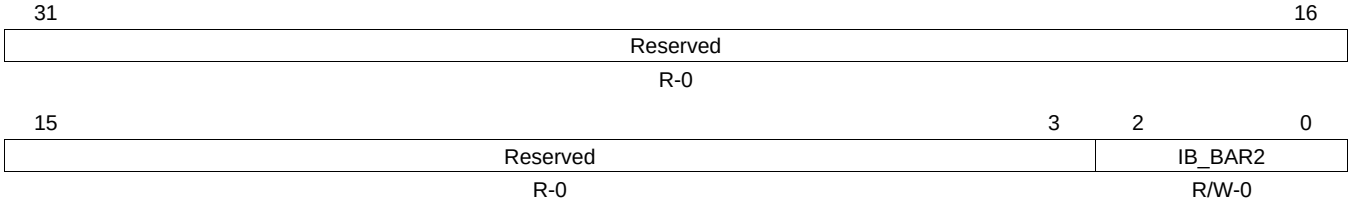
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | IB_OFFSET1 | 0-FF FFFFh | Offset address bits 31-8 for inbound translation region 1 |
| 7-0 | Reserved | 0 | Reserved |

### 19.5.1.50 IB_BAR2 Register

The inbound translation bar match 2 register (IB_BAR2) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-54. IB_BAR2 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

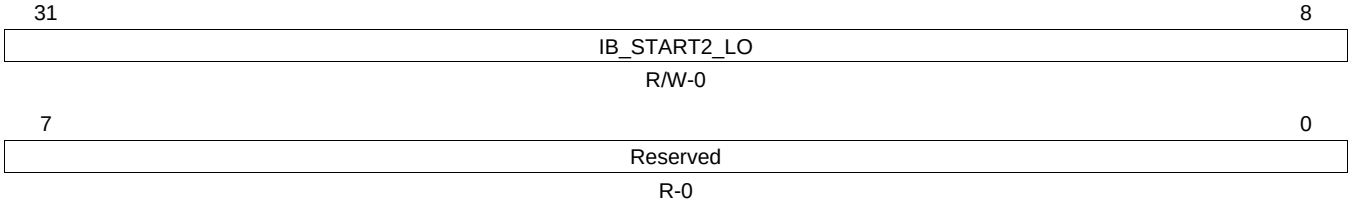
**Table 19-60. IB_BAR2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2-0 | IB_BAR2 | 0-7h | BAR number to match for inbound translation region 2 |

### 19.5.1.51 IB_START2_LO Register

The inbound translation 2 start address low register (B_START2_LO) is described in the figure and table below.

**Figure 19-55. IB_START2_LO Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

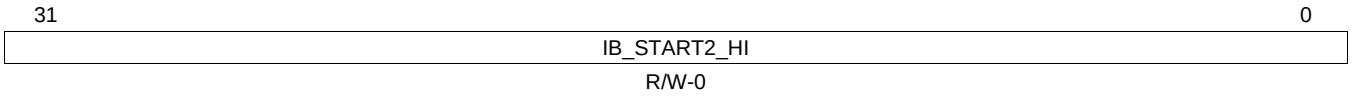
**Table 19-61. IB_START2_LO Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | IB_START2_LO | 0-FF FFFFh | Start address bits 31-8 for inbound translation region 2 |
| 7-0 | Reserved | 0 | Reserved |

### 19.5.1.52 IB_START2_HI Register

The inbound translation 2 start address low register (B_START2_HI) is described in the figure and table below.

**Figure 19-56. IB_START2_HI Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-62. IB_START2_HI Register Field Descriptions**

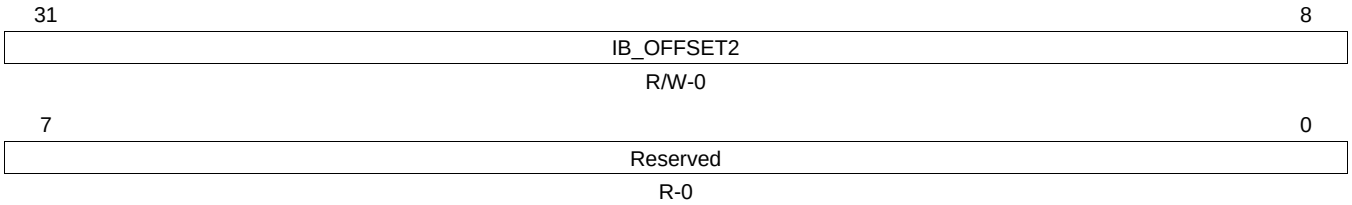
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | IB_START2_HI | 0-FFFF FFFFh | Start address bits 63-0 for inbound translation region 2 |

### 19.5.1.53 IB_OFFSET2 Register

The inbound translation 2 address offset register (IB_OFFSET2) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-57. IB_OFFSET2 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

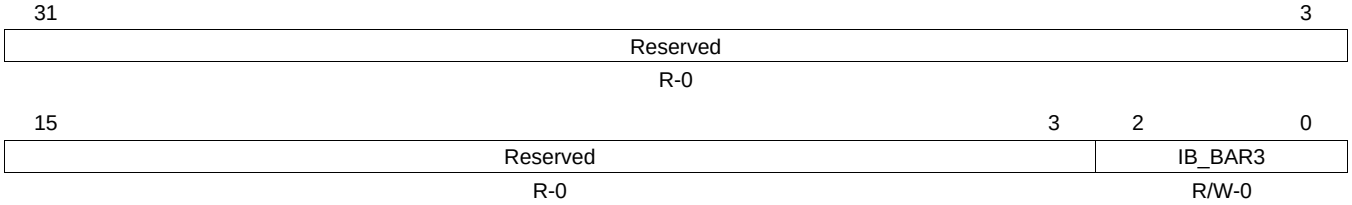
**Table 19-63. IB_OFFSET2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | IB_OFFSET2 | 0-FF FFFFh | Offset address bits 31-8 for inbound translation region 2 |
| 7-0 | Reserved | 0 | Reserved |

### 19.5.1.54 IB_BAR3 Register

The inbound translation bar match 3 register (B_BAR3) is described in the figure and table below.

**Figure 19-58. IB_BAR3 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

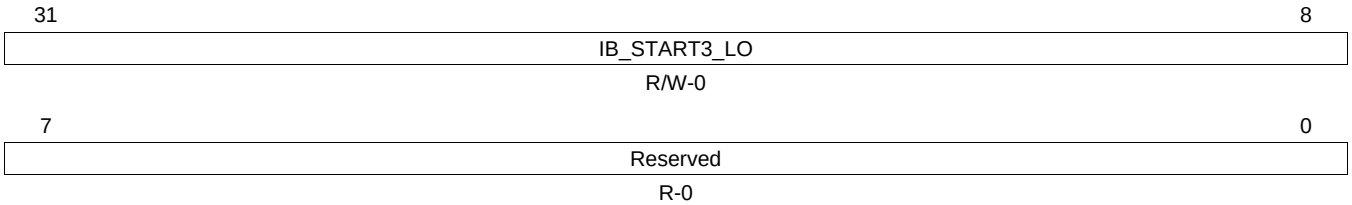
**Table 19-64. IB_BAR3 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2-0 | IB_BAR3 | 0-7h | BAR number to match for inbound translation region 3 |

### 19.5.1.55 IB_START3_LO Register

The inbound translation 3 start address low register (IB_START3_LO) is described in the figure and table below.

**Figure 19-59. IB_START3_LO Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-65. IB_START3_LO Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | IB_START3_LO | 0-FF FFFFh | Start address bits 31-8 for inbound translation region 3 |
| 7-0 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.1.56 IB_START3_HI Register

The inbound translation 3 start address high register (IB_START3_HI) is described in the figure and table below.

**Figure 19-60. IB_START3_HI Register**

| 31 | 0 |
|---|---|
| IB_START3_HI | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-66. IB_START3_HI Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | IB_START3_HI | 0-FFFF FFFFh | Start address bits 63-0 for inbound translation region 3 |

### 19.5.1.57 IB_OFFSET3 Register

The inbound translation 3 address offset register (IB_OFFSET3) is described in the figure and table below.

**Figure 19-61. IB_OFFSET3 Register**

| 31 | 8 |
|---|---|
| IB_OFFSET3 | |
| R/W-0 | |

| 7 | 0 |
|---|---|
| Reserved | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-67. IB_OFFSET3 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | IB_OFFSET3 | 0-FF FFFFh | Offset address bits 31-8 for inbound translation region 3 |
| 7-0 | Reserved | 0 | Reserved |

### 19.5.1.58 PCS_CFG0

The PCS configuration 0 (PCS_CFG0) is described in the figure and table below.

**Figure 19-62. PCS_CFG0 Register**

| 31 | 29 | 28 | 24 | 23 | 16 |
|---|---|---|---|---|---|
| Reserved | | PCS_SYNC | | PCS_HOLDOFF | |
| R/W-0 | | R/W-0 | | R/W-0 | |

| 15 | 14 | 13 | 12 | 11 | 8 |
|---|---|---|---|---|---|
| Reserved | | PCS_RC_DELAY | | PCS_DET_DELAY | |
| R-0 | | R/W-2h | | R/W-1 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PCS_SHRT_TM | PCS_STAT186 | PCS_FIX_TERM | PCS_FIX_STD | PCS_L2_ENIDL_OFF | PCS_L0S_RX_OFF | PCS_RXTX_ON | PCS_RXTX_RST |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-68. PCS_CFG0 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-29 | Reserved | 0 | Reserved |
| 28-24 | PCS_SYNC | 0-1Fh | Receiver Lock/Sync Control |
| 23-16 | PCS_HOLDOFF | 0-FFh | Receiver Initialization Hold Off Control |
| 15-14 | Reserved | 0 | Reserved |
| 13-12 | PCS_RC_DELAY | 0-3h | Rate Change Delay |
| 11-8 | PCS_DET_DELAY | 0-Fh | Detection Delay |
| 7 | PCS_SHRT_TM | 0 | Enable short times for debug purposes. |
| 6 | PCS_STAT186 | 0 | Enable PIPE Spec 1.86 for PHY status behavior |
| 5 | PCS_FIX_TERM | 0 | Fed term output to 3'b100 during reset. |
| 4 | PCS_FIX_STD | 0 | Fix std output to 2'b10 |
| 3 | PCS_L2_ENIDL_OFF | 0 | Deassert ENIDL during L2 state. |
| 2 | PCS_L0S_RX_OFF | 0 | Deassert Rx Enable in L0s state. |
| 1 | PCS_RXTX_ON | 0 | RX and TX on during reset. TX also on in P1 state. |
| 0 | PCS_RXTX_RST | 0 | RX and TX on during reset |

### 19.5.1.59 PCS_CFG1

The PCS configuration 1 (PCS_CFG1) is described in the figure and table below.

**Figure 19-63. PCS_CFG1 Register**

| 31 | 26 | 25 | 16 |
|---|---|---|---|
| Reserved | | PCS_ERR_BIT | |
| R/W-0 | | R/W-0 | |

| 15 | 10 | 9 | 8 | 7 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | PCS_ERR_LN | | Reserved | | PCS_ERR_MODE | |
| R/W-0 | | R/W-0 | | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 19-69. PCS_CFG1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-26 | Reserved | 0 | Reserved |
| 25-16 | PCS_ERR_BIT | 0-3FFh | Error Bit enable |
| 15-10 | Reserved | 0 | Reserved |
| 9-8 | PCS_ERR_LN | 0-3h | Error Lane enable |
| 7-2 | Reserved | 0 | Reserved |
| 1-0 | PCS_ERR_MODE | 0-3h | Error Injection Mode |

### 19.5.1.60 PCS_STATUS

The PCS status register is described in the figure and table below.

**Figure 19-64. PCS_STATUS Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 14 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rsvd | PCS_REV | | Rsvd | | PCS_LN_EN | | Rsvd | | PCS_TX_EN | | Rsvd | | PCS_RX_EN | |
| R-0 | R-1 | | R-0 | | R-3h | | R-0 | | R-0 | | R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-70. PCS_STATUS Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved |
| 14-12 | PCS_REV | 0-7h | PCS RTL Revision |
| 11-10 | Reserved | 0 | Reserved |
| 9-8 | PCS_LN_EN | 0-3h | PCS Lanes enabled status |
| 7-6 | Reserved | 0 | Reserved |
| 5-4 | PCS_TX_EN | 0-3h | PCS Transmitters enabled status |
| 3-2 | Reserved | 0 | Reserved |
| 1-0 | PCS_RX_EN | 0-3h | PCS Receivers enabled status |

### 19.5.1.61 SERDES_STATUS

The SerDes status register (SERDES_STATUS) is described in the figure and table below.

**Figure 19-65. SERDES_STATUS Register**

| 31 | 16 | 15 | 0 |
|---|---|---|---|
| STSTX | | SRSRX | |
| R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

### 19.5.1.62 SERDES_RXCFG0

The SerDes RX Config 0 status register (SERDES_RXCFG0) is described in the figure and table below.

**Figure 19-66. SERDES_RXCFG0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| LOOPBACK | | RX_TRIM_BYPASS | CDR_ELV_IDLE_FIX | CDRAUX | | CAL_FILTER_DEPTH | |
| R/W-0x0 | | R/W-0 | R/W-0 | R/W-0x0 | | R/W-0x0 | |

| 23 | 22 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|
| ENOC | EQ | | CDR | | |
| R/W-0 | R/W-0x0 | | R/W-0x0 | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 8 |
|---|---|---|---|---|---|---|
| LOS | | | ALIGN | | TERM | |
| R/W-0x0 | | | R/W-0x0 | | R/W-0x0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| INVPAIR | RATE | | BUSWIDTH | | | ENRXLDO | ENRX |
| R/W-0 | R/W-0x0 | | R/W-0x0 | | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-71. SERDES_RXCFG0 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | LOOPBACK | 10/11 | Internal Loopback with following options<br>CFGRX_1[14] Loop back from TX (pre-driver digital output) to RX (input of RXDIG20). Digital loop back.<br>CFGRX_1[14] Loop back from TX (pre-driver digital output) to RX (input of equalizer). Analog loop back. |
| | | 0 | Loopback disabled |
| | | 1 | Not supported |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-71. SERDES_RXCFG0 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 29 | RX_TRIM_BYPASS | | For calibration --> Bypasses the trim bits generated from the calibration algorithm |
| | | 0 | Use the trim code generated by trx_dig |
| | | 1 | Bypass the trim code generated by trx_dig |
| 28 | CDR_ELV_IDLE_FIX | 0 | CDR ELV Idle fix is enabled |
| | | 1 | CDR ELV Idle fix is disabled |
| 27-26 | CDRAUX | | CDRAUX[1:0]: Clock/data recovery auxilliary. Along with the CDR field, configures the clock/data recovery algorithm. |
| 25-24 | CAL_FILTER_DEPTH | | For calibration --> Average engine can be programmed to have depth of 7,15, 31 |
| | | 00 | Depth of 7 samples |
| | | 01 | Depth of 15 samples |
| | | 10 | Depth of 31 samples |
| | | 11 | Depth of 7 samples |
| 23 | ENOC | | ENOC: Enable offset compensation. Enables samplers offset compensation. Can be written DYNAMIC |
| | | 0 | |
| | | 1 | |
| 22-19 | EQ | | Equalizer. Enables and configures the adaptive equalizer to compensate for loss in the transmission media |
| | | 0 | Equalizer enabled |
| | | 1 | Equalizer disabled |
| 18-16 | CDR | | Clock/data recovery. Along with the CDRAUX field, configures the clock/data recovery algorithm. |
| | | 0 | |
| | | 1 | |
| 15-13 | LOS | | Enables loss of signal detection |
| | | 0000 | LOS disabled |
| | | 100 | LOS enabled, without CDR override control [CDR enable] |
| | | 110 | LOS enabled, with CDR override control [CDR disable] |
| | | 001 | LOS disabled |
| | | 01x | LOS disabled |
| | | 101 | LOS disabled |
| | | 11x | -LOS disabled |
| 12-11 | ALIGN | | Symbol alignment. Enables internal or external symbol alignment |
| | | 00 | Alignment disabled |
| | | 01 | Comma alignment enabled |
| | | 10 | Alignment Jog |
| | | 11 | Reserved |
| 10-8 | TERM | | Selects input termination options. External AC and DC coupled modes supported. |
| | | 000 | Reserved |
| | | 001 | Common point set to 0.8 VDDA. This configuration is for AC coupled systems |
| | | 010 | Reserved |
| | | 011 | Common point floating. This configuration is for DC coupled systems which require the common mode voltage to be determined by the transmitter only. |
| | | 100 | Common point set to VSSA. This configuration is for PCI- Express and USB3.0 |
| | | 101 | Common point set to 0.2 VDDA. (For SATA system). |
| | | 110 | Reserved |
| | | 111 | Common point floating, wide common mode range. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-71. SERDES_RXCFG0 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7 | INVPAIR | | Inverts polarity of RXPi and RXNi. |
| | | 0 | Normal Polarity. RXP considered to be positive |
| | | 1 | Inverted Polarity. RXN considered to be positive |
| 6-5 | RATE | | Operating rate. Selects full, full, half, quarter or twentieth rate operation |
| | | 00 | Full rate |
| | | 01 | Half rate |
| | | 1x | Reserved |
| | | | Bit 5 ONLY can be written DYNAMIC. |
| 4-2 | BUSWIDTH | | Selects the parallel interface width (8 or 10 bit). |
| | | | Only 10-bit mode will be used. 8-bit mode not officially supported. |
| | | 000 | 10-bit operation |
| | | 001 | Reserved |
| | | 01x | Reserved |
| | | 1xx | Reserved |
| 1 | ENRXLDO | | Enables RXLDO when high. When ENRXLDO='0', RX Analog circuits will be powered down. RXLDO has to be enabled and allowed to settle prior to enabling the receiver with ENRX. |
| | | 0 | |
| | | 1 | |
| 0 | ENRX | | Enables receiver when high. When ENRX='0', all digital circuitry will be disabled and clocks are gated. All current sources in analog will be powered down with the exception of those associated with LOS detector and IEEE 1149.6 boundary scan comparators |
| | | 0 | |
| | | 1 | |

### 19.5.1.63 SERDES_RXCFG1

The SerDes RX Config 1 status register (SERDES_RXCFG1) is described in the figure and table below.

**Figure 19-67. SERDES_RXCFG1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| EQ_ICM_S2 | | EQ_ICM_S1 | | RESERVED_RX1_27_24 | | | |
| R/W-0x0 | | R/W-0x0 | | R/W-0x0 | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EQ_I_STAGE2 | | | EQ_I_STAGEFB | | | EQ_I_STAGE1 | |
| R/W-0x0 | | | R/W-0x0 | | | R/W-0x0 | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| EQ_I_STAGE1 | ANALOG_LOOPBACK | RXTRIM_CALIB | RXTRIM_BYPASS_CTRL | RXTRIM_BYPASS_BITS | | | |
| R/W-0x0 | R/W-0 | R/W-0 | R/W-0 | R/W-0x0 | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXTRIM_BYPASS_BITS | BYPASS_CALOUT_AVG | CTG_DIG_RSVD1 | CTG_DIG_RSVD0 | ENTEST | TESTPATT | | |
| R/W-0x0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0x0 | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-72. SERDES_RXCFG1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | EQ_ICM_S2 | | Trims common mode pullup current in second equalizer stage |
| | | 0 | |
| | | 1 | |
| 29-28 | EQ_ICM_S1 | | Trims common mode pullup current in first equalizer stage |
| | | 0 | |
| | | 1 | |
| 27-24 | RESERVED_RX1_27_24 | | Reserved for future use |
| | | 0 | |
| | | 1 | |
| 23-21 | EQ_I_STAGE2 | | Trims the current in the second stage of the equalizer |
| | | 0 | |
| | | 1 | |
| 20-18 | EQ_I_STAGEFB | | Trims the current in the feedback stage of the equalizer |
| | | 0 | |
| | | 1 | |
| 17-15 | EQ_I_STAGE1 | | Trims the current in the first stage of the equalizer |
| | | 0 | |
| | | 1 | |
| 14 | ANALOG_LOOPBACK | | Enables analog loop back |
| | | 0 | |
| | | 1 | |
| 13 | RXTRIM_CALIB | | RX trimming control to the calibration block |
| | | 0 | In functioanl mode |
| | | 1 | eFuse training mode |
| 12 | RXTRIM_BYPASS_CTRL | 0 | res_nwell settings are controlled by fuse/calib settingsRXTRIM[4:0] |
| | | 1 | res_nwell settings are controlled by cfg_ctrl[11:7](CFGRX_1[11:7]) |
| 11-7 | RXTRIM_BYPASS_BITS | | Bypass bits for RXTRIM[4:0] when CFGRX_1[12]=1 |
| | | 0 | |
| | | 1 | |
| 6 | BYPASS_CALOUT_AVG | | For calibration --> Bypasses the averaging filter for the calout signal inside digital |
| | | 0 | Use the output of the digital filter for CALOUT |
| | | 1 | Ignore the output of the digital filter and use the output from the bubble removal circuit for CALOUT |
| 5 | CTG_DIG_RSVD1 | | Clock inversion for rpclk. In X2256, this was mapped to ctl_dig_rsvd[1]. This is a temporary mapping. |
| | | 0 | |
| | | 1 | |
| 4 | CTG_DIG_RSVD0 | | Clock inversion for fclk. In X2256, this was mapped to ctl_dig_rsvd[0]. This is a temporary mapping. |
| | | 0 | |
| | | 1 | |
| 3 | ENTEST | | Enable test. Enables test modes specified via LOOPBACK (CFGRX_0[31:30]). |
| | | 0 | |
| | | 1 | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-72. SERDES_RXCFG1 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 2-0 | TESTPATT | | Enables and selects test patterns. |
| | | 000 | Reserved |
| | | 100 | Uses a 31-bit LFSR with feedback polynomial x31+x28+1. |
| | | 001 | An alternating 0/1 pattern with a period of 2 UI. |
| | | 010 | Uses a 7-bit LFSR with feedback polynomial x7+x6+1. |
| | | 011 | Uses a 23-bit LFSR with feedback polynomial x23+x18+1 |
| | | 101 | Reserved |
| | | 110 | Reserved |
| | | 111 | Reserved |

### 19.5.1.64 SERDES_RXCFG2

The SerDes RX Config 2 status register (SERDES_RXCFG2) is described in the figure and table below.

**Figure 19-68. SERDES_RXCFG2 Register**

| 31 27 | 26 22 | 21 17 | 16 |
|---|---|---|---|
| SAMP_BYPASS_SAD1 | SAMP_BYPASS_SAT0 | SAMP_BYPASS_SAD0 | SAMP_BYPASS_SAT1 |
| R/W-0x00 | R/W-0x00 | R/W-0x00 | R/W-0x00 |

| 15 12 | 11 10 | 9 8 |
|---|---|---|
| SAMP_BYPASS_SAT1 | RESERVED_RX2_11_0 | |
| R/W-0x00 | R/W-0x00 | |

| 7 0 |
|---|
| RESERVED_RX2_11_0 |
| R/W-0x000 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-73. SERDES_RXCFG2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-27 | SAMP_BYPASS_SAD1 | | Bypass bits for SAD1[4:0] when CFG_CTRL[94](CFGRX_3[30]) = 1 |
| | | 0 | |
| | | 1 | |
| 26-22 | SAMP_BYPASS_SAT0 | | Bypass bits for SAT0[4:0] when CFG_CTRL[94](CFGRX_3[30]) = 1 |
| | | 0 | |
| | | 1 | |
| 21-17 | SAMP_BYPASS_SAD0 | | Bypass bits for SAD0[4:0] when CFG_CTRL[94](CFGRX_3[30]) = 1 |
| | | 0 | |
| | | 1 | |
| 16-12 | SAMP_BYPASS_SAT1 | | Bypass bits for SAT1[4:0] when CFG_CTRL[94](CFGRX_3[30]) = 1 |
| | | 0 | |
| | | 1 | |
| 11-0 | RESERVED_RX2_11_0 | | Reserved for future use |
| | | 0 | |
| | | 1 | |

### 19.5.1.65 SERDES_RXCFG3

The SerDes RX Config 3 status register (SERDES_RXCFG3) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-69. SERDES_RXCFG3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| AMUX_EYESCAN_REF | SAMP_OC_SEL | SAMP_ES_VREF_BYPASS_BITS | | | | | |
| R/W-0 | R/W-0 | R/W-0x00 | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| SAMP_ESCM_RES | | | | SAMP_ESCM_I | | | SAMP_3_VREF_2_ES |
| R/W-0x0 | | | | R/W-0 | | | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SAMP_2_VREF_2_ES | SAMP_1_VREF_2_ES | SAMP_0_VREF_2_ES | SAMP_IBIAS_Z | | | | |
| R/W-0 | R/W-0 | R/W-0 | R/W-0x00 | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SAMP_ES_VREF_BYPASS_CTRL | RESERVED_RX3_6_4 | | | SAMP_EN_3_ODD | SAMP_EN_2_EOTR | SAMP_EN_1_EVEN | SAMP_EN_0_OETR |
| R/W-0 | R/W-0x0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-74. SERDES_RXCFG3 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | AMUX_EYESCAN_REF | | =1 Connect the eyescan reference voltages to the test mux inputs (used with amuxsel_rx[3:0]) |
| | | 0 | |
| | | 1 | |
| 30 | SAMP_OC_SEL | 0 | =0 Sampler offset correction is controlled via the TRXDIG SAT1, SAD0, SAT0, SAD1 |
| | | 1 | =1 Sampler offset correction is controlled via the configuration register (CFG_CTRL[63:44])(CFGRX_2[31:12]) |
| 29-24 | SAMP_ES_VREF_BYPASS_BITS | | Adjusts the eyescan reference voltage level, which introduces linearly decreasing offset with code |
| | | 0 | |
| | | 1 | |
| 23-20 | SAMP_ESCM_RES | | Trims the pullup resistors in the eyescan common mode loop; roughly should track CFGRX_1[10:7] |
| | | 0 | |
| | | 1 | |
| 19-17 | SAMP_ESCM_I | | Trims the current in the eyescan common mode loop; roughly should track CFG_CTRL[23:21](CFGRX_1[23:21]) |
| | | 0 | |
| | | 1 | |
| 16 | SAMP_3_VREF_2_ES | 0 | =0 reference voltages are EQCM |
| | | 1 | =1 Connect reference voltage of sampler 3 to eyescan |
| 15 | SAMP_2_VREF_2_ES | 0 | =0 reference voltages are EQCM |
| | | 1 | =1 Connect reference voltage of sampler 2 to eyescan |
| 14 | SAMP_1_VREF_2_ES | 0 | =0 reference voltages are EQCM |
| | | 1 | =1 Connect reference voltage of sampler 1 to eyescan |
| 13 | SAMP_0_VREF_2_ES | 0 | =0 reference voltages are EQCM |
| | | 1 | =1 Connect reference voltage of sampler 0 to eyescan |
| 12-8 | SAMP_IBIAS_Z | | Adjusts the current mirror ratio in the sampler bias current block. Lower numbers correspond to higher bias current. |
| | | 0 | |
| | | 1 | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-74. SERDES_RXCFG3 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7 | SAMP_ES_VREF_BYPASS_CTRL | 0<br>1 | =0 Eyescan reference offset voltage controled through IEEE1500<br>=1 Eyescan reference offset voltage controled through CFGRX_3[29:24] |
| 6-4 | RESERVED_RX3_6_4 | 0<br>1 | UNUSED |
| 3 | SAMP_EN_3_ODD | 0<br>1 | =1 Enable sampler 3 (odd data) |
| 2 | SAMP_EN_2_EOTR | 0<br>1 | =1 Enable sampler 2 (even to odd transition) |
| 1 | SAMP_EN_1_EVEN | 0<br>1 | =1 Enable sampler 1 (even data) |
| 0 | SAMP_EN_0_OETR | 0<br>1 | =1 Enable sampler 0 (odd to even transition) |

### 19.5.1.66 SERDES_RXCFG4

The SerDes RX Config 4 status register (SERDES_RXCFG4) is described in the figure and table below.

**Figure 19-70. SERDES_RXCFG4 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED_RX4_31 | RESERVED_RX4_30 | RCLK_SAMP | | RCLK_DIG | | RESERVED_RX4_25 | DCD_EN_BUF |
| R/W-0 | R/W-0 | R/W-0x0 | | R/W-0x0 | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DCD_EN_CLK | DCD_EN_M | DCD_EN_P | RESERVED_RX4_20 | PI_VCM_Z_0 | PI_VCM_1 | PI_VCM_Z_2 | PI_VCM_Z_3 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PI_BIAS_DISABLE | PI_ED_CAL_LEVEL_DEC | PI_ED_CAL_LEVEL_INC | PI_ED_CAL | PI_ED_RESET | PI_ED_EN | PI_ED_VOUTP | PI_ED_VOUTM |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PI_I50U | PI_I100U_Z | PI_IBIAS_Z | | | | | |
| R/W-0 | R/W-0 | R/W-0x00 | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-75. SERDES_RXCFG4 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | RESERVED_RX4_31 | 0<br>1 | UNUSED |
| 30 | RESERVED_RX4_30 | 0<br>1 | UNUSED |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-75. SERDES_RXCFG4 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 29-28 | RCLK_SAMP | | RCLK_SAMP[1:0] |
| | | 0 | |
| | | 1 | |
| 27-26 | RCLK_DIG | | RCLK_DIG[1:0] |
| | | 0 | |
| | | 1 | |
| 25 | RESERVED_RX4_25 | | UNUSED |
| | | 0 | |
| | | 1 | |
| 24 | DCD_EN_BUF | | DCD_EN_BUF |
| | | 0 | |
| | | 1 | |
| 23 | DCD_EN_CLK | | DCD_EN_CLK |
| | | 0 | |
| | | 1 | |
| 22 | DCD_EN_M | | DCD_EN_M |
| | | 0 | |
| | | 1 | |
| 21 | DCD_EN_P | | DCD_EN_P |
| | | 0 | |
| | | 1 | |
| 20 | RESERVED_RX4_20 | | UNUSED |
| | | 0 | |
| | | 1 | |
| 19 | PI_VCM_Z_0 | | PI_VCM_Z[0] |
| | | 0 | |
| | | 1 | |
| 18 | PI_VCM_1 | | PI_VCM[1] |
| | | 0 | |
| | | 1 | |
| 17 | PI_VCM_Z_2 | | PI_VCM_Z[2] |
| | | 0 | |
| | | 1 | |
| 16 | PI_VCM_Z_3 | | PI_VCM_Z[3] |
| | | 0 | |
| | | 1 | |
| 15 | PI_BIAS_DISABLE | | PI_BIAS_DISABLE |
| | | 0 | |
| | | 1 | |
| 14 | PI_ED_CAL_LEVEL_DEC | | PI_ED_CAL_LEVEL_DEC |
| | | 0 | |
| | | 1 | |
| 13 | PI_ED_CAL_LEVEL_INC | | PI_ED_CAL_LEVEL_INC |
| | | 0 | |
| | | 1 | |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-75. SERDES_RXCFG4 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 12 | PI_ED_CAL | | PI_ED_CAL |
| | | 0 | |
| | | 1 | |
| 11 | PI_ED_RESET | | PI_ED_RESET |
| | | 0 | |
| | | 1 | |
| 10 | PI_ED_EN | | PI_ED_EN |
| | | 0 | |
| | | 1 | |
| 9 | PI_ED_VOUTP | | PI_ED_VOUTP |
| | | 0 | |
| | | 1 | |
| 8 | PI_ED_VOUTM | | PI_ED_VOUTM |
| | | 0 | |
| | | 1 | |
| 7 | PI_I50U | | PI_I50U |
| | | 0 | |
| | | 1 | |
| 6 | PI_I100U_Z | | PI_I100U_Z |
| | | 0 | |
| | | 1 | |
| 5-0 | PI_IBIAS_Z | | PI_IBIAS_Z[5:0] |
| | | 0 | |
| | | 1 | |

### 19.5.1.67 SERDES_TXCFG0 ( AND CHECK VALUES)

The SerDes TX Config 0 register (SERDES_TXCFG0) is described in the figure and table below.

**Figure 19-71. SERDES_TXCFG0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| LOOPBACK | | CAL_FILTER_DEPTH | | RESERVED | DET_CTL | | ENIDL |
| R/W-0x0 | | R/W-0x0 | | R/W-0 | R/W-0x0 | | R/W-0 |

| 23 | 22 | 21 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|
| RESERVED | | TM_EXTRA_LOAD | | RESERVED | DEEMP | |
| R/W-0 | | R/W-0x0 | | R/W-0 | R/W-0x00 | |

| 15 | 14 | 13 | 12 | 9 | 8 |
|---|---|---|---|---|---|
| DEEMP | | | SWING | | RESERVED |
| R/W-0x00 | | | R/W-0x0 | | R/W-0 |

| 7 | 6 | 5 | 4 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| INVPAIR | RATE | | BUSWIDTH | | ENTXLDO | ENTX |
| R/W-0 | R/W-0x0 | | R/W-0x0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-76. SERDES_TXCFG0 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | LOOPBACK | | Not Used. The RX has similar settings, hence this field is not used. |
| 29-28 | CAL_FILTER_DEPTH | | For calibration --> Average engine can be programmed to have depth of 7,15, 31. |
| | | 00 | Depth of 7 samples |
| | | 01 | Depth of 15 samples |
| | | 10 | Depth of 31 samples |
| | | 11 | Depth of 7 samples |
| 27 | RESERVED | | Reserved |
| 26-25 | DET_CTL | | Receiver detect control. Controls PCI-E receiver detect. |
| 24 | ENIDL | | Idle Control. Enables transmission of electricle idle |
| | | 0 | Disabled |
| | | 1 | Enabled |
| 23-22 | RESERVED | | Reserved - drive low |
| 21-19 | TM_EXTRA_LOAD | | TestMode. Default is 000 : auto program dummy load based on mode. |
| | | 0 | [19] -> '1': Enable different extraload resistors to make driver constant current at speed. Current level defined in CFGTX_2[26:25] |
| | | 1 | [20] -> '1': Enable static dummy load during electrical idle. Load current level defined in CFGTX_2[28:27] |
| | | | [21] -> '1': Enable switching dummy load during electrical idle. Load current level defined in CFGTX_2[30:29] |
| 18 | RESERVED | | Reserved- drive low |
| 17-13 | DEEMP | | De-emphasis. Selects one of output de-emphasis settings from 0 to 50.2% |
| | | | For PCIE |
| | | | Value --> Amplitude Reduction (%) --> Amplitude Reduction (dB) |
| | | | 00000 --> 0 --> 0 |
| | | | 00001 to 01111 --> 33.7 --> -3.57 |
| | | | 10000 to 11111 --> 50.2 --> -6.05 |
| | | | For eSATA - FULL RATE (CFGTX_0[6:5]=00) |
| | | | Value --> Amplitude Reduction (%) --> Amplitude Reduction (dB) |
| | | | 00000 --> 0 --> 0 |
| | | | 00001 to 01111 --> 10 --> -1 |
| | | | 10000 to 11111 --> 25 --> -2.5 |
| | | | For eSATA - HALF RATE (CFGTX_0[6:5]=01) |
| | | | Value --> Amplitude Reduction (%) --> Amplitude Reduction (dB) |
| | | | 00000 --> 0 --> 0 |
| | | | 00001 to 01111 --> 10 --> -1 |
| | | | 10000 to 11111 --> RESERVED --> RESERVED |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-76. SERDES_TXCFG0 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 12-9 | SWING | | TX Output swing selection. Selects one of output amplitude settings.<br>For PCIE<br>Value --> Amplitude (mVdfpp) --> ANA {swingsel,swing[1:0]}<br>0000 --> 148 (half) --> 111<br>0001 --> 148 (half) --> 111<br>0010 --> 258 (half) --> 110<br>0011 --> 258 (half) --> 110<br>0100 --> 295 (full/half) --> 101<br>0101 --> 295 (full/half) --> 101<br>0110 --> 516 (full/half) --> 100<br>0111 --> 516 (full/half) --> 100<br>1000 --> 774 (full/half) --> 001<br>1001 --> 774 (full/half) --> 001<br>1010 --> 1069 (full/half) --> 000<br>1011 --> 1069 (full/half) --> 000<br>1100 --> 1069 (full/half) --> 000<br>1101 --> 1069 (full/half) --> 000<br>1110 --> 1069 (full/half) --> 000<br>1111 --> 1069 (full/half) --> 000<br>For eSATA<br>Only 516mV (for Gen1) and 590mV (for Gen2) swing value is supported in SATA mode and all the codes above will map to this in TXA |
| 8 | RESERVED | | Reserved |
| 7 | INVPAIR | | Inverts polarity of TXPi and TXNi. |
| | | 0 | Normal Polarity. TXP is positive data |
| | | 1 | Inverted Polarity. TXN is positive data |
| 6-5 | RATE | | Operating rate. Selects full, full, half, quarter or twentieth rate operation. |
| | | 00 | Full Rate |
| | | 01 | Half Rate |
| | | 1x | Reserved |
| 4-2 | BUSWIDTH | | Selects the parallel interface width (8 or 10 bit).<br>Only 10-bit mode will be used. 8-bit mode not officially supported. |
| | | 000 | 10-bit operation |
| | | 001 | Reserved |
| | | 01x | Reserved |
| | | 1xx | Reserved |
| 1 | ENTXLDO | | Enables TX Analog LDO when high. When ENTXLDO='0', TX Analog circuits will be powered down. TXLDO has to be enabled and allowed to settle prior to enabling the transmitter with ENTX. |
| | | 0 | Disabled |
| | | 1 | Enabled |
| 0 | ENTX | | Enables this transmitter when high. |
| | | 0 | Disabled |
| | | 1 | Enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.1.68 SERDES_TXCFG1

The SerDes TX Config 1 register (SERDES_TXCFG1) is described in the figure and table below.

**Figure 19-72. SERDES_TXCFG1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TRIM_HIGH_THRESHOLD | | | | TX_DISABLE_ON_THE_FLY | TX_FORCE_UPDATE | TX_TRIM_BYPASS | TRIM_2B_MODE |
| R/W-0X0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| RESERVED | TRIM_STEP_CHANGE | | TRIM_LOW_THRESHOLD | | CALIB_WAIT_CYCLES | | ENABLE_LOOP_DELAY |
| R/W-0x0 | | | R/W-0x0 | | R/W-0x0 | | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| BYPASS_CALOUT_AVG | NOLCKRST | ENBSPLS | ENBSRX | ENBSTX | BSINITCLK | BSINRXN | BSINRXP |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TSYNC_ENABLE | ENTEST | BSTX | TESTPATT | | | USE_STAIRCASE | ENLFPS |
| R/W-0 | R/W-0 | R/W-0 | R/W-0x0 | | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-77. SERDES_TXCFG1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | TRIM_HIGH_THRESHOLD | | For calibration --> 4-b programmable threshold<br>Unless the difference between the previous and new trim code required exceeds 4-b threshold, trim bit will be updated in one shot. Otherwise the trim bits will be updated in steps determined by TRIM_STEP_CHANGE. |
| 27 | TX_DISABLE_ON_THE_FLY | | For calibration --> Disables the pattern match. Updates can only be made through electrical idle or TX_FORCE_UPDATE bit. |
| | | 0 | Enables pattern match |
| | | 1 | Disables pattern match |
| 26 | TX_FORCE_UPDATE | | For calibration --> Forces the update to be made to the Analog trim bits without looking for any specific data pattern or electrical idle. |
| | | 0 | Disables forced update. |
| | | 1 | Enables forced update |
| | | | Can be written DYNAMIC. |
| 25 | TX_TRIM_BYPASS | | For calibration --> Bypasses the trim bits generated from the calibration algorithm |
| | | 0 | Use the trim code generated by trx_dig |
| | | 1 | Bypass the trim code generated by trx_dig |
| 24 | TRIM_2B_MODE | | For calibration --> Enables 2-b data pattern detection mode for trim updates. |
| | | 0 | 2-b detection mode, e.g. "00" or "11" |
| | | 1 | 3-b detection mode, e.g. "00" or "111" |
| 23 | RESERVED | | Reserved |
| 22-21 | TRIM_STEP_CHANGE | | For calibration --> Step change value for staircase logic |
| | | 00 | make updates in steps of 1 |
| | | 01 | make updates in steps of 2 |
| | | 10 | make updates in steps of 4 |
| | | 11 | make updates in steps of 8 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-77. SERDES_TXCFG1 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 20-19 | TRIM_LOW_THRESHOLD | | For calibration --> 2-b programmable threshold. |
| | | 00 | make updates only when diff is >= 1 |
| | | 01 | make updates only when diff is >= 2 |
| | | 10 | make updates only when diff is >= 3 |
| | | 11 | make updates only when diff is >= 4 |
| | | | Unless the difference between the previous and current trim code exceeds or equals above mentioned threshold, no updates trim bit updates are made in Analog. |
| 18-17 | CALIB_WAIT_CYCLES | | For calibration --> 2-b programmable number of wait cycles before polling the callout signal |
| | | 0 | Additional clock cycle delay of 1 MHz clock |
| | | 1 | Additional clock cycle delay of 1 MHz clock |
| | | 2 | Additional clock cycle delay of 1 MHz clock |
| | | 3 | Additional clock cycle delay of 1 MHz clock |
| 16 | ENABLE_LOOP_DELAY | | For calibration --> Enables the additional loop delay of 40 clock cycles before sweeping the trim code during the calibration |
| | | 0 | Disables the loop delay |
| | | 1 | Enables the loop delay, ensures at least 32-clock cycles between two full cycles of trim code sweep |
| 15 | BYPASS_CALOUT_AVG | | For calibration --> Bypasses the averaging filter for the callout signal inside digital |
| | | 0 | Use the output of the digital filter for CALOUT |
| | | 1 | Ignore the output of the digital filter and use the output from the bubble removal circuit for CALOUT. |
| 14 | NOLCKRST | | Reset override bit during PLL unlock. Only for internal use. Not for SOC. |
| 13 | ENBSPLS | | Receiver pulse boundary scan. Configures IEEE 1149.6 boundary scan comparators connected to RXP and RXN for EXTEST_PULSE and EXTEST_TRAIN when high and EXTEST when low |
| 12 | ENBSRX | | Enables IEEE 1149.6 boundary scan control of RXP and RXN from the BSRX bit |
| 11 | ENBSTX | | Enables IEEE 1149.6 boundary scan control of TXP and TXN from the BSTX bit |
| 10 | BSINITCLK | | BS initialization clock |
| 9 | BSINRXN | | Boundary scan initialization for RXN. The IEEE1149.6 boundary scan comparator attached to RXN is initialized to the value of this bit on the rising edge of BSINITCLK |
| 8 | BSINRXP | | Boundary scan initialization for RXP. The IEEE1149.6 boundary scan comparator attached to RXP is initialized to the value of this bit on the rising edge of BSINITCLK |
| 7 | TSYNC_ENABLE | | In x2256: was mapped to CFGTX_0 [21] |
| | | | In x2282: DO NOT USE |
| 6 | ENTEST | | Enable test. Enables test modes specified via LOOPBACK (CFGRX_0[31:30]). |
| 5 | BSTX | | Boundary scan data. Determines the logic level output on TXP and TXN when boundary scan outputs are enabled. |
| 4-2 | TESTPATT | | Not Used. The RX has similar settings, hence this field is not used. |
| 1 | USE_STAIRCASE | | For calibration |
| | | 0 | Disables staircase logic for trim code change |
| | | 1 | Enables staircase logic for trim code change |
| 0 | ENLFPS | | DO NOT USE: Drive Low |

### 19.5.1.69 SERDES_TXCFG2

The SerDes TX Config 2 register (SERDES_TXCFG2) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-73. SERDES_TXCFG2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | TM_DUMMY2IDLE | | TM_DUMMY1IDLE | | TM_TEXTRALOAD | | DISABLE_FB2_UNUSED |
| R/W-0 | R/W-0X0 | | R/W-0x0 | | R/W-0x0 | | R/W-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DISABLE_FB1_TM_SRCONTROL | SC_SATA | | SC_PCIE | | DISABLE_PCIE_SC | DISABLE_SATA_SC | Reserved |
| R/W-0 | R/W-0x0 | | R/W-0x0 | | R/W-0 | R/W-0 | R/W-0 |
| | | | | | | | TMTRIM |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | TXDCC_PWRDN | Reserved | | | | | TMTRIM |
| R/W-0 | R/W-0 | R/W-0x00 | | | | | R/W-0x00 |
| 7 | | | 4 | 3 | 2 | 1 | 0 |
| TMTRIM | | | | TRIMBYPASS | RDTCT_VTMODE | | RESERVED |
| R/W-0x00 | | | | R/W-0x0 | R/W-0x0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-78. SERDES_TXCFG2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | | Reserved |
| 30-29 | TM_DUMMY2IDLE | | TestMode.Default is 00. If tm_dummy2_enz=1, then this controls ldo dummy digital load. |
| | | 00 | no resistive dummy load |
| | | 01 | 1.5 mA dummy load |
| | | 10 | 1.5 mA dummy load |
| | | 11 | 3mA dummy load |
| | | | Other settings: Extra load turned on |
| 28-27 | TM_DUMMY1IDLE | | TestMode.Default is 00. If tm_dummy1_enz=1, then this controls ldo dummy resistor load. |
| | | 00 | no resistive dummy load |
| | | 01 | 6 mA resistive dummy load |
| | | 10 | 9.6 mA resistive dummy load |
| | | 11 | 15mA resistive dummy load |
| | | | Other settings: Extra load turned on |
| 26-25 | TM_TEXTRALOAD | | TestMode: default is 00. If txa_tmel_enz=1, the driver dummy current controlled by tx_extraload. |
| | | 00 | Extra load current turned off |
| | | 01 | 1mA dummy load |
| | | 10 | 2 mA dummy load |
| | | 11 | 3 mA dummy load |
| 24 | DISABLE_FB2_UNUSED | | Reserved |
| 23 | DISABLE_FB1_TM_SRCONTROL | | TM_SRCONTROL |
| 22-21 | SC_SATA | | TestMode: Control for SATA slew-control. Default value is 00. |
| 20-19 | SC_PCIE | | TestMode: Control for PCIe slew-control. Default value is 00 |
| 18 | DISABLE_PCIE_SC | | TestMode: Default value for PCIE/SATA mode=0. |
| | | 0 | Enable output divider slew-rate control for PCIe mode (default) |
| | | 1 | Disable slew-rate control from digital and instead force from SC_PCIE |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-78. SERDES_TXCFG2 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 17 | DISABLE_SATA_SC | | TestMode. Automatically disabled in pcie mode. Default value for PCIE/SATA mode=0. |
| | | 0 | Enable output driver slew-rate control for SATA mode (default). Slew rate automatically adjusted based on RATE in SATA mode. |
| | | 1 | Disable slew-rate control from digital and instead force from SC_SATA |
| 16 | RESERVED | | Boundary scan initialization for RXN. The IEEE1149.6 boundary scan comparator attached to RXN is initialized to the value of this bit on the rising edge of BSINITCLK |
| 15 | RESERVED | | Boundary scan initialization for RXP. The IEEE1149.6 boundary scan comparator attached to RXP is initialized to the value of this bit on the rising edge of BSINITCLK |
| 14 | TXDCC_PWRDN | | TestMode. Default is 0. |
| | | 0 | TX clock will be duty-cycle corrected (default) |
| | | 1 | TX clock will bypass duty-cycle correction logic |
| 13-9 | RESERVED1 | | Reserved. Default is 00000 |
| 8-4 | TMTRIM | | TestMode: Termination resistor calibration code for grounded resistor. Default is 00000 |
| 3 | TRIMBYPASS | | TestMode: Default is 0. |
| | | 0 | TX Termination resistor calibrated through continuous digital loop (default) |
| | | 1 | Over-ride digital control of termination resistor with TMTRIM |
| 2-1 | RDTCT_VTMODE | | Receive detect test mode to program threshold. Default is 00. |
| | | 00 | .285V threshold |
| | | 01 | 0.3V threshold |
| | | 10 | .27V threshold |
| | | 11 | 0.35V (VBG_REF) |
| 0 | RESERVED | | Reserved |

### 19.5.1.70 SERDES_TXCFG3

The SerDes TX Config 3 register (SERDES_TXCFG3) is described in the figure and table below.

**Figure 19-74. SERDES_TXCFG3 Register**

| 31 | 28 | 27 | 16 |
|---|---|---|---|
| RESERVED | | RXLDO_CTRL | |
| R/W-0x0 | | R/W-0x0000 | |

| 15 | 12 | 11 | 10 | 1 | 0 |
|---|---|---|---|---|---|
| RXLDO_CTRL | | RXLDO_HITRAN_UNUSED | TXLDO_CTRL | | TXLDO_DISABLE_OVERSHOOT |
| R/W-0x0000 | | R/W-0 | R/W-0x000 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-79. SERDES_TXCFG3 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | RESERVED | | Reserved |
| 27-12 | RXLDO_CTRL | | TestMode: RX LDO control bits<br>[27:24] = 0111 (LDO loop compensation control)<br>[23] = 0 (When '1', overrides LDO UP signal to be always high)<br>[22] = 0 (Low dropout mode)<br>[21] = 1 ( Vref magnitude selection)<br>[20] = 1 ( Iref magnitude selection)<br>[19:18]= 11 (Quiescent current programmability)<br>[17] = 0 (disable overshoot control block)<br>[16:12]=01110 (default trim setting to get 1.2V) |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-79. SERDES_TXCFG3 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 11 | RXLDO_HITRAN_UNUSED | | TestMode: RX LDO transient improvement bit |
| 10-1 | TXLDO_CTRL | | TestMode:TX LDO control bits<br>[10:7] = 0111 (LDO loop compensation control)<br>[6] = 0 (When '1', overrides LDO UP signal to be always high)<br>[5] = 0 (Low dropout mode)<br>[4] = 1 ( Vref magnitude selection)<br>[3] = 1 ( Iref magnitude selection)<br>[2:1]= 11 (Quiescent current programmability) |
| 0 | TXLDO_DISABLE_OVERSHOOT | | TestMode: TX LDO disable overshoot control if 1. Default is 0. |

### 19.5.1.71 SERDES_TXCFG4

The SerDes TX Config 4 register (SERDES_TXCFG4) is described in the figure and table below.

**Figure 19-75. SERDES_TXCFG4 Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R/W-0x000 0000 | |

| 15 | 5 | 4 | 3 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | TM_BIASCTR | TRIM_MODE | | HYST_DISABLE |
| R/W-0x000 0000 | | R/W-0 | R/W-0X0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-80. SERDES_TXCFG4 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | | Reserved |
| 4 | TM_BIASCTR | | TM_BIASCTR used in DCC |
| 3-1 | TRIM_MODE | | TRIM_MODE[2:0] |
| 0 | HYST_DISABLE | | HYST_DISABLE_Z and CALOUT_TX |

## *19.5.2 Configuration Registers Common to Type 0 and Type 1 Headers*

**Table 19-81. Configuration Registers Common to Type 0 and Type 1 Headers**

| Offset | Acronym |
|---|---|
| 0h | VENDOR_DEVICE_ID |
| 4h | STATUS_COMMAND |
| 8h | CLASSCODE_REVID |

### 19.5.2.1 VENDOR_DEVICE_ID

The vendor and device identification register (VENDOR_DEVICE_ID) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-76. VENDOR_DEVICE_ID Register**

| 31 | 16 |
|---|---|
| DEVICE_ID | |
| R-8888h | |
| 15 | 0 |
| VENDOR_ID | |
| R-104Ch | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-82. VENDOR_DEVICE_ID Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | DEVICE_ID | 0-FFFFh | PCIe Vendor ID. Writable from internal bus interface. |
| 15-0 | VENDOR_ID | 0-FFFFh | PCIe Device ID. Writable from internal bus interface. |

### 19.5.2.2 STATUS_COMMAND Register

The status and command register (STATUS_COMMAND) is described in the figure and table below.

**Figure 19-77. STATUS_COMMAND Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Parity Error | Signaled System Error | Received Master Abort | Received Target Abort | Signaled Target Abort | Reserved | | Data Parity Error |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R-0 | | R/W1C-0 |

| 23 | 21 | 20 | 19 | 18 | 16 |
|---|---|---|---|---|---|
| Reserved | | Capabilities List | Interrupt Status | Reserved | |
| R-0 | | R-1 | R-0 | R-0 | |

| 15 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|
| Reserved | | INTx Disable | Reserved | SERR Enable |
| R-0 | | R/W-0 | R-0 | R/W-0 |

| 7 | 6 | 5 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | Parity Error | Reserved | | Bus Master | Memory Space | IO Space |
| R-0 | R/W-0 | R-0 | | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-83. STATUS_COMMAND Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Parity Error | 0 | Set if function receives poisoned TLP. |
| 30 | Signaled System Error | 0 | Set if function sends a ERR_FATAL or ERR_NONFATAL message and SERR enable bit is set to one. |
| 29 | Received Master Abort | 0 | Set when a Requester receives a Completion with Unsupported Request Completion Status. |
| 28 | Received Target Abort | 0 | Set when a Requester receives a Completion with Completer Abort Status. |
| 27 | Signaled Target Abort | 0 | Set when a function acting as a Completer terminates a request by issuing Completer Abort Completion Status to the Requester. |
| 26-25 | Reserved | 0 | Reserved |
| 24 | Data Parity Error | 0 | This bit is set by a Requester if the Parity Error Enable bit is set in its Command register and either the condition that the requester receives a poisoned Completion or the condition that the requester poisons a write request are true. |
| 23-21 | Reserved | 0 | Reserved |
| 20 | Capabilities List | 0 | For PCIe, this field must be set to 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-83. STATUS_COMMAND Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 19 | Interrupt Status | 0 | Indicates that the function has received an interrupt. |
| 18-11 | Reserved | 0 | Reserved |
| 10 | INTx Disable | 0 | Setting this bit disables generation of INTx messages. |
| 9 | Reserved | 0 | Reserved |
| 8 | SERR Enable | 0 | When set, it enables System Error reporting to the Root Complex. |
| 7 | Reserved | 0 | Reserved |
| 6 | Parity Error | 0 | This bit controls whether or not the device responds to detected parity errors. If this bit is set, the PCIESS will respond normally to parity errors. If this bit is cleared, the PCIES will ignore detected parity errors. |
| 5-3 | Reserved | 0 | Reserved |
| 2 | Bus Master | 0 | Enables mastership of the bus. |
| 1 | Memory Space | 0 | This bit is set to enable the device to respond to memory accesses. |
| 0 | IO Space | 0 | This bit is set to enable the device to respond to I/O accesses. This functionality is not supported in PCIESS and there this bit is set to zero. |

#### 19.5.2.3 CLASSCODE_REVID

The class code and revision ID register (CLASSCODE_REVID) is described in the figure and table below.

**Figure 19-78. CLASSCODE_REVID Register**

| 31 | 8 |
|---|---|
| Class Code | |
| R-0 | |

| 7 | 0 |
|---|---|
| Revision ID | |
| R-1 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-84. CLASSCODE_REVID Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Class Code | 0-FF FFFFh | PCIe Class Code per PCIe Base Specifications Revision 2.0. Writable from internal bus interface. |
| 7-0 | Revision ID | 0-FFh | Updated with each revision of hardware. Writable from internal bus interface. |

### *19.5.3 Configuration Type 0 Registers*

**Table 19-85. Configuration Type 0 Register Summary**

| Offset | Acronym | Section |
|---|---|---|
| Ch | BIST_HEADER | Section 19.5.3.1 |
| 10h | BAR0 | Section 19.5.3.2 |
| 14h | BAR1 | Section 19.5.3.3 |
| | BAR1 | Section 19.5.3.3.1 |
| 18h | BAR2 | Section 19.5.3.4 |
| 1Ch | BAR3 | Section 19.5.3.5 |
| | BAR3 | Section 19.5.3.5.1 |
| 20h | BAR4 | Section 19.5.3.6 |
| 24h | BAR5 | Section 19.5.3.7 |
| | BAR5 | Section 19.5.3.7.1 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-85. Configuration Type 0 Register Summary (continued)**

| Offset | Acronym | Section |
|---|---|---|
| 28h | CARDBUS | Section 19.5.3.8 |
| 2Ch | SUBSYS_VNDR_ID | Section 19.5.3.9 |
| 30h | EXPNSN_ROM | Section 19.5.3.10 |
| 34h | CAP_PTR | Section 19.5.3.11 |
| 3Ch | INT_PIN | Section 19.5.3.13 |

### 19.5.3.1 BIST_HEADER

The BIST, Header Type, Latency Time and Cache Line Size register (BIST_HEADER) is described in the figure and table below.

**Figure 19-79. BIST_HEADER Register**

| 31 | 30 | 29 28 | 27 24 | 23 | 22 16 |
|---|---|---|---|---|---|
| BIST Capable | Start BIST | Reserved | Completion Code | Multi-function Device | Header Type |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

| 15 8 | 7 0 |
|---|---|
| Latency Timer | Cache Line Size |
| R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-86. BIST_HEADER Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | BIST Capable | 0 | Returns a 1 for BIST capability and 0 otherwise. Not supported by PCIESS. |
| 30 | Start BIST | 0 | Write a 1 to start BIST. Not supported by PCIESS. |
| 29-28 | Reserved | 0 | Reserved |
| 27-24 | Completion Mode | 0-Fh | Completion Code. Not supported by PCIESS. |
| 23 | Multi-function Device | 0 | Returns 1 if it is a multi function device. Writable from internal bus interface. |
| 22-16 | Header Type | 0-7Fh | Configuration Header Format. It is set to 1 for RC and cleared to 0 for EP. |
| 15-8 | Latency Timer | 0-FFh | Not applicable in PCIe |
| 7-0 | Cache Line Size | 0-FFh | Not applicable in PCIe |

### 19.5.3.2 BAR0 Register

The base address register 0 (BAR0) is described in the figure and table below.

**Figure 19-80. BAR0 Register**

| 31 8 |
|---|
| Base Address |
| R-0 |

| 7 | 6 4 | 3 | 2 1 | 0 |
|---|---|---|---|---|
| Base Address | Reserved | Prefetchable | Type | Memory Space |
| R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-87. BAR0 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Base Address | 0-1FF FFFFh | Base Address |
| 6-4 | Reserved | 0 | Reserved |
| 3 | Prefetchable | 0 | For memory BARs, it indicates whether the region is prefetchable. For IO Bars, it is used as second LSB of the base address. Writable from internal bus interface. |
| 2-1 | Type | 0-3h | Decode type. Writable from internal bus interface. |
| | | 0h | 32 bit decode |
| | | 1h | Reserved |
| | | 2h | 64 bit decode |
| | | 3h | Reserved |
| 0 | Memory Space | 0 | Set to indicate Memory Space. Writable from internal bus interface. |

### 19.5.3.3 BAR1

The base address register 1 (BAR1) is described in the figure and table below.

**Figure 19-81. BAR1 Register**

| 31 | 8 |
|---|---|
| Base Address | |
| R-0 | |

| 7 | 6 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Base Address | Reserved | | Prefetchable | Type | | Memory Space |
| R-0 | R-0 | | R-1 | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-88. BAR1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Base Address | 0-1FF FFFFh | Base Address |
| 6-4 | Reserved | 0 | Reserved |
| 3 | Prefetchable | 0 | For memory BARs, it indicates whether the region is prefetchable. For IO Bars, it is used as second LSB of the base address. Writable from internal bus interface. |
| 2-1 | Type | 0-3h | Decode type. Writable from internal bus interface. |
| | | 0h | 32 bit decode |
| | | 1h | Reserved |
| | | 2h | 64 bit decode |
| | | 3h | Reserved |
| 0 | Memory Space | 0 | Set to indicate Memory Space. Writable from internal bus interface. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 19.5.3.3.1 BAR1 (64bit BAR0) Register

The base address register 1 (BAR1) (64bit BAR0) is described in the figure and table below.

**Figure 19-82. BAR1 Register**

| 31 | 0 |
|---|---|
| Base Address | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 19-89. BAR1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Base Address | 0-FFFF FFFFh | Upper 32 bits of BAR0 address |

### 19.5.3.4 BAR2

The base address register 2 (BAR2) is described in the figure and table below.

**Figure 19-83. BAR2 Register**

| 31 | 8 |
|---|---|
| Base Address | |
| R-0 | |

| 7 | 6 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Base Address | Reserved | | Prefetchable | Type | | Memory Space |
| R-0 | R-0 | | R-1 | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 19-90. BAR2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Base Address | 0-1FF FFFFh | Base Address |
| 6-4 | Reserved | 0 | Reserved |
| 3 | Prefetchable | 0 | For memory BARs, it indicates whether the region is prefetchable. For IO Bars, it is used as second LSB of the base address. Writable from internal bus interface. |
| 2-1 | Type | 0-3h | Decode type. Writable from internal bus interface. |
| | | 0h | 32 bit decode |
| | | 1h | Reserved |
| | | 2h | 64 bit decode |
| | | 3h | Reserved |
| 0 | Memory Space | 0 | Set to indicate Memory Space. Writable from internal bus interface. |

### 19.5.3.5 BAR3 Register

The base address register 3 (BAR3) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-84. BAR3 Register**

| 31 | 8 |
|---|---|
| Base Address | |
| R-0 | |

| 7 | 6 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Base Address | Reserved | | Prefetchable | Type | | Memory Space |
| R-0 | R-0 | | R-1 | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-91. BAR3 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Base Address | 0-1FF FFFFh | Base Address |
| 6-4 | Reserved | 0 | Reserved |
| 3 | Prefetchable | 0 | For memory BARs, it indicates whether the region is prefetchable. For IO Bars, it is used as second LSB of the base address. Writable from internal bus interface. |
| 2-1 | Type | 0-3h | Decode type. Writable from internal bus interface. |
| | | 0h | 32 bit decode |
| | | 1h | Reserved |
| | | 2h | 64 bit decode |
| | | 3h | Reserved |
| 0 | Memory Space | 0 | Set to indicate Memory Space. Writable from internal bus interface. |

#### *19.5.3.5.1 BAR3 (64 bit BAR2)Register*

The base address register 3 (BAR2) (64 bit BAR2) is described in the figure and table below.

**Figure 19-85. BAR3 Register**

| 31 | 0 |
|---|---|
| Base Address | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-92. BAR3 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Base Address | 0-FFFF FFFFh | Upper 32 bits of BAR2 address |

### 19.5.3.6 BAR4

The base address register 4 (BAR4) is described in the figure and table below.

**Figure 19-86. BAR4 Register**

| 31 | 8 |
|---|---|
| Base Address | |
| R-0 | |

| 7 | 6 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Base Address | Reserved | | Prefetchable | Type | | Memory Space |
| R-0 | R-0 | | R-1 | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-93. BAR4 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Base Address | 0-1FF FFFFh | Base Address |
| 6-4 | Reserved | 0 | Reserved |
| 3 | Prefetchable | 0 | For memory BARs, it indicates whether the region is prefetchable. For IO Bars, it is used as second LSB of the base address. Writable from internal bus interface. |
| 2-1 | Type | 0-3h | Decode type. Writable from internal bus interface. |
| | | 0h | 32 bit decode |
| | | 1h | Reserved |
| | | 2h | 64 bit decode |
| | | 3h | Reserved |
| 0 | Memory Space | 0 | Set to indicate Memory Space. Writable from internal bus interface. |

### 19.5.3.7 BAR5 Register

The base address register 5 (BAR5) is described in the figure and table below.

**Figure 19-87. BAR5 Register**

| 31 | 8 |
|---|---|
| Base Address | |
| R-0 | |

| 7 | 6 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Base Address | Reserved | | Prefetchable | Type | | Memory Space |
| R-0 | R-0 | | R-1 | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-94. BAR5 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Base Address | 0-1FF FFFFh | Base Address |
| 6-4 | Reserved | 0 | Reserved |
| 3 | Prefetchable | 0 | For memory BARs, it indicates whether the region is prefetchable. For IO Bars, it is used as second LSB of the base address. Writable from internal bus interface. |
| 2-1 | Type | 0-3h | Decode type. Writable from internal bus interface. |
| | | 0h | 32 bit decode |
| | | 1h | Reserved |
| | | 2h | 64 bit decode |
| | | 3h | Reserved |
| 0 | Memory Space | 0 | Set to indicate Memory Space. Writable from internal bus interface. |

#### *19.5.3.7.1 BAR5 (64 bit BAR4) Register*

The base address register 5 (BAR5) (64 bit BAR4) is described in the figure and table below.

**Figure 19-88. BAR5 Register**

| 31 | 0 |
|---|---|
| Base Address | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

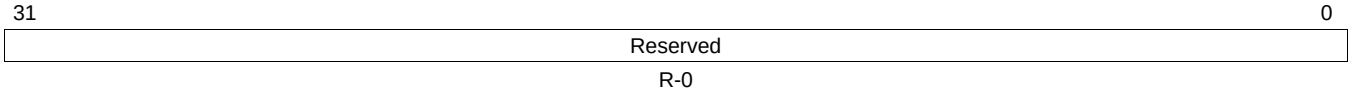
**Table 19-95. BAR5 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Base Address | 0-FFFF FFFFh | Upper 32 bits of BAR4 address |

### 19.5.3.8 CARDBUS

The CardBus CIS pointer (CARDBUS) register is described in the figure and table below.

**Figure 19-89. CARDBUS Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

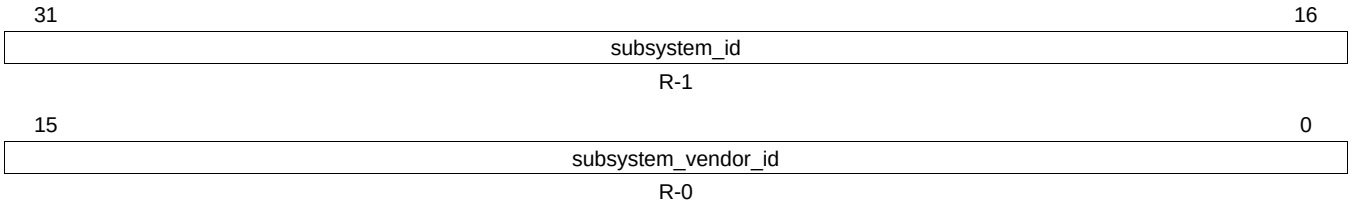
**Table 19-96. CARDBUS Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Reserved | 0 | Reserved |

### 19.5.3.9 SUBSYS_VNDR_ID

The subsystem and subsystem vendor ID register (SUBSYS_VNDR_ID) is described in the figure and table below.

**Figure 19-90. SUBSYS_VNDR_ID Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

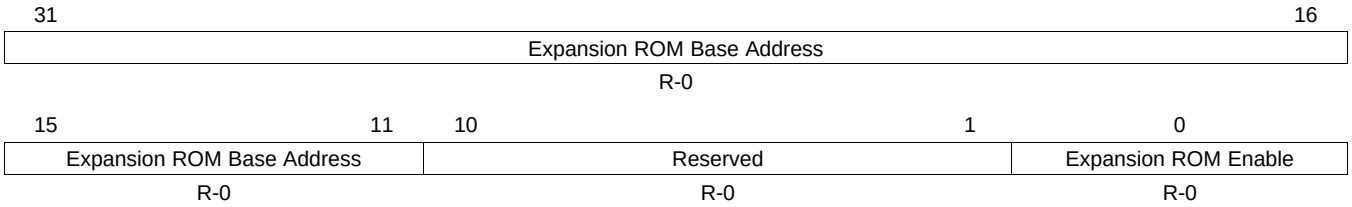
**Table 19-97. SUBSYS_VNDR_ID Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | subsystem_id | 0-FFFFh | PCIe Subsystem ID (TBD). Writable from internal bus interface. |
| 15-0 | subsystem_vendor_id | 0-FFFFh | PCIe Subsystem Vendor ID (TBD). Writable from internal bus interface. |

### 19.5.3.10 EXPNSN_ROM

The expansion ROM base address (EXPNSN_ROM) is described in the figure and table below.

**Figure 19-91. EXPNSN_ROM Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

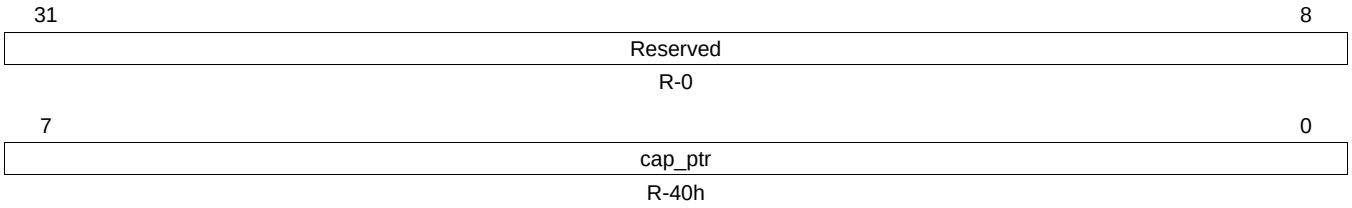
**Table 19-98. EXPNSN_ROM Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Expansion ROM Base Address | 0-1F FFFFh | Address of Expansion ROM |
| 10-1 | Reserved | 0 | Reserved |
| 0 | Expansion ROM Enable | 0 | Expansion ROM Enable |

### 19.5.3.11 CAP_PTR

The capabilities pointer register (CAP_PTR) is described in the figure and table below.

**Figure 19-92. CAP_PTR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

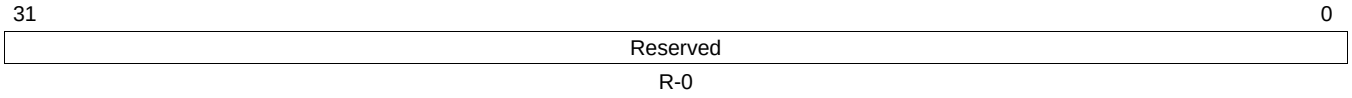
**Table 19-99. CAP_PTR Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | cap_ptr | 0-FFh | First Capability Pointer. By default, it points to Power Management Capability structure. Writable from internal bus interface. |

### 19.5.3.12 Reserved Register

This register is reserved.

**Figure 19-93. Reserved Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-100. Reserved Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Reserved | 0 | Reserved |

### 19.5.3.13 INT_PIN

The interrupt pin register (INT_PIN) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-94. INT_PIN Register**

| 31–24 | 23–16 |
|---|---|
| max_latency | min_grant |
| R-0 | R-0 |

| 15–8 | 7–0 |
|---|---|
| int_pin | int_line |
| R-1 | R/W-FFh |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-101. INT_PIN Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | max_latency | 0-FFh | Not applicable to PCI Express |
| 23-16 | min_grant | 0-FFh | Not applicable to PCI Express |
| 15-8 | int_pin | 0-FFh | Interrupt Pin. Valid values at 00h, 01h, 02h, 03h and 04h for unused legacy interrupt, INTA, INTB, INTC or INTD respectively. For single function configuration, the core only uses INTA. Writable from internal bus interface. |
| 7-0 | int_line | 0-FFh | Interrupt Line |

## 19.5.4 Configuration Type 1 Registers

**Table 19-102. Configuration Type 1 Registers**

| Offset | Acronym | Section |
|---|---|---|
| Ch | BIST_HEADER | Section 19.5.4.1 |
| 10h | BAR0 (64/32 Bit Mode) | Section 19.5.4.2 |
| 14h | BAR1 (32-bit mode) | Section 19.5.4.3 |
| | BAR1 (64-bit mode) | Section 19.5.4.3.1 |
| 18h | BUSNUM | Section 19.5.4.4 |
| 1Ch | SECSTAT | Section 19.5.4.5 |
| 20h | MEMSPACE | Section 19.5.4.6 |
| 24h | PREFETCH_MEM | Section 19.5.4.7 |
| 28h | PREFETCH_BASE | Section 19.5.4.8 |
| 2Ch | PREFETCH_LIMIT | Section 19.5.4.9 |
| 30h | IOSPACE | Section 19.5.4.10 |
| 34h | CAP_PTR | Section 19.5.4.11 |
| 38h | EXPNSN_ROM | Section 19.5.4.12 |
| 3Ch | BRIDGE_INT | Section 19.5.4.13 |

### 19.5.4.1 BIST_HEADER

The BIST, Header Type, Latency Time and Cache Line Size register (BIST_HEADER) is described in the figure and table below.

**Figure 19-95. BIST_HEADER Register**

| 31 | 30 | 29–28 | 27–24 | 23 | 22–16 |
|---|---|---|---|---|---|
| BIST Capable | Start BIST | Reserved | Completion Code | Multi-function Device | Header Type |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-1 |

| 15–8 | 7–0 |
|---|---|
| Latency Timer | Cache Line Size |
| R-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-103. BIST_HEADER Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | BIST Capable | 0 | Returns a 1 for BIST capability and 0 otherwise. Not supported by PCIESS. |
| 30 | Start BIST | 0 | Write a 1 to start BIST. Not supported by PCIESS. |
| 29-28 | Reserved | 0 | Reserved |
| 27-24 | Completion Code | 0-Fh | Completion Code. Not supported by PCIESS. |
| 23 | Multi-function Device | 0 | Returns 1 if it is a multi function device. Writable from internal bus interface. |
| 22-16 | Header Type | 0-7Fh | Configuration Header Format. It is set to 1 for RC and cleared to 0 for EP. |
| 15-8 | Latency Timer | 0-FFh | Not applicable in PCIe. |
| 7-0 | Cache Line Size | 0-FFh | Not applicable in PCIe. |

### 19.5.4.2 BAR0 (64/32 Bit Mode)

The base address register 0 (64/32 bit mode) is described in the figure and table below.

**Figure 19-96. BAR0 Register**

| 31 | 8 |
|---|---|
| Base Address | |
| R-0 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Base Address | | Prefetchable | Type | | Memory Space |
| R-0 | | R-0 | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-104. BAR0 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Base Address | 0-FFF FFFFh | Base Address. Actual writable bits are determined by BAR0 Mask Register. |
| 3 | Prefetchable | 0 | Set to indicate prefetchable space. This field is writable from internal bus interface. |
| 2-1 | Type | 0-3h | Decode type. Writable from internal bus interface. |
| | | 0h | 32 bit decode |
| | | 1h | Reserved |
| | | 2h | 64 bit decode |
| | | 3h | Reserved |
| 0 | Memory Space | 0 | Set to indicate Memory Space. Writable from internal bus interface. |

### 19.5.4.3 BAR1 (32-Bit Mode)

The base address register 1 (BAR1) (32-bit mode) is described in the figure and table below.

**Figure 19-97. BAR1 Register**

| 31 | 8 |
|---|---|
| Base Address | |
| R-0 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Base Address | | Prefetchable | Type | | Memory Space |
| R-0 | | R-1 | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-105. BAR1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Base Address | 0-FFF FFFFh | Base Address. Actual writable bits are determined by BAR0 Mask Register. |
| 3 | Prefetchable | 0 | Set to indicate prefetchable space. This field is writable from internal bus interface. |
| 2-1 | Type | 0-3h | Decode type. Writable from internal bus interface. |
| | | 0h | 32 bit decode |
| | | 1h | Reserved |
| | | 2h | 64 bit decode |
| | | 3h | Reserved |
| 0 | Memory Space | 0 | Set to indicate Memory Space. Writable from internal bus interface. |

#### *19.5.4.3.1 BAR1(64 Bit Mode)*

The base address register 1 (BAR1) (64 bit mode) is described in the figure and table below.

**Figure 19-98. BAR1 Register**

| 31 | 0 |
|---|---|
| Base Address | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-106. BAR1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Base Address | 0-FFFF FFFFh | Base Address high 32 bits for BAR0. Actual writable bits are determined by BAR0 Mask Register. |

### 19.5.4.4 BUSNUM

The latency timer and bus number register (BUSNUM) is described in the figure and table below.

**Figure 19-99. BUSNUM Register**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| Secondary Latency Timer | | Subordinate Bus Number | |
| R-0 | | R/W-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Secondary Bus Number | | Primary Bus Number | |
| R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-107. BUSNUM Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Secondary Latency Timer | 0-FFh | Not applicable in PCI Express. |
| 23-16 | Subordinate Bus Number | 0-FFh | Subordinate Bus Number. This is highest bus number on downstream interface. |
| 15-8 | Secondary Bus Number | 0-FFh | Secondary Bus Number. It is typically 1h for RC. |
| 7-0 | Primary Bus Number | 0-FFh | Primary Bus Number. It is zero for RC and nonzero for switch devices only. |

### 19.5.4.5 SECSTAT

The secondary status and IO base/limit register (SECSTAT) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-100. SECSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DTCT_PERROR | RX_SYS_ERROR | RX_MST_ABORT | RX_TGT_ABORT | TX_TGT_ABORT | Reserved | | MST_DPERR |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R-0 | | R/W1C-0 |

| 23 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 12 | 11 | 8 |
|---|---|---|---|
| IO Limit | | Reserved | IO Addressing |
| R/W-0 | | R-0 | R-0 |

| 7 | 4 | 3 | 0 |
|---|---|---|---|
| IO Base | | Reserved | IO Addressing |
| R/W-0 | | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-108. SECSTAT Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | DTCT_PERROR | 0 | Detected Parity Error |
| 30 | RX_SYS_ERROR | 0 | Received System Error |
| 29 | RX_MST_ABORT | 0 | Received Master Abort |
| 28 | RX_TGT_ABORT | 0 | Received Target Abort |
| 27 | TX_TGT_ABORT | 0 | Signaled Target Abort |
| 26-25 | Reserved | 0 | Reserved |
| 24 | MST_DPERR | 0 | Master Data Parity Error |
| 23-16 | Reserved | 0 | Reserved |
| 15-12 | IO Limit | 0-Fh | I/O Space Limit |
| 11-9 | Reserved | 0 | Reserved |
| 8 | IO Addressing | 0 | 32 bit IO Space indication for IO Limit Register. Indicates 16 bit and 32 bit addressing for 0 and 1 respectively. |
| 7-4 | IO Base | 0-Fh | IO Space Base |
| 3-1 | Reserved | 0 | Reserved |
| 0 | IO Addressing | 0 | Indicates 0 for 16 bit and 1 for 32 bit addressing for the IO Base Register. Writable from internal bus interface. |

### 19.5.4.6 MEMSPACE

The memory limit and base register (MEMSPACE) is described in the figure and table below.

**Figure 19-101. MEMSPACE Register**

| 31 | 20 | 19 | 16 |
|---|---|---|---|
| Memory Limit | | Reserved | |
| R/W-0 | | R-0 | |

| 15 | 4 | 3 | 0 |
|---|---|---|---|
| Memory Base | | Reserved | |
| R/W-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-109. MEMSPACE Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | Memory Limit | 0-FFFh | Memory Limit Address |
| 19-16 | Reserved | 0 | Reserved |
| 15-4 | Memory Base | 0-FFFh | Memory Base Address |
| 3-0 | Reserved | 0 | Reserved |

### 19.5.4.7 PREFETCH_MEM

The prefetchable memory limit and base register (PREFETCH_MEM) is described in the figure and table below.

**Figure 19-102. PREFETCH_MEM Register**

| 31 | 20 | 19 | 17 | 16 |
|---|---|---|---|---|
| End Address | | Reserved | | Memory Addressing |
| R/W-0 | | R-0 | | R-0 |

| 15 | 4 | 3 | 1 | 0 |
|---|---|---|---|---|
| Start Address | | Reserved | | Memory Addressing |
| R/W-0 | | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-110. PREFETCH_MEM Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | End Address | 0-FFFh | Upper 12 bits of 32 bit Prefetchable Memory End Address |
| 19-17 | Reserved | 0 | Reserved |
| 16 | Memory Addressing | | Memory addressing. Writable from internal bus interface. |
| | | 0 | 32 bit memory addressing |
| | | 1 | 64 bit memory addressing |
| 15-4 | Start Address | 0-FFFh | Upper 12 bits of 32 bit Prefetchable Memory Start Address |
| 3-1 | Reserved | 0 | Reserved |
| 0 | Memory Addressing | | Memory addressing. Writable from internal bus interface. |
| | | 0 | 32 bit memory addressing |
| | | 1 | 64 bit memory addressing |

### 19.5.4.8 PREFETCH_BASE

The prefetchable memory base upper 32 bits register (PREFETCH_BASE) is described in the figure and table below.

**Figure 19-103. PREFETCH_BASE Register**

| 31 | 0 |
|---|---|
| Base Address | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-111. PREFETCH_BASE Field Descriptions**

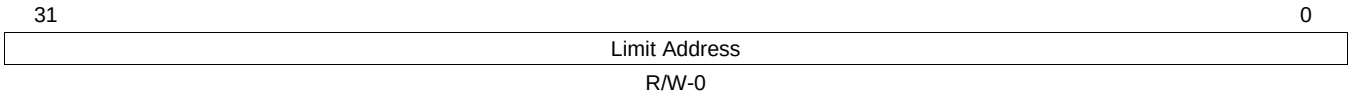
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Base Address | 0-FFFF FFFFh | Upper 32 bits of base address of prefetchable memory space. Used when 64 bit addressing is enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.4.9 PREFETCH_LIMIT

The prefetchable limit upper 32 bits register (PREFETCH_LIMIT) is described in the figure and table below.

**Figure 19-104. PREFETCH_LIMIT Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

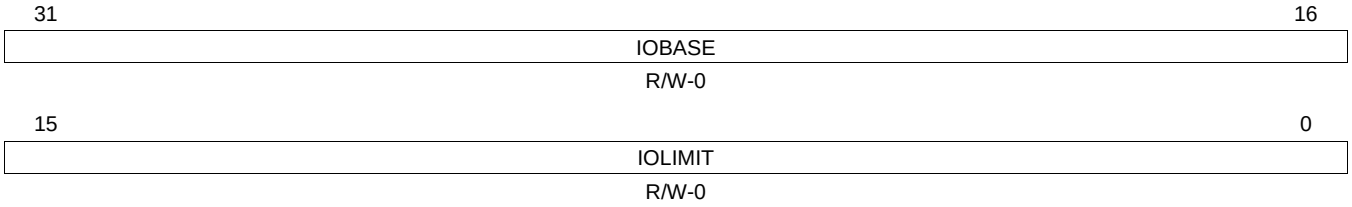
**Table 19-112. PREFETCH_LIMIT Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | Limit Address | 0-FFFF FFFFh | Upper 32 bits of Limit Address of Prefetchable Memory Space. Used with 64 bit prefetchable memory addressing only. |

### 19.5.4.10 IOSPACE

The IO base and limit upper 16 bits register (IOSPACE) is described in the figure and table below.

**Figure 19-105. IOSPACE Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

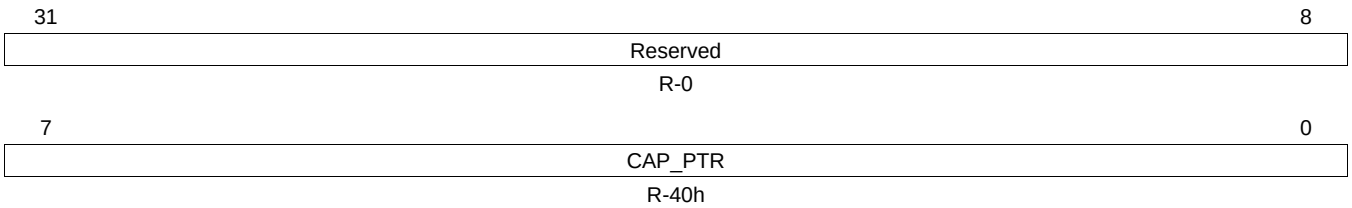
**Table 19-113. IOSPACE Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | IOBASE | 0-FFFFh | Upper 16 bits of IO Base |
| 15-0 | IOLIMIT | 0-FFFFh | Upper 16 bits of IO Limit |

### 19.5.4.11 CAP_PTR

The capabilities pointer register (CAP_PTR) is described in the figure and table below.

**Figure 19-106. CAP_PTR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-114. CAP_PTR Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | CAP_PTR | 0-FFh | First Capability Pointer. By default, it points to Power Management Capability structure. Writable from internal bus interface. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.4.12 EXPNSN_ROM

The expansion ROM base address register (EXPNSN_ROM) is described in the figure and table below.

**Figure 19-107. EXPNSN_ROM Register**

| 31 | 16 |
|---|---|
| Expansion ROM Base Address | |
| R-0 | |

| 15 | 11 | 10 | 1 | 0 |
|---|---|---|---|---|
| Expansion ROM Base Address | | Reserved | | Expansion ROM Enable |
| R-0 | | R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-115. EXPNSN_ROM Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-11 | Expansion ROM Base Address | 0-1F FFFFh | Address of Expansion ROM |
| 10-1 | Reserved | 0 | Reserved |
| 0 | Expansion ROM Enable | 0 | Expansion ROM Enable |

### 19.5.4.13 BRIDGE_INT

The bridge control register (BRIDGE_INT) is described in the figure and table below.

**Figure 19-108. BRIDGE_INT Register**

| 31 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|
| Reserved | | SERREN_STATUS | TIMER_STATUS | SEC_TIMER | PRI_TIMER |
| R-0 | | R-0 | R-0 | R-0 | R-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| B2B_EN | SEC_BUS_RST | MST_ABORT_MODE | VGA_DECODE | VGA_EN | ISA_EN | SERR_EN | PERR_RESP_EN |
| R-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 8 |
|---|---|
| INT_PIN | |
| R-1 | |

| 7 | 0 |
|---|---|
| INT_LINE | |
| R/W-FFh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-116. BRIDGE_INT Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | Reserved | 0 | Reserved |
| 27 | SERREN_STATUS | 0 | Discard Timer SERR Enable Status. Not Applicable to PCI Express. |
| 26 | TIMER_STATUS | 0 | Discard Timer Status. Not applicable to PCI Express. |
| 25 | SEC_TIMER | 0 | Secondary Discard Timer. Not applicable to PCI Express |
| 24 | PRI_TIMER | 0 | Primary Discard Timer. Not applicable to PCI Express. |
| 23 | B2B_EN | 0 | Fast Back to Back Transactions Enable. Not applicable to PCI Express. |
| 22 | SEC_BUS_RST | 0 | Secondary Bus Reset |
| 21 | MST_ABORT_MODE | 0 | Master Abort Mode. Not applicable to PCI Express. Always zero. |
| 20 | VGA_DECODE | 0 | VGA 16 bit Decode |
| 19 | VGA_EN | 0 | VGA Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-116. BRIDGE_INT Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 18 | ISA_EN | 0 | ISA Enable |
| 17 | SERR_EN | 0 | SERR Enable |
| 16 | PERR_RESP_EN | 0 | Parity Error Response Enable |
| 15-8 | INT_PIN | 0-FFh | Interrupt Pin. It identifies the legacy interrupt message that the device uses. Valid values are 00h for legacy interrupt not used and 01h, 02h, 03h and 04h for INTA, INTB, INTC or INTD respectively. For single function configuration, the core only uses INTA. This register is writable through internal bus interface. |
| 7-0 | INT_LINE | 0-FFh | Interrupt Line. Value is system software specified. |

### *19.5.5 Power Management Capabilities Registers*

**Table 19-117. Power Management Capability Registers**

| Offset | Acronym | Section |
|---|---|---|
| 0h | PMCAP | Section 19.5.5.1 |
| 4h | PM_CTL_STAT | Section 19.5.5.2 |

#### 19.5.5.1 PMCAP Register

The power management capability register (PMCAP) is described in the figure and table below.

**Figure 19-109. PMCAP Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PME_SUPP_N | | | | | D2_SUPP_N | D1_SUPP_N | AUX_CURR_N |
| R-0 | | | | | R-0 | R-0 | R-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| AUX_CURR_N | | DSI_N | Reserved | PME_CLK | PME_SPEC_VER | | |
| R-0 | | R-0 | R-0 | R-0 | R-3h | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| PM_NEXT_PTR | | | | | | | |
| R-50h | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PM_CAP_ID | | | | | | | |
| R-1 | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-118. PMCAP Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-27 | PME_SUPP_N | 0-1Fh | PME Support. Writable from internal bus interface. |
| 26 | D2_SUPP_N | 0 | D2 Support. Writable from internal bus interface. |
| 25 | D1_SUPP_N | 0 | D1 Support. Writable from internal bus interface. |
| 24-22 | AUX_CURR_N | 0-7h | Auxiliary Current. Writable from internal bus interface. |
| 21 | DSI_N | 0 | Device Specific Initialization. Writable from internal bus interface. |
| 20 | Reserved | 0 | Reserved |
| 19 | PME_CLK | 0 | PME Clock. Hardwired to Zero. |
| 18-16 | PME_SPEC_VER | 0-7h | Power Management Specification Version. Writable from internal bus interface. |
| 15-8 | PM_NEXT_PTR | 0-FFh | Next Capability Pointer. Writable from internal bus interface. |
| 7-0 | PM_CAP_ID | 0-FFh | Power Management Capability ID. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.5.2 PM_CTL_STAT Register

The power management control and status register (PM_CTL_STAT) is described in the figure and table below.

**Figure 19-110. PM_CTL_STAT Register**

| 31 | 24 |
|---|---|
| DATA_REG | |
| R-0 | |

| 23 | 22 | 21 | 16 |
|---|---|---|---|
| CLK_CTRL_EN | B2_B3_SUPPORT | Reserved | |
| R-0 | R-0 | R-0 | |

| 15 | 14 | 13 | 12 | 9 | 8 |
|---|---|---|---|---|---|
| PME_STATUS | DATA_SCALE | | DATA_SELECT | | PME_EN |
| R/W1C-0 | R-0 | | R-0 | | R/W-0 |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | NO_SOFT_RST | Reserved | PWR_STATE | |
| R-0 | | R-0 | R-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 19-119. PM_CTL-STAT Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | DATA_REG | 0-FFh | Data register for additional information. Not supported. |
| 23 | CLK_CTRL_EN | 0 | Bus Power/Clock Control Enable. Hardwired to zero. |
| 22 | B2_B3_SUPPORT | 0 | B2 and B3 support. Hardwired to zero. |
| 21-16 | Reserved | 0 | Reserved |
| 15 | PME_STATUS | 0 | PME Status. Indicates if a previously enabled PME event occurred or not. |
| 14-13 | DATA_SCALE | 0-3h | Data Scale. Not supported. |
| 12-9 | DATA_SELECT | 0-Fh | Data select. Not supported. |
| 8 | PME_EN | 0 | PME Enable. Value of 1 indicates device is enabled to generate PME. Writable from internal bus interface. |
| 7-4 | Reserved | 0 | Reserved |
| 3 | NO_SOFT_RST | 0 | No soft reset. It is set to disable reset during a transition from D3 to D0. Writable from internal bus interface. |
| 2 | Reserved | 0 | Reserved |
| 1-0 | PWR_STATE | 0-3h | Power State. Controls the device power state. Writes are ignored if the state is not supported. Writable from internal bus interface. |
| | | 0 | D0 power state |
| | | 1h | D1 power state |
| | | 2h | D2 power state |
| | | 3h | D3 power state |

## *19.5.6 Message Signaled Interrupts Registers*

**Table 19-120. Message Signaled Interrupts Registers**

| Offset | Acronym | Section |
|---|---|---|
| 0h | MSI_CAP | Section 19.5.6.1 |
| 4h | MSI_LOW32 | Section 19.5.6.2 |
| 8h | MSI_UP32 | Section 19.5.6.3 |
| Ch | MSI_DATA | Section 19.5.6.4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.6.1 MSI_CAP Register

The MSI capabilities register is described in the figure and table below.

**Figure 19-111. MSI_CAP Register**

| 31 | 24 | 23 | 22 | 20 | 19 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | 64BIT_EN | MULT_MSG_EN | | MULT_MSG_CAP | | MSI_EN |
| R-0 | | R-1 | R/W-0 | | R-0 | | R/W-0 |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| NEXT_CAP | | CAP_ID | |
| R-70h | | R-5h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-121. MSI_CAP Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Reserved | 0 | Reserved |
| 23 | 64BIT_EN | 0 | 64-Bit address enabled. Writable from internal bus interface. |
| 22-20 | MULT_MSG_EN | 0-7h | Multiple Message Enabled. Indicates that multiple message mode is enabled by software. Number of messages enabled must not be greater than Multiple Message Capable value. |
| 19-17 | MULT_MSG_CAP | 0-7h | Multiple Message Capable. Writable from internal bus interface. |
| 16 | MSI_EN | 0 | MSI Enabled. When set, INTx must be disabled. |
| 15-8 | NEXT_CAP | 0-FFh | Next Capability Pointer containing the offset to next capability structure. Writable from internal bus interface. |
| 7-0 | CAP_ID | 0-FFh | MSI Capability ID |

### 19.5.6.2 MSI_LOW32 Register

The MSI lower 32 bits register (MSI_LOW32) is described in the figure and table below.

**Figure 19-112. MSI_LOW32 Register**

| 31 | 0 |
|---|---|
| LOW32_ADDR | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-122. MSI_LOW32 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | LOW32_ADDR | 0-FFFF FFFFh | Lower 32 bit address |

### 19.5.6.3 MSI_UP32 Register

The MSI upper 32 bits register (MSI_UP32) is described in the figure and table below.

**Figure 19-113. MSI_UP32 Register**

| 31 | 16 |
|---|---|
| UP32_ADDR | |
| R/W-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| UP32_ADDR | | Reserved | |
| R/W-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

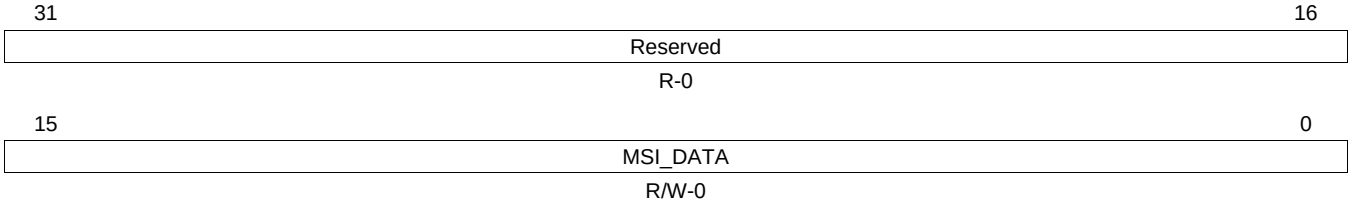
**Table 19-123. MSI_UP32 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | UP32_ADDR | 0-3FFF FFFFh | Upper 32 bit address |
| 1-0 | Reserved | 0 | Reserved |

#### 19.5.6.4 MSI_DATA Register

The MSI data register (offset is 0x08 is 64 bit not enabled) (MSI_DATA) is described in the figure and table below.

**Figure 19-114. MSI_DATA Register**




LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 19-124. MSI_DATA Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15-0 | MSI_DATA | 0-FFFFh | MSI Data |

### *19.5.7 PCI Express Capabilities Registers*

**Table 19-125. PCIExpress Capabilities Registers**

| Offset | Acronym | Section |
|---|---|---|
| 0h | PCIES_CAP | Section 19.5.7.1 |
| 4h | DEVICE_CAP | Section 19.5.7.2 |
| 8h | DEV_STAT_CTRL | Section 19.5.7.3 |
| Ch | LINK_CAP | Section 19.5.7.4 |
| 10h | LINK_STAT_CTRL | Section 19.5.7.5 |
| 14h | SLOT_CAP | Section 19.5.7.6 |
| 18h | SLOT_STAT_CTRL | Section 19.5.7.7 |
| 1Ch | ROOT_CTRL_CAP | Section 19.5.7.8 |
| 20h | ROOT_STATUS | Section 19.5.7.9 |
| 24h | DEV_CAP2 | Section 19.5.7.10 |
| 28h | DEV_STAT_CTRL2 | Section 19.5.7.11 |
| 30h | LINK_CTRL2 | Section 19.5.7.12 |

#### 19.5.7.1 PCIES_CAP

The PCI Express capabilities register (PCIES_CAP) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-115. PCIES_CAP Register**

| 31 | 30 | 29 | 25 | 24 | 23 | 20 | 19 | 16 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | INT_MSG | | SLT_IMPL_N | DPORT_TYPE | | PCIE_CAP | |
| R-0 | | R-0 | | R-0 | R-x | | R-2h | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| NEXT_CAP | | CAP_ID | |
| R-0 | | R-10h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-126. PCIES_CAP Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Reserved |
| 29-25 | INT_MSG | 0-1Fh | Interrupt Message Number. Updated by hardware and writable through internal bus Interface. |
| 24 | SLT_IMPL_N | 0 | Slot Implemented. Writable from internal bus interface. |
| 23-20 | DPORT_TYPE | 0-Fh | Device Port Type (4h for RC, 0 for EP) |
| 19-16 | PCIE_CAP | 0-Fh | PCI Express Capability Version |
| 15-8 | NEXT_CAP | 0-FFh | PCIe Next Capability Pointer containing the offset to next capability structure. Writable from internal bus interface. |
| 7-0 | CAP_ID | 0-FFh | PCIe Capability ID |

### 19.5.7.2 DEVICE_CAP

The device capabilities register (DEVICE_CAP) is described in the figure and table below.

**Figure 19-116. DEVICE_CAP Register**

| 31 | 28 | 27 | 26 | 25 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | PWR_LIMIT_SCALE | | PWR_LIMIT_VALUE | | Reserved | |
| R-0 | | R-0 | | R-0 | | R-0 | |

| 15 | 14 | 12 | 11 | 9 | 8 | 6 | 5 | 4 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ERR_RPT | Reserved | | L1_LATENCY | | L0_LATENCY | | EXT_TAG_FLD | PHANTOM_FLD | | MAX_PAYLD_SZ | |
| R-1 | R-0 | | R-3h | | R-4h | | R-0 | R-0 | | R-1 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-127. DEVICE_CAP Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | Reserved | 0 | Reserved |
| 27-26 | PWR_LIMIT_SCALE | 0-3h | Captured Slot Power Limit Scale |
| 25-18 | PWR_LIMIT_VALUE | 0-FFh | Captured Slow Power Limit Value |
| 17-16 | Reserved | 0 | Reserved |
| 15 | ERR_RPT | 0 | Role based Error Reporting. Writable from internal bus interface. |
| 14-12 | Reserved | 0 | Reserved |
| 11-9 | L1_LATENCY | 0-7h | Endpoint L1 Acceptable Latency. Must be 0h in RC mode. It is 3h for Endpoint. |
| 8-6 | L0_LATENCY | 0-7h | Endpoint L1 Acceptable Latency. Must be 0h in RC mode. |
| 5 | EXT_TAG_FLD | 0 | Extended Tag Field Supported. Writable from internal interface but should not be as the hardware is not capable. |
| 4-3 | PHANTOM_FLD | 0-3h | Phantom Field Supported. Writable from internal bus interface. |
| 2-0 | MAX_PAYLD_SZ | 0-7h | Maximum Payload size supported. Writable from internal bus interface. |

### 19.5.7.3 DEV_STAT_CTRL

The device status and control register (DEV_STAT_CTRL) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-117. DEV_STAT_CTRL Register**

| 31 | 24 |
|---|---|
| Reserved | |
| R-0 | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | TPEND | AUX_PWR | UNSUP_RQ_DET | FATAL_ERR | NFATAL_ERR | CORR_ERR |
| R-0 | | R-0 | R-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

| 15 | 14 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| Reserved | MAX_REQ_SZ | | NO_SNOOP | AUX_PWR_PM_EN | PHANTOM_EN | XTAG_FIELD_EN |
| R-0 | R/W-2h | | R/W-1 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| MAX_PAYLOAD | | RELAXED | UNSUP_REQ_REP | FATAL_ERR_REP | NFATAL_ERR_REP | CORR_ERR_REP |
| R/W-0 | | R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-128. DEV_STAT_CTRL Register Field Descriptions**

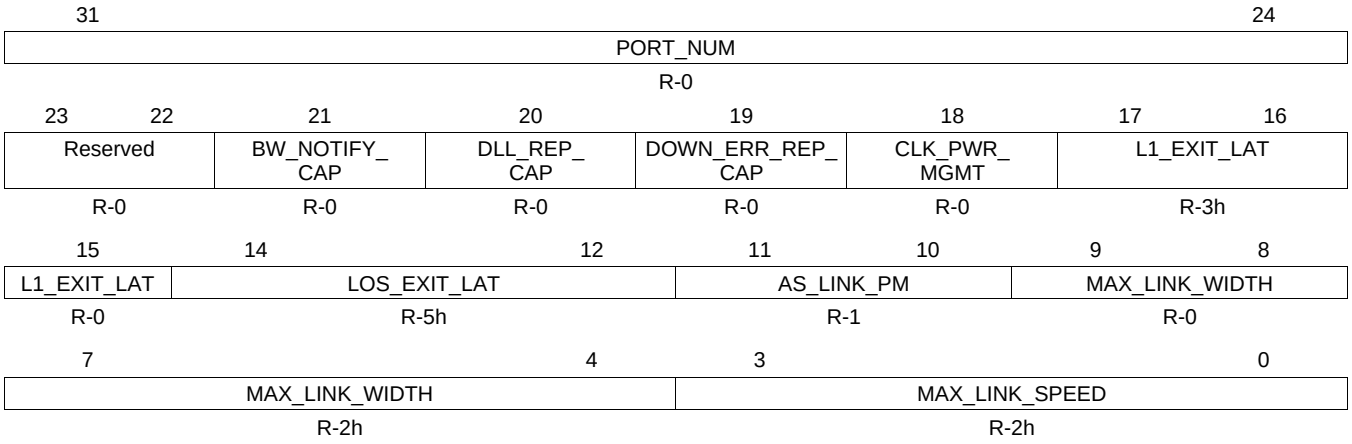
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-22 | Reserved | 0 | Reserved |
| 21 | TPEND | 0 | Transaction Pending |
| 20 | AUX_PWR | 0 | Auxiliary Power Detected |
| 19 | UNSUP_RQ_DET | 0 | Unsupported Request Detected |
| 18 | FATAL_ERR | 0 | Fatal Error Detected |
| 17 | NFATAL_ERR | 0 | Non-fatal Error Detected |
| 16 | CORR_ERR | 0 | Correctable Error Detected |
| 15 | Reserved | 0 | Reserved |
| 14-12 | MAX_REQ_SZ | 0-7h | Maximum Read Request Size |
| 11 | NO_SNOOP | 0 | Enable no snoop |
| 10 | AUX_PWR_PM_EN | 0 | AUX Power PM Enable |
| 9 | PHANTOM_EN | 0 | Phantom Function Enable |
| 8 | XTAG_FIELD_EN | 0 | Extended Tag Field Enable |
| 7-5 | MAX_PAYLOAD | 0-7h | Maximum Payload Size |
| 4 | RELAXED | 0 | Enable Relaxed Ordering |
| 3 | UNSUP_REQ_REP | 0 | Enable Unsupported Request Reporting |
| 2 | FATAL_ERR_REP | 0 | Fatal Error Reporting Enable |
| 1 | NFATAL_ERR_REP | 0 | Non-Fatal Error Reporting Enable |
| 0 | CORR_ERR_REP | 0 | Correctable Error Reporting Enable |

### 19.5.7.4 LINK_CAP

The register LINK_CAP) is (described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-118. LINK_CAP Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-129. LINK_CAP Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | PORT_NUM | 0-FFh | Port Number. Writable from internal bus interface. |
| 23-22 | Reserved | 0 | Reserved |
| 21 | BW_NOTIFY_CAP | 0 | Link Bandwidth Notification Capable. Always 1 for downstream and 0 for upstream. |
| 20 | DLL_REP_CAP | 0 | Data Link Layer Active Reporting Capable. Always 1 for downstream and 0 for upstream. |
| 19 | DOWN_ERR_REP_CAP | 0 | Surprise Down Error Reporting Capable. Not supported. Always zero. |
| 18 | CLK_PWR_MGMT | 0 | Clock Power Management. Zero for downstream ports. Writable from internal bus interface. |
| 17-15 | L1_EXIT_LAT | 0-7h | L1 Exit Latency when common clock is used. Writable from internal bus interface. |
| 14-12 | LOS_EXIT_LAT | 0-7h | L0s Exit Latency. Writable from internal bus interface. |
| 11-10 | AS_LINK_PM | 0-3h | Active State Link PM Support. Writable from internal bus interface. By default, L0s is enabled and L1 is disabled. |
| 9-4 | MAX_LINK_WIDTH | 0-3Fh | Maximum Link Width. Writable from internal bus interface. |
| 3-0 | MAX_LINK_SPEED | 0-Fh | Maximum Link Speed. Writable from internal bus interface. |

### 19.5.7.5 LINK_STAT_CTRL

The link status and control register (LINK_STAT_CTRL) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-119. LINK_STAT_CTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| LINK_BW_STATUS | LINK_BW_MGMT_STATUS | DLL_ACTIVE | SLOT_CLK_CFG | LINK_TRAINING | UNDEF | NEGOTIATED_LINK_WD | |
| R/W1C-0 | R/W1C-0 | R-0 | R-1 | R-0 | R-0 | R-0 | |

| 23 | 20 | 19 | 16 |
|---|---|---|---|
| NEGOTIATED_LINK_WD | | LINK_SPEED | |
| R-1 | | R-1 | |

| 15 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|
| Reserved | | LINK_BW_INIT_EN | LINK_BW_MGMT_INT_EN | HW_AUTO_WIDTH_DIS | CLK_PWR_MGMT_EN |
| R-0 | | R-0 | R-0 | R-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EXT_SYNC | COMMON_CLK_CFG | RETRAIN_LINK | LINK_DISABLE | RCB | Reserved | ACTIVE_LINK_PM | |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-1 | R-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-130. LINK_STAT_CTRL Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | LINK_BW_STATUS | 0 | Link Autonomous Bandwidth Status. NA and Reserved for End Point. |
| 30 | LINK_BW_MGMT_STATUS | 0 | Link Bandwidth Management Status. NA and Reserved for End Point. |
| 29 | DLL_ACTIVE | 0 | Data Link Layer Active |
| 28 | SLOT_CLK_CFG | 0 | Slot Clock Configuration. Writable from internal bus interface. |
| 27 | LINK_TRAINING | 0 | Link Training. Not applicable to Root Complex. |
| 26 | UNDEF | 0 | Undefined for PCI Express |
| 25-20 | NEGOTIATED_LINK_WD | 0-3Fh | Negotiated Link Width. Set automatically by hardware after link initialization. |
| 19-16 | LINK_SPEED | 0-Fh | Link Speed. Set automatically by hardware after link initialization. |
| 15-12 | Reserved | 0 | Reserved |
| 11 | LINK_BW_INT_EN | 0 | Link Autonomous Bandwidth Interrupt Enable. Not applicable and is Reserved for End Point. |
| 10 | LINK_BW_MGMT_INT_EN | 0 | Link Bandwidth Management Interrupt Enable. Not applicable and is Reserved for End Point. |
| 9 | HW_AUTO_WIDTH_DIS | 0 | Hardware Autonomous Width Disable. Not supported and hardwired to zero. |
| 8 | CLK_PWR_MGMT_EN | 0 | Enable Clock Power Management |
| 7 | EXT_SYNC | 0 | Extended Synch |
| 6 | COMMON_CLK_CFG | 0 | Common Clock Configuration |
| 5 | RETRAIN_LINK | 0 | Retrain Link. Not applicable and Reserved for EP. |
| 4 | LINK_DISABLE | 0 | Link disable. |
| 3 | RCB | 0 | Read Completion Boundary. Writable via internal bus interface for Root Complex. |
| 2 | Reserve | 0 | Reserved |
| 1-0 | ACTIVE_LINK_PM | 0-3h | Active State Link PM Control |

### 19.5.7.6 SLOT_CAP

The slot capabilities register (RC Mode only) (SLOT_CAP) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-120. SLOT_CAP Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SLOT_NUM | | | | | | | |
| R-0 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| SLOT_NUM | | | | | CMD_COMP_SUPP | EML_PRESENT | PWR_LMT_SCALE |
| R-0 | | | | | R-0 | R-0 | R-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| PWR_LMT_SCALE | PWR_LMT_VALUE | | | | | | |
| R-0 | R-0 | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| PWR_LMT_VALUE | HP_CAP | HP_SURPRISE | PWR_IND | ATTN_IND | MRL_SENSOR | PWR_CTL | ATTN_BUTTON |
| R-0 | R-1 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 19-131. SLOT_CAP Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-19 | SLOT_NUM | 0-1FFF | Physical Slot Number. Writable from internal bus interface. |
| 18 | CMD_COMP_SUPP | 0 | No Command Complete Support. Writable from internal bus interface. |
| 17 | EML_PRESENT | 0 | Electromechanical Interlock Present. Writable from internal bus interface. |
| 16-15 | PWR_LMT_SCALE | 0-3h | Slow Power Limit Scale. Writable from internal bus interface. |
| 14-7 | PWR_LMT_VALUE | 0-FFh | Slow Power Limit Value. Writable from internal bus interface. |
| 6 | HP_CAP | 0 | Hot Plug Capable. Writable from internal bus interface. |
| 5 | HP_SURPRISE | 0 | Hot Plug Surprise. Writable from internal bus interface. |
| 4 | PWR_IND | 0 | Power Indicator Present. Writable from internal bus interface. |
| 3 | ATTN_IND | 0 | Attention Indicator Present. Writable from internal bus interface. |
| 2 | MRL_SENSOR | 0 | MRL Sensor Present. Writable from internal bus interface. |
| 1 | PWR_CTL | 0 | Power Controller Present. Writable from internal bus interface. If there is no power controller, software must ensure that system power is up before reading Presence Detect state. |
| 0 | ATTN_BUTTON | 0 | Attention Indicator Present. Writable from internal bus interface. |

### 19.5.7.7 SLOT_STAT_CTRL

The slot status and control register (RC mode only register (SLOT_STAT_CTRL) is described in the figure and table below.

**Figure 19-121. SLOT_STAT_CTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | DLL_STATE |
| R-0 | | | | | | | R/W1C-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| EM_LOCK | PRESENCE_DET | MRL_STATE | CMD_COMLETE | PRESENCE_CHG | MRL_CHANGE | PWR_FAULT | ATTN_PRESSED |
| R-0 | R-1 | R-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | DLL_CHG_EN | EM_LOCK_CTL | PM_CTL | PM_IND_CTL | |
| R-0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-3h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ATTN_IND_CTL | | HP_INT_EN | CMD_CMP_INT_EN | PRS_DET_CHG_EN | MRL_CHG_EN | PWR_FLT_DET_EN | ATTN_BUTT_EN |
| R/W-3h | | R/W-3 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-132. SLOT_STAT_CTRL Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-25 | Reserved | 0 | Reserved |
| 24 | DLL_STATE | 0 | Data Link Layer State Changed |
| 23 | EM_LOCK | 0 | Electromechanical Lock Status |
| 22 | PRESENCE_DET | 0 | Presence Detect State |
| 21 | MRL_STATE | 0 | MRL Sensor State |
| 20 | CMD_COMLETE | 0 | Command Completed |
| 19 | PRESENCE_CHG | 0 | Presence Detect Changed |
| 18 | MRL_CHANGE | 0 | MRL Sensor Changed |
| 17 | PWR_FAULT | 0 | Power Fault Detected |
| 16 | ATTN_PRESSED | 0 | Attention Button Pressed |
| 15-13 | Reserved | 0 | Reserved |
| 12 | DLL_CHG_EN | 0 | Data Link Layer State Changed Enable |
| 11 | EM_LOCK_CTL | 0 | Electromechanical Interlock Control |
| 10 | PM_CTL | 0 | Power Controller Control |
| 9-8 | PM_IND_CTL | 0-3h | Power Indicator Control |
| 7-6 | ATTN_IND_CTL | 0-3h | Attention Indicator Control |
| 5 | HP_INT_EN | 0 | Hot Plug Interrupt Enable |
| 4 | CMD_CMP_INT_EN | 0 | Command Completed Interrupt Enable |
| 3 | PRS_DET_CHG_EN | 0 | Presence Detect Changed Enable |
| 2 | MRL_CHG_EN | 0 | MRL Sensor Changed Enable |
| 1 | PWR_FLT_DET_EN | 0 | Power Fault Detected Enable |
| 0 | ATTN_BUTT_EN | 0 | Attention Button Pressed Enable |

### 19.5.7.8 ROOT_CTRL_CAP

The root control and capabilities Rregister (RC Mode only) (ROOT_CTRL_CAP) is described in the figure and table below.

**Figure 19-122. ROOT_CTRL_CAP Register**

| 31 | 17 | 16 |
|---|---|---|
| Reserved | | CRS_SW |
| R-0 | | R-0 |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | CRS_SW_EN | PME_INT_EN | SERR_FATAL_ERR | SERR_NFATAL_ERR | SERR_EN |
| R-0 | | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-133. ROOT_CTRL_CAP Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved |
| 16 | CRS_SW | 0 | CRS Software Visibility. Not Supported. Hardwired to Zero. |
| 15-5 | Reserved | 0 | Reserved |
| 4 | CRS_SW_EN | 0 | CRS Software Visibility Enable. Not Supported and set to 0h. |
| 3 | PME_INT_EN | 0 | PME Interrupt Enable |
| 2 | SERR_FATAL_ERR | 0 | System Error on Fatal Error Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-133. ROOT_CTRL_CAP Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 1 | SERR_NFATAL_ERR | 0 | System Error on Non-fatal Error Enable |
| 0 | SERR_EN | 0 | System Error on Correctable Error Enable |

### 19.5.7.9 ROOT_STATUS

The root status and control register (RC mode only) (ROOT_STATUS) is described in the figure and table below.

**Figure 19-123. ROOT_STATUS Register**

| 31 | 18 | 17 | 16 |
|---|---|---|---|
| Reserved | | PME_PEND | PME_STATUS |
| R-0 | | R-0 | R/W1C-0 |

| 15 | 0 |
|---|---|
| PME_REQ_ID | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-134. ROOT_STATUS Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-18 | Reserved | 0 | Reserved |
| 17 | PME_PEND | 0 | PME is pending. |
| 16 | PME_STATUS | 0 | Indicates that PME was asserted by the PME Requester. |
| 15-0 | PME_REQ_ID | 0 | ID of the last PME Requester. |

### 19.5.7.10 DEV_CAP2

The device capabilities 2 register (DEV_CAP2) is described in the figure and table below.

**Figure 19-124. DEV_CAP2 Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|
| Reserved | | CMPL_TO_DIS_SUPP | CMPL_TO_EN | |
| R-0 | | R-1 | R-Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-135. DEV_CAP2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Reserved |
| 4 | CMPL_TO_DIS_SUPP | 0 | Completion timeout disable supported |
| 3-0 | CMPL_TO_EN | 0-Fh | Completion timeout ranges supported. Applicable to RC/EP. |

### 19.5.7.11 DEV_STAT_CTRL2

The device status and control register 2 (DEV_STAT_CTRL2) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-125. DEV_STAT_CTRL2 Register**

| 31 | 5 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|
| Reserved | | CMPL_TO_DIS | CMPL_TO | |
| R-0 | | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-136. DEV_STAT_CTRL2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Reserved |
| 4 | CMPL_TO_DIS | 0 | Completion timeout disabled |
| 3-0 | CMPL_TO | 0-Fh | Completion timeout value |

### 19.5.7.12 LINK_CTRL2

The link control register 2 (LINK_CTRL2) is described in the figure and table below.

**Figure 19-126. LINK_CTRL2 Register**

| 31 | 17 | 16 |
|---|---|---|
| Reserved | | DE_EMPH |
| R-0 | | R-1 |

| 15 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| Reserved | | POLL_DEEMPH | CMPL_SOS | ENTR_MOD_COMPL | TX_MARGIN | |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |

| 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| TX_MARGIN | SEL_DEEMPH | HW_AUTO_SPEED_DIS | ENTR_COMPL | TGT_SPEED | |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-2h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-137. LINK_CTRL2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-17 | Reserved | 0 | Reserved |
| 16 | DE_EMPH | 0 | Current De-emphasis level |
| 15-13 | Reserved | 0 | Reserved |
| 12 | POLL_DEEMPH | 0 | DE-emphasis level in polling-compliance state |
| 11 | CMPL_SOS | 0 | Compliance SOS |
| 10 | ENTR_MOD_COMPL | 0 | Enter modified compliance |
| 9-7 | TX_MARGIN | 0-7h | Value of non-de-emphasized voltage level at transmitter pins |
| 6 | SEL_DEEMPH | 0 | Selectable De-emphasis (0 for 6 dB and 1 for 3.5 dB) |
| 5 | HW_AUTO_SPEED_DIS | 0 | Hardware autonomous speed disable |
| 4 | ENTR_COMPL | 0 | Enter compliance |
| 3-0 | TGT_SPEED | 0-Fh | Gen-1 is 1h and Gen-2 is 2h. |

## *19.5.8 PCI Express Extended Capabilities Registers*


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-138. PCI Express Extended Capabilities Registers**

| Offset | Acronym | Section |
|---|---|---|
| 100h | PCIE_EXTCAP | Section 19.5.8.1 |
| 104h | PCIE_UNCERR | Section 19.5.8.2 |
| 108h | PCIE_UNCERR_MASK | Section 19.5.8.3 |
| 10Ch | PCIE_UNCERR_SVRTY | Section 19.5.8.4 |
| 110h | PCIE_CERR | Section 19.5.8.5 |
| 114h | PCIE_CERR_MASK | Section 19.5.8.6 |
| 118h | PCIE_ACCR | Section 19.5.8.7 |
| 11Ch | HDR_LOG0 | Section 19.5.8.8 |
| 120h | HDR_LOG1 | Section 19.5.8.9 |
| 124h | HDR_LOG2 | Section 19.5.8.10 |
| 128h | HDR_LOG3 | Section 19.5.8.11 |
| 12Ch | RC_ERR_CMD | Section 19.5.8.12 |
| 130h | RC_ERR_ST | Section 19.5.8.13 |
| 134h | ERR_SRC_ID | Section 19.5.8.14 |

### 19.5.8.1 PCIE_EXTCAP

The PCI express extended capabilities header (PCIE_EXTCAP) is described in the figure and table below.

**Figure 19-127. PCIE_EXTCAP Register**

| 31 | 20 | 19 | 16 |
|---|---|---|---|
| NEXT_CAP | | EXT_CAP_VER | |
| R-0 | | R-1 | |

| 15 | 0 |
|---|---|
| EXT_CAP_ID | |
| R-1 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 19-139. PCIE_EXTCAP Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | NEXT_CAP | 0 | Next Capability Offset |
| 19-16 | EXT_CAP_VER | 0-Fh | Extended Capability Version |
| 15-0 | EXT_CAP_ID | 0-FFFFh | PCIe Extended Capability ID |

### 19.5.8.2 PCIE_UNCERR

The PCI express uncorrectable error status register (PCIE_UNCERR) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-128. PCIE_UNCERR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | UR_ERR_ST | ECRC_ERR_ST | MTLP_ERR_ST | RCVR_OF_ST | UCMP_ST |
| R-0 | | | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CMPL_ABRT_ST | CMPL_TMOT_ST | FCP_ERR_ST | PSND_TLP_ST | Reserved | | | |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R-0 | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | SRPS_DN_ST | DLP_ERR_ST | Reserved | | | |
| R-0 | | R-0 | R/W1C-0 | R-0 | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-140. PCIE_UNCERR Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-21 | Reserved | 0 | Reserved |
| 20 | UR_ERR_ST | 0 | Unsupported Request Error Status |
| 19 | ECRC_ERR_ST | 0 | ECRC Error Status |
| 18 | MTLP_ERR_ST | 0 | Malformed TLP Status |
| 17 | RCVR_OF_ST | 0 | Receiver Overflow Status |
| 16 | UCMP_ST | 0 | Unexpected Completion Status |
| 15 | CMPL_ABRT_ST | 0 | Completer Abort Status |
| 14 | CMPL_TMOT_ST | 0 | Completion Timeout Status |
| 13 | FCP_ERR_ST | 0 | Flow Control Protocol Error Status |
| 12 | PSND_TLP_ST | 0 | Poisoned TLP Status |
| 11-6 | Reserved | 0 | Reserved |
| 5 | SRPS_DN_ST | 0 | Surprise Down Error Status (not supported) |
| 4 | DLP_ERR_ST | 0 | Data Link Protocol Error Status |
| 3-0 | Reserved | 0 | Reserved |

### 19.5.8.3 PCIE_UNCERR_MASK

The PCI express uncorrectable error mask register (PCIE_UNCERR_MASK) is described in the figure and table below.

**Figure 19-129. PCIE_UNCERR_MASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | UR_ERR_MSK | ECRC_ERR_MSK | MTLP_ERR_MSK | RCVR_OF_MSK | UCMP_MSK |
| R-0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CMPL_ABRT_MSK | CMPL_TMOT_MSK | FCP_ERR_MSK | PSND_TLP_MSK | Reserved | | | |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | SRPS_DN_MSK | DLP_ERR_MSK | Reserved | | | |
| R-0 | | R-0 | R/W-0 | R-0 | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-141. PCIE_UNCERR_MASK Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-21 | Reserved | 0 | Reserved |
| 20 | UR_ERR_MSK | 0 | Unsupported Request Error Mask |
| 19 | ECRC_ERR_MSK | 0 | ECRC Error Mask |
| 18 | MTLP_ERR_MSK | 0 | Malformed TLP Mask |
| 17 | RCVR_OF_MSK | 0 | Receiver Overflow Mask |
| 16 | UCMP_MSK | 0 | Unexpected Completion Mask |
| 15 | CMPL_ABRT_MSK | 0 | Completer Abort Mask |
| 14 | CMPL_TMOT_MSK | 0 | Completion Timeout Mask |
| 13 | FCP_ERR_MSK | 0 | Flow Control Protocol Error Mask |
| 12 | PSND_TLP_MSK | 0 | Poisoned TLP Mask |
| 11-6 | Reserved | 0 | Reserved |
| 5 | SRPS_DN_MSK | 0 | Surprise Down Error Mask (not supported) |
| 4 | DLP_ERR_MSK | 0 | Data Link Protocol Error Mask |
| 3-0 | Reserved | 0 | Reserved |

### 19.5.8.4 PCIE_UNCERR_SVRTY

The PCI express uncorrectable error severity register (PCIE_UNCERR_SVRTY) is described in the figure and table below.

**Figure 19-130. PCIE_UNCERR_SVRTY Register**

| 31 | 24 |
|---|---|
| Reserved | |
| R-0 | |

| 23 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|
| Reserved | | UR_ERR_SVRTY | ECRC_ERR_SVRTY | MTLP_ERR_SVRTY | RCVR_OF_SVRTY | UCMP_SVRTY |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 8 |
|---|---|---|---|---|---|
| CMPL_ABRT_SVRTY | CMPL_TMOT_SVRTY | FCP_ERR_SVRTY | PSND_TLP_SVRTY | Reserved | |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | |

| 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | SRPS_DN_SVRTY | DLP_ERR_SVRTY | Reserved | |
| R-0 | | R-0 | R/W-0 | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-142. PCIE_UNCERR_SVRTY Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-21 | Reserved | 0 | Reserved |
| 20 | UR_ERR_SVRTY | 0 | Unsupported Request Error Severity |
| 19 | ECRC_ERR_SVRTY | 0 | ECRC Error Severity |
| 18 | MTLP_ERR_SVRTY | 0 | Malformed TLP Severity |
| 17 | RCVR_OF_SVRTY | 0 | Receiver Overflow Severity |
| 16 | UCMP_SVRTY | 0 | Unexpected Completion Severity |
| 15 | CMPL_ABRT_SVRTY | 0 | Completer Abort Severity |
| 14 | CMPL_TMOT_SVRTY | 0 | Completion Timeout Severity |
| 13 | FCP_ERR_SVRTY | 0 | Flow Control Protocol Error Severity |
| 12 | PSND_TLP_SVRTY | 0 | Poisoned TLP Severity |
| 11-6 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

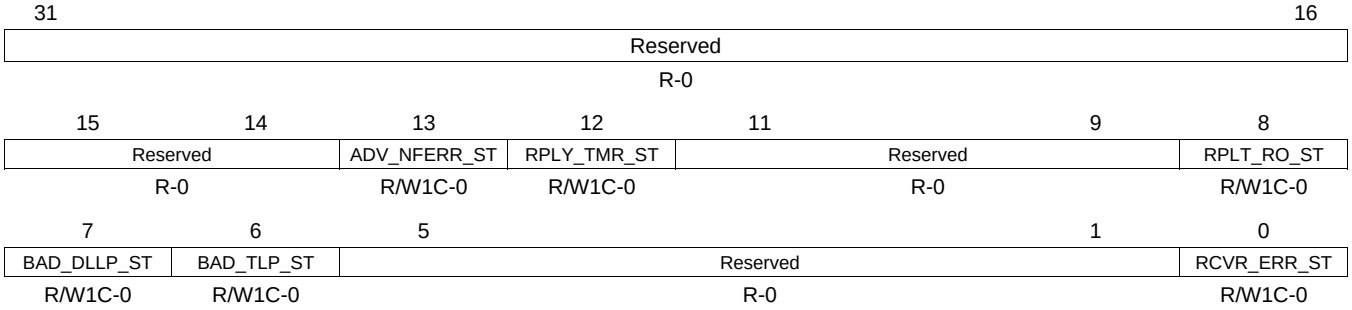
**Table 19-142. PCIE_UNCERR_SVRTY Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 5 | SRPS_DN_SVRTY | 0 | Surprise Down Error Severity (not supported) |
| 4 | DLP_ERR_SVRTY | 0 | Data Link Protocol Error Severity |
| 3-0 | Reserved | 0 | Reserved |

### 19.5.8.5 PCIE_CERR

The PCI express correctable error status register (PCIE_CERR) is described in the figure and table below.

**Figure 19-131. PCIE_CERR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

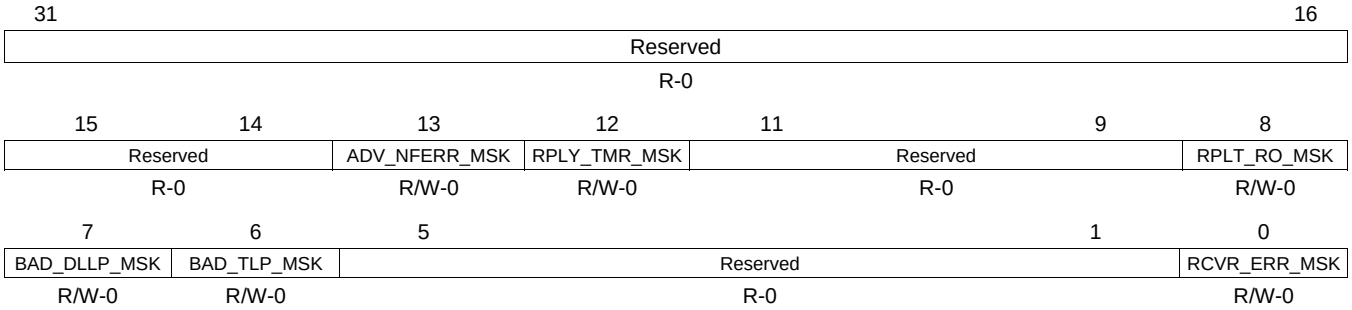
**Table 19-143. PCIE_CERR Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-14 | Reserved | 0 | Reserved |
| 13 | ADV_NFERR_ST | 0 | Advisory Non-fatal Error Mask |
| 12 | RPLY_TMR_ST | 0 | Reply Timer Timeout Mask |
| 11-9 | Reserved | 0 | Reserved |
| 8 | RPLT_RO_ST | 0 | REPLAY_NUM Rollover Mask |
| 7 | BAD_DLLP_ST | 0 | Bad DLLP Mask |
| 6 | BAD_TLP_ST | 0 | Bad TLP Mask |
| 5-1 | Reserved | 0 | Reserved |
| 0 | RCVR_ERR_ST | 0 | Receiver Error Mask |

### 19.5.8.6 PCIE_CERR_MASK

PCI express correctable error mask register (PCIE_CERR_MASK) is described in the figure and table below.

**Figure 19-132. PCIE_CERR_MASK Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

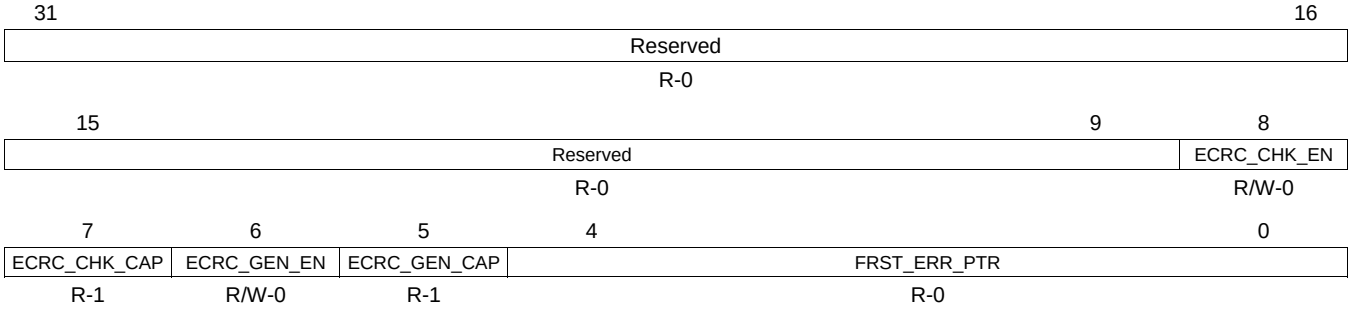
**Table 19-144. PCIE_CERR_MASK Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-14 | Reserved | 0 | Reserved |
| 13 | ADV_NFERR_MSK | 0 | Advisory Non-fatal Error Mask |
| 12 | RPLY_TMR_MSK | 0 | Reply Timer Timeout Mask |
| 11-9 | Reserved | 0 | Reserved |
| 8 | RPLT_RO_MSK | 0 | REPLAY_NUM Rollover Mask |
| 7 | BAD_DLLP_MSK | 0 | Bad DLLP Mask |
| 6 | BAD_TLP_MSK | 0 | Bad TLP Mask |
| 5-1 | Reserved | 0 | Reserved |
| 0 | RCVR_ERR_MSK | 0 | Receiver Error Mask |

### 19.5.8.7 PCIE_ACCR

The PCI express advanced capabilities and control register (PCIE_ACCR) is described in the figure and table below.

**Figure 19-133. PCIE_ACCR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

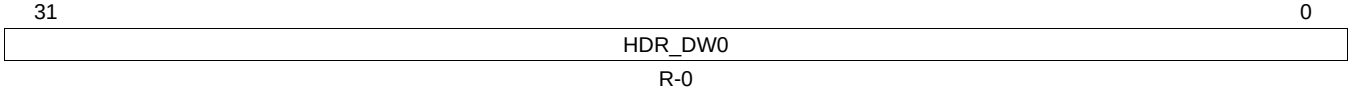
**Table 19-145. PCIE_ACCR Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | Reserved |
| 8 | ECRC_CHK_EN | 0 | ECRC Check Enable |
| 7 | ECRC_CHK_CAP | 0 | ECRC Check Capable |
| 6 | ECRC_GEN_EN | 0 | ECRC Generation Enable |
| 5 | ECRC_GEN_CAP | 0 | ECRC Generation Capable |
| 4-0 | FRST_ERR_PTR | 0 | First Error Pointer |

### 19.5.8.8 HDR_LOG0

The header log0 register (HDR_LOG0) is described in the figure and table below.

**Figure 19-134. HDR_LOG0 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-146. HDR_LOG0 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | HDR_DW0 | 0-FFFF FFFFh | First DWORD of Header for a detected error |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.8.9 HDR_LOG1

The header log 1 register (HDR_LOG1) is described in the figure and table below.

**Figure 19-135. HDR_LOG1 Register**

| 31 | 0 |
|---|---|
| HDR_DW1 | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-147. HDR_LOG1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | HDR_DW1 | 0-FFFF FFFFh | Second DWORD of Header for a detected error |

### 19.5.8.10 HDR_LOG2

The header log2 register (HDR_LOG2) is described in the figure and table below.

**Figure 19-136. HDR_LOG2 Register**

| 31 | 0 |
|---|---|
| HDR_DW2 | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-148. HDR_LOG2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | HDR_DW2 | 0-FFFF FFFFh | Third DWORD of Header for a detected error |

### 19.5.8.11 HDR_LOG3

The header log 3 register (HDR_LOG3) is described in the figure and table below.

**Figure 19-137. HDR_LOG3 Register**

| 31 | 0 |
|---|---|
| HDR_DW3 | |
| R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-149. HDR_LOG3 Register Field Descriptions**

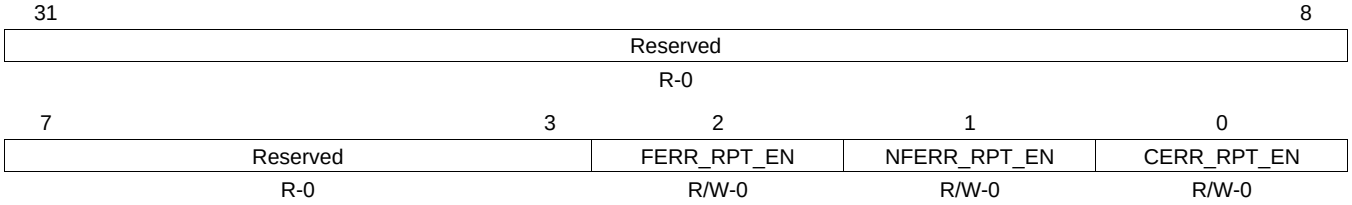
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | HDR_DW3 | 0-FFFF FFFFh | Fourth DWORD of Header for a detected error |

### 19.5.8.12 RC_ERR_CMD

The root error command register (RC_ERR_CMD) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-138. RC_ERR_CMD Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

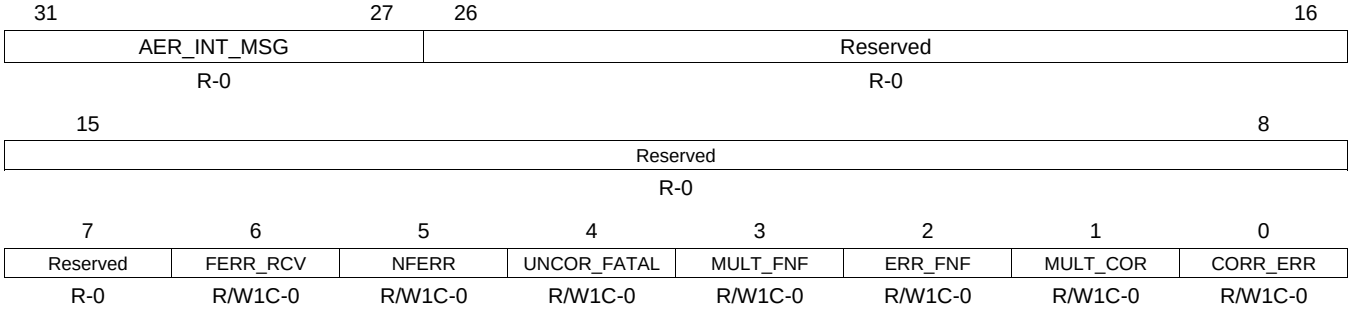
**Table 19-150. RC_ERR_CMD Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2 | FERR_RPT_EN | 0 | Fatal Error Reporting Enable |
| 1 | NFERR_RPT_EN | 0 | Nonfatal Error Reporting Enable |
| 0 | CERR_RPT_EN | 0 | Correctable Error Reporting Enable |

### 19.5.8.13 RC_ERR_ST

The root error status Register (RC_ERR_ST) is described in the figure and table below.

**Figure 19-139. RC_ERR_ST Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-151. RC_ERR_ST Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-27 | AER_INT_MSG | 0-1Fh | AER Interrupt Message Number. Writable through internal bus interface |
| 26-7 | Reserved | 0 | Reserved |
| 6 | FERR_RCV | 0 | Fatal Error Messages Received |
| 5 | NFERR | 0 | 0h Non-Fatal Error Messages Received |
| 4 | UNCOR_FATAL | 0 | First Uncorrectable Fatal |
| 3 | MULT_FNF | 0 | Multiple ERR_FATAL/NONFATAL Received |
| 2 | ERR_FNF | 0 | ERR_FATAL/NONFATAL Received |
| 1 | MULT_COR | 0 | Multiple ERR_COR Received |
| 0 | CORR_ERR | 0 | ERR_COR Received |

### 19.5.8.14 ERR_SRC_ID

The error source identification register (ERR_SRC_ID) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-140. ERR_SRC_ID Register**

| 31 | 16 |
|---|---|
| FNF_SRC_ID | |
| RO-0 | |

| 15 | 0 |
|---|---|
| CORR_SRC_ID | |
| RO-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-152. ERR_SRC_ID Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | FNF_SRC_ID | 0-FFFFh | Error Fatal or Non-Fatal source identification |
| 15-0 | CORR_SRC_ID | 0-FFFFh | Error Correctable source identification |

## *19.5.9 Port Logic Registers*

**Table 19-153. Port Logic Registers**

| Offset | Acronym | Section |
|---|---|---|
| 700h | PL_ACKTIMER | Section 19.5.9.1 |
| 704h | PL_OMSG | Section 19.5.9.2 |
| 708h | PL_FORCE_LINK | Section 19.5.9.3 |
| 70Ch | ACK_FREQ | Section 19.5.9.4 |
| 710h | PL_LINK_CTRL | Section 19.5.9.5 |
| 714h | LANE_SKEW | Section 19.5.9.6 |
| 718h | SYM_NUM | Section 19.5.9.7 |
| 71Ch | SYMTIMER_FLTMASK | Section 19.5.9.8 |
| 720h | FLT_MASK2 | Section 19.5.9.9 |
| 728h | DEBUG0 | Section 19.5.9.10 |
| 72Ch | DEBUG1 | Section 19.5.9.11 |
| 80Ch | PL_GEN2 | Section 19.5.9.12 |

### 19.5.9.1 PL_ACKTIMER Register

The Ack latency time and replay timer register (PL_ACKTIMER) is described in the figure and table below.

**Figure 19-141. PL_ACKTIMER Register**

| 31 | 16 |
|---|---|
| RPLY_LIMT | |
| R/W1S-C0h | |

| 15 | 0 |
|---|---|
| RND_TRP_LMT | |
| R/W1S-40h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-154. PL_ACKTIMER Register Field Descriptions**

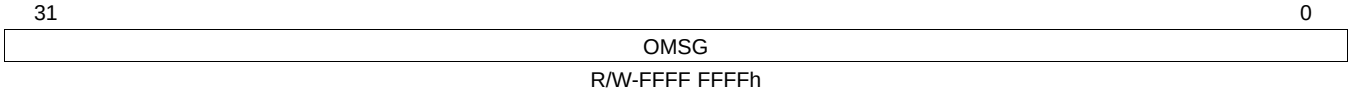
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | RPLY_LIMT | 0-FFFFh | Replay Time Limit |
| 15-0 | RND_TRP_LMT | 0-FFFFh | Round Trip Latency Time Limit |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.9.2 PL_OMSG Register

The other message register (PL_OMSG ) is described in the figure and table below.

**Figure 19-142. PL_OMSG Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

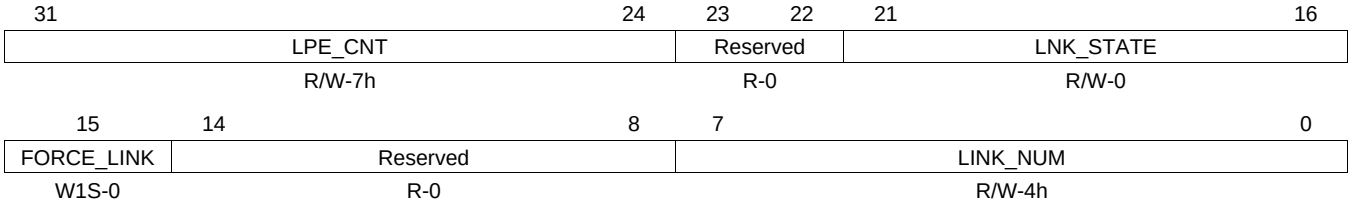
**Table 19-155. PL_OMSG Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | OMSG | 0-FFFF FFFFh | Other Message Register. It can be used to send a specific PCI Express message in which case this register is programmed with the payload and bit 0 of Port Link Control Register is set to transmit the message. |

### 19.5.9.3 PL_FORCE_LINK Register

The port force link register (PL_FORCE_LINK) is described in the figure and table below.

**Figure 19-143. PL_FORCE_LINK Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

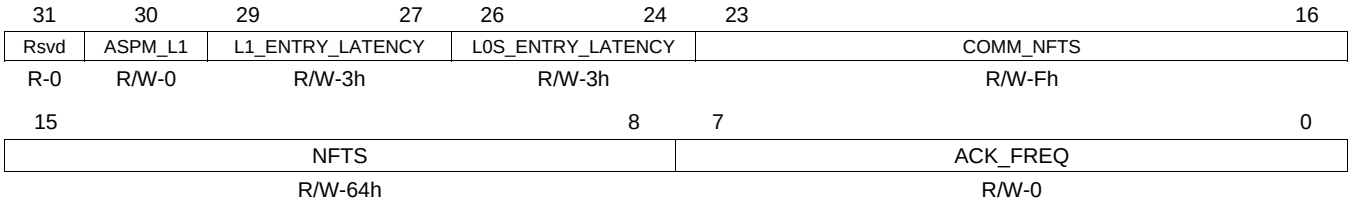
**Table 19-156. PL_FORCE_LINK Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | LPE_CNT | 0-FFh | Low Power Entrance Count |
| 23-22 | Reserved | 0 | Reserved |
| 21-16 | LNK_STATE | 0-3Fh | Link State |
| 15 | FORCE_LINK | 0 | Force Link |
| 14-8 | Reserved | 0 | Reserved |
| 7-0 | LINK_NUM | 0-FFh | Link Number |

### 19.5.9.4 ACK_FREQ Register

The ack frequency register (ACK_FREQ) is described in the figure and table below.

**Figure 19-144. ACK_FREQ Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-157. ACK_FREQ Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved |
| 30 | ASPM_L1 | 0 | Set to allow entering ASPM L1 even when link partner did not to L0s. When cleared, the ASPM L1 state is entered only after idle period during which both RX and TX are in L0s. |
| 29-27 | L1_ENTRY_LATENCY | 0-7h | L1 Entrance Latency. The latency is set to 2^L1_ENTRY_LATENCY microseconds with the max being 64 microseconds. |
| 26-24 | L0S_ENTRY_LATENCY | 0-7h | L0s Entrance Latency. The latency is set to L0S_ENTRY_LATENCY+1 microseconds. Maximum is 7 microseconds. |
| 23-16 | COMM_NFTS | 0-FFh | Number of Fast Training Sequences when common clock is used and when transitioning from L0s to L0. |
| 15-8 | NFTS | 0-FFh | Number of Fast Training Sequences to be transmitted when transitioning from L0s to L0. Value of zero is not supported. |
| 7-0 | ACK_FREQ | 0-FFh | Ack Frequency. Default is to wait until 255 Ack DLLPs are pending before it is sent. |

### 19.5.9.5 PL_LINK_CTRL Register

The port link control register (PL_LINK_CTRL) is described in the figure and table below.

**Figure 19-145. PL_LINK_CTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | LNK_MODE | | | | | |
| R-0 | | | | | | | | | | R/W-3h | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | LNK_RATE | | | |
| R-0 | | | | R/W-1 | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FLNK_MODE | Reserved | DLL_EN | Reserved | RST_ASRT | LPBK_EN | SCRM_DIS | OMSG_REQ |
| R/W-0 | R-0 | R/W-1 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-158. PL_LINK_CTRL Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-22 | Reserved | 0 | Reserved |
| 21-16 | LNK_MODE | 0-3Fh | Link Mode Enable. |
| | | 0 | Reserved |
| | | 1h | ×1 |
| | | 2h | Reserved |
| | | 3h | ×2 |
| | | 4h-6h | Reserved |
| | | 7h | ×4 |
| | | 8h-Eh | Reserved |
| | | Fh | ×8 |
| | | 10h-1Eh | Reserved |
| | | 1Fh | ×16 |
| | | 20h-3Eh | Reserved |
| | | 3Fh | ×32 |
| 15-12 | Reserved | 0 | Reserved |
| 11-8 | LNK_RATE | 0-Fh | Default Link Rate. For 2.5, it is 1h. This register does not affect any functionality. |
| 7 | FLNK_MODE | 0 | Fast Link Mode |
| 6 | Reserved | 0 | Reserved |
| 5 | DLL_EN | 0 | DLL Link Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-158. PL_LINK_CTRL Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 4 | Reserved | 0 | Reserved |
| 3 | RST_ASRT | 0 | Reset Assert |
| 2 | LPBK_EN | 0 | Loopback Enable |
| 1 | SCRM_DIS | 0 | Scramble Disable |
| 0 | OMSG_REQ | 0 | Other Message Request |

### 19.5.9.6 LANE_SKEW Register

The lane skew register (LANE_SKEW is described in the figure and table below.

**Figure 19-146. LANE_SKEW Register**

| 31 | 30 | 26 | 25 | 24 |
|---|---|---|---|---|
| L2L_DESKEW | Reserved | | ACK_DISABLE | FC_DISABLE |
| R/W-0 | R-0 | | R/W-0 | R/W-0 |

| 23 | 0 |
|---|---|
| LANE_SKEW | |
| R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-159. LANE_SKEW Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | L2L_DESKEW | 0 | Disable Lane to Lane deskew |
| 30-26 | Reserved | 0 | Reserved |
| 25 | ACK_DISABLE | 0 | Disable Ack and Nak DLLP transmission. |
| 24 | FC_DISABLE | 0 | Flow control disable. Set to disable transmission of Flow Control DLLPs. |
| 23-0 | LANE_SKEW | 0-FF FFFFh | Insert Lane Skew for Transmit. The value is in units of one symbol time. Thus a value 02h will force a skew of two symbol times for that lane. Max allowed is five symbol times. This 24 bit field is used for programming skew for eight lanes with three bits per lane. |

### 19.5.9.7 SYM_NUM Register

The symbol number register (SYM_NUM) is described in the figure and table below.

**Figure 19-147. SYM_NUM Register**

| 31 | 29 | 28 | 24 | 23 | 19 | 18 | 16 |
|---|---|---|---|---|---|---|---|
| MAX_FUNC | | FCWATCH_TIMER | | ACK_LATENCY_TIMER | | REPLAY_TIMER | |
| R/W-0 | | R/W-0 | | R/W-0 | | R/W-1 | |

| 15 | 14 | 13 | 11 | 10 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| REPLAY_TIMER | | Reserved | | SKP_COUNT | | NUM_TS2_SYMBOLS | | TS_COUNT | |
| R/W-0 | | R-0 | | R/W-3h | | R/W-Ah | | R/W-Ah | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-160. SYM_NUM Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-29 | MAX_FUNC | 0-7h | Configuration requests at function numbers above this value will result in UR response. |
| 28-24 | FCWATCH_TIMER | 0-1Fh | Timer Modifier for Flow Control Watchdog Timer. Increases the timer value for Flow control watchdog timer in increments of 16 clock cycles. |
| 23-19 | ACK_LATENCY_TIMER | 0-1Fh | Timer Modifier for Ack/Nak latency timer in increments of 64 clock periods. |
| 18-14 | REPLAY_TIMER | 0-1Fh | Timer for replaying TLPs in increments of 64 clock cycles. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 19-160. SYM_NUM Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 13-11 | Reserved | 0 | Reserved |
| 10-8 | SKP_COUNT | 0-7h | Number of SKP symbols |
| 7-4 | NUM_TS2_SYMBOLS | 0-Fh | Number of TS2 symbols. This field does not affect any functionality. |
| 3-0 | TS_COUNT | 0-Fh | Number of TS symbols. Set the number of TS identifier symbols that are sent in TS1 and TS2 ordered sets. |

### 19.5.9.8 SYMTIMER_FLTMASK Register

The symbol timer and filter mask register (SYMTIMER_FLTMASK) is described in the figure and table below.

**Figure 19-148. SYMTIMER_FLTMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| F1_CFG_DROP | F1_IO_DROP | F1_MSG_DROP | F1_CPL_ECRC_DROP | F1_ECRC_DROP | F1_CPL_LEN_TEST | F1_CPL_ATTR_TEST | F1_CPL_TC_TEST |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| F1_CPL_FUNC_TEST | F1_CPL_REQID_TEST | F1_CPL_TAGERR_TEST | F1_LOCKED_RD_AS_UR | F1_CFG1_RE_AS_US | F1_UR_OUT_OF_BAR | F1_UR_POISON | F1_UR_FUN_MISMATCH |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 11 | 10 0 |
|---|---|---|
| FC_WDOG_DISABLE | Reserved | SKP_VALUE |
| R/W-0 | R-0 | R/W-500h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-161. SYMTIMER_FLTMASK Register Field Descriptions**

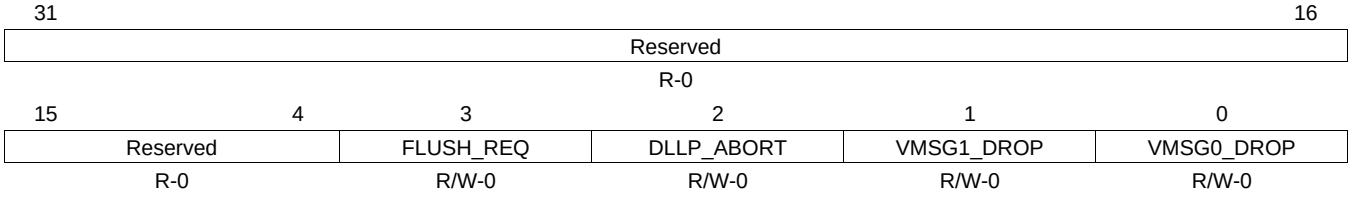
| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | F1_CFG_DROP | 0 | Set to allow CFG TLPs on RC |
| 30 | F1_IO_DROP | 0 | Set to allow IO TLPs on RC |
| 29 | F1_MSG_DROP | 0 | Set to allow MSG TLPs on RC |
| 28 | F1_CPL_ECRC_DROP | 0 | Set to allow Completion TLPs with ECRC to pass up |
| 27 | F1_ECRC_DROP | 0 | Set to allow TLPs with ECRC to pass up |
| 26 | F1_CPL_LEN_TEST | 0 | Set to mask length match for received completion TLPs |
| 25 | F1_CPL_ATTR_TEST | 0 | Set to mask attribute match on received completion TLPs |
| 24 | F1_CPL_TC_TEST | 0 | Set to mask traffic class match on received completion TLPs |
| 23 | F1_CPL_FUNC_TEST | 0 | Set to mask function match for received completion TLPs |
| 22 | F1_CPL_REQID_TEST | 0 | Set to mask request ID match for received completion TLPs |
| 21 | F1_CPL_TAGERR_TEST | 0 | Set to mask tag error rules for received completion TLPs |
| 20 | F1_LOCKED_RD_AS_UR | 0 | Set to treat locked read TLPs as supported for EP, UR for RC. |
| 19 | F1_CFG1_RE_AS_US | 0 | Set to treat type 1 CFG TLPs as supported for EP and UR for RC |
| 18 | F1_UR_OUT_OF_BAR | 0 | Set to treat out-of-BAR TLPs as supported requests |
| 17 | F1_UR_POISON | 0 | Set to treat poisoned TLPs as supported requests |
| 16 | F1_UR_FUN_MISMATCH | 0 | Set to treat mismatched TLPs as supported |
| 15 | FC_WDOG_DISABLE | 0 | Disable FC Watchdog Timer |
| 14-11 | Reserved | 0 | Reserved |
| 10-0 | SKP_VALUE | 0-7FFh | Number of symbol times to wait between transmitting SKP ordered sets. For example, for a setting of 1536 decimal, the wait will be for 1537 symbol times. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.9.9 FLT_MASK2 Register

The filter mask two register (FLT_MASK2) is described in the figure and table below.

**Figure 19-149. FLT_MASK2 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

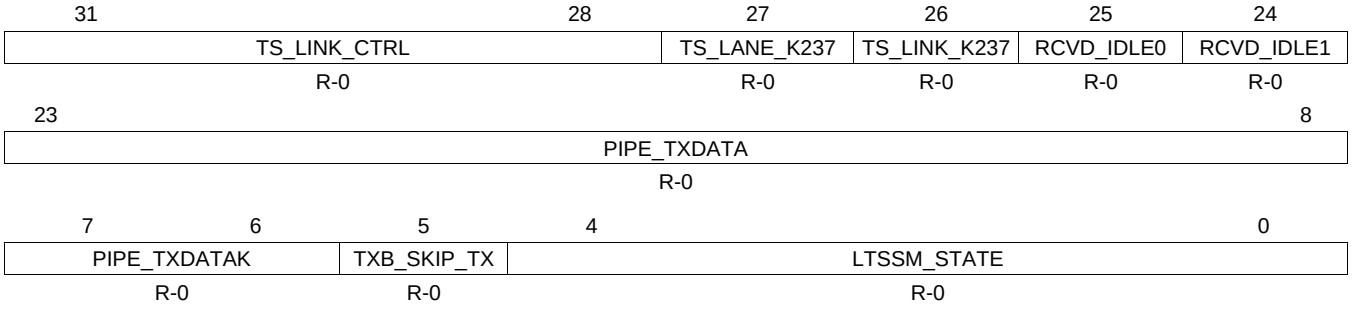
**Table 19-162. FLT_MASK2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | FLUSH_REQ | 0 | Set to enable the filter to handle flush request |
| 2 | DLLP_ABORT | 0 | Set to disable DLLP Abort for unexpected CPL |
| 1 | VMSG1_DROP | 0 | Set to disable dropping of Vendor MSG Type 1. When cleared, Vendor MSG Type 1 will be passed to internal bus interface |
| 0 | VMSG0_DROP | 0 | Set to disable dropping of Vendor MSG Type 0 with UR reporting. When cleared, Vendor MSG Type 0 will be passed to internal bus interface |

### 19.5.9.10 DEBUG0 Register

The debug 0 register (DEBUG0) is described in the figure and table below.

**Figure 19-150. DEBUG0 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-163. DEBUG0 Register Field Descriptions**

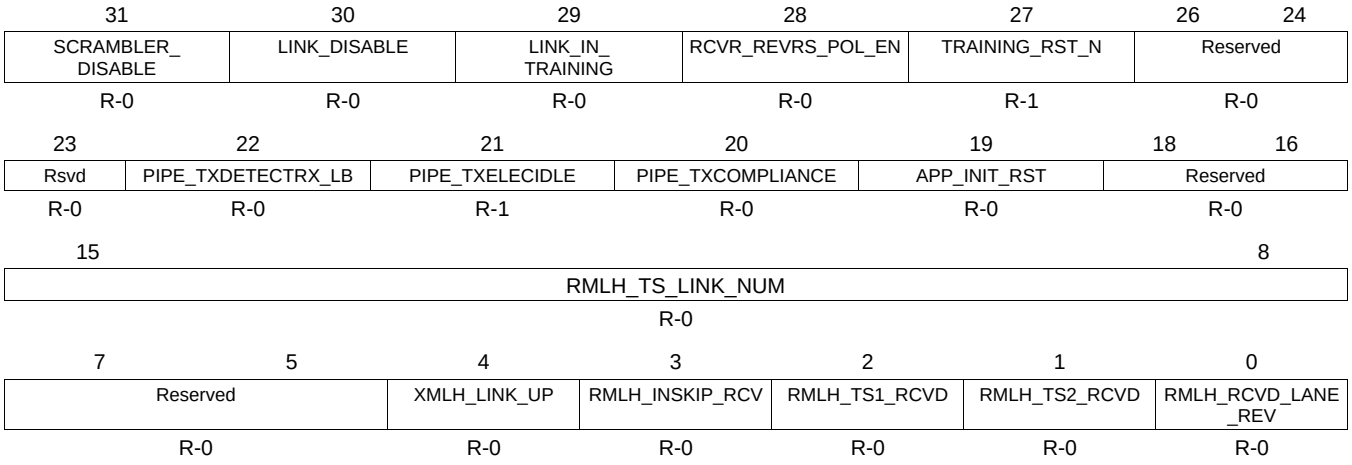
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | TS_LINK_CTRL | 0-Fh | Link control bits advertised by link partner |
| 27 | TX_LANE_K237 | 0 | Currently receiving k237 (PAD) in place of lane number |
| 26 | TS_LINK_K237 | 0 | Currently receiving k237 (PAD) in place of link number |
| 25 | RCVD_IDLE0 | 0 | Receiver is receiving logical idle |
| 24 | RCVD_IDLE1 | 0 | 2nd symbol is also idle (16bit PHY interface only) |
| 23-8 | PIPE_TXDATA | 0-FFFFh | PIPE Transmit data. Reset value is zero but changes at every clock after that. |
| 7-6 | PIPE_TXDATAK | 0-3h | PIPE transmit K indication |
| 5 | TXB_SKIP_TX | 0 | A skip ordered set has been transmitted |
| 4-0 | LTSSM_STATE | 0-1Fh | LTSSM current state. A read value of 11h indicates the up status of the Link. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 19.5.9.11 DEBUG1 Register

The debug 1 register (DEBUG1) is described in the figure and table below.

**Figure 19-151. DEBUG1 Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-164. DEBUG1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | SCRAMBLER_DISABLE | 0 | Scrambling disabled for the link |
| 30 | LINK_DISABLE | 0 | LTSSM in DISABLE state. Link inoperable |
| 29 | LINK_IN_TRAINING | 0 | LTSSM performing link training |
| 28 | RCVR_REVRS_POL_EN | 0 | LTSSM testing for polarity reversal |
| 27 | TRAINING_RST_N | 0 | LTSSM-negotiated link reset |
| 26-23 | Reserved | 0 | Reserved |
| 22 | PIPE_TXDETECTRX_LB | 0 | PIPE receiver detect/loopback request |
| 21 | PIPE_TXELECIDLE | 0 | PIPE transmit electrical idle request |
| 20 | PIPE_TXCOMPLIANCE | 0 | PIPE transmit compliance request |
| 19 | APP_INIT_RST | 0 | Application request to initiate training reset |
| 18-16 | Reserved | 0 | Reserved |
| 15-8 | RMLH_TS_LINK_NUM | 0-FFh | Link number advertised/confirmed by link partner |
| 7-5 | Reserved | 0 | Reserved |
| 4 | XMLH_LINK_UP | 0 | LTSSM reports PHY link up |
| 3 | RMLH_INSKIP_RCV | 0 | Receiver reports skip reception |
| 2 | RMLH_TS1_RCVD | 0 | TS1 training sequence received (pulse) |
| 1 | RMLH_TS2_RCVD | 0 | TS2 training sequence received (pulse) |
| 0 | RMLH_RCVD_LANE_REV | 0 | Receiver detected lane reversal |

### 19.5.9.12 PL_GEN2 Register

The Gen2 register (PL_GEN2) is described in the figure and table below.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 19-152. PL_GEN2 Register**

| 31 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|
| Reserved | | DEEMPH | CFG_TX_CMPL | CFG_TX_SWING | DIR_SPD | LN_EN |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| LN_EN | | NUM_FTS | |
| R/W-2h | | R/W-Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 19-165. PL_GEN2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-21 | Reserved | 0 | Reserved |
| 20 | DEEMPH | 0 | Set De-emphasis level for upstream ports |
| 19 | CFG_TX_CMPL | 0 | Configure TX Compliance Receive Bit |
| 18 | CFG_TX_SWING | 0 | Configure PHY TX Swing |
| 17 | DIR_SPD | 0 | Directed Speed Change |
| 16-8 | LN_EN | 0-1FFh | Lane Enable. It is 1 for ×1, 2 for ×2, and so on. |
| 7-0 | NUM_FTS | 0-FFh | Number of fast training sequences. |


Copyright © 2011–2016, Texas Instruments Incorporated

# Real-Time Clock (RTC)

This chapter provides a functional presentation of real-time clock (RTC).

**Topic** **Page**

**20.1 Introduction** ..... **3368**
**20.2 Architecture** ..... **3369**
**20.3 Registers** ..... **3376**


Copyright © 2011–2016, Texas Instruments Incorporated

## 20.1 Introduction

### 20.1.1 Overview

The real-time clock is a precise timer which can generate interrupts on intervals specified by the user. Interrupts can occur every second, minute, hour, or day. The clock itself can track the passage of real time for durations of several years, provided it has a sufficient power source the whole time.

The basic purpose for the RTC is to keep time of day. The other equally important purpose of RTC is for Digital Rights management. Some degree of tamper proofing is needed to ensure that simply stopping, resetting, or corrupting the RTC does not go unnoticed so that if this occurs, the application can re-acquire the time of day from a trusted source. The final purpose of RTC is to wake the rest of chip up from a power down state.

Alarms are available to interrupt the CPU at a particular time, or at periodic time intervals, such as once per minute or once per day. In addition, the RTC can interrupt the CPU every time the calendar and time registers are updated, or at programmable periodic intervals.

### 20.1.2 Features

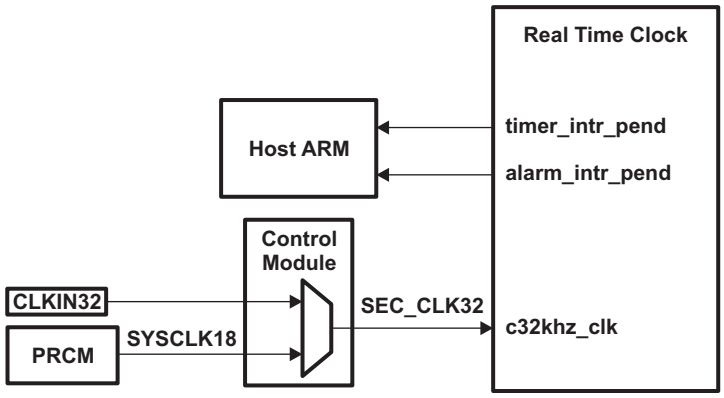
The real-time clock (RTC) provides the following features:

- 100-year calendar (xx00 to xx99)
- Counts seconds, minutes, hours, day of the week, date, month, and year with leap year compensation
- Binary-coded-decimal (BCD) representation of time, calendar, and alarm
- 12-hour clock mode (with AM and PM) or 24-hour clock mode
- Alarm interrupt
- Periodic interrupt
- Single interrupt to the CPU
- Supports external 32.768-kHz crystal or external clock source of the same frequency

### 20.1.3 Functional Block Diagram

Figure 20-1 shows the RTC module block diagram. Figure 20-2 shows a functional block diagram of the RTC.

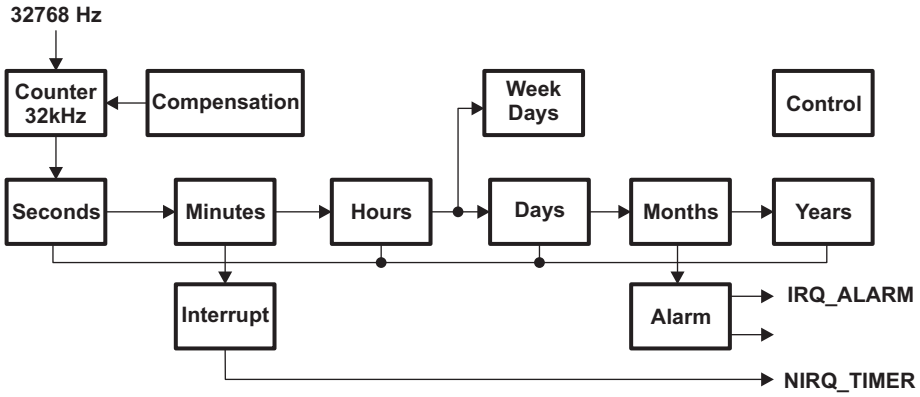
**Figure 20-1. RTC Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 20-2. RTC Functional Block Diagram**




## 20.2 Architecture

This section defines the module interrupt capabilities and requirements.

### *20.2.1 Clock Source*

The clock reference for the RTC is an internal 32.768-kHz crystal or an external clock source of the same frequency. The source for the RTC reference clock may be provided by a crystal or by an external clock source. The RTC has an internal oscillator buffer to support direct operation with a crystal. The crystal is connected between pins RTC_XI and RTC_XO. RTC_XI is the input to the on-chip oscillator and RTC_XO is the output from the oscillator back to the crystal. For more information about the RTC crystal connection, see your device-specific data manual.

An external 32.768-kHz clock source may be used instead of a crystal. In such a case, the clock source is connected to RTC_XI, and RTC_XO is left unconnected.

If the RTC is not used, the RTC_XI pin should be held low and RTC_XO should be left unconnected. The RTC_disable bit in the control register (CTRL_REG) can be set to save power; however, the RTC_disable bit should not be cleared once it has been set. If the application requires the RTC module to stop and continue, the STOP_RTC bit in CTRL should be used instead.

### *20.2.2 Signal Descriptions*

Table 20-1 lists the signals and their descriptions for the RTC.

**Table 20-1. RTC Signals**

| Signal | I/O | Description |
|---|---|---|
| RTC_XI | I | RTC time base input signal. RTC_XI can either be driven with a 32.768-kHz reference clock, or RTC_XI and RTC_XO can be connected to an external crystal. This signal is the input to the RTC internal oscillator. |
| RTC_XO | O | RTC time base output signal. RTC_XO is the output from the RTC internal oscillator. If a crystal is not used as the time base for RTC_XI, RTC_XO should be left unconnected. |


Copyright © 2011–2016, Texas Instruments Incorporated

## 20.2.3 Interrupt Support

### 20.2.3.1 CPU Interrupts

The RTC generates two interrupt outputs:

- timer_intr is a timer interrupt.
- alarm_intr is an alarm interrupt.

**NOTE:** Both interrupt outputs support high-level and high-pulse.

### 20.2.3.2 Interrupt Description

#### 20.2.3.2.1 Timer Interrupt (timer_intr)

The timer interrupt can be generated periodically: every second, every minute, every hour, or every day (see INTERRUPTS_REG[1:0] for a description of how to set this up). The IT_TIMER bit of the interrupt register enables this interrupt. The timer interrupt is active-low.

The RTC_STATUS_REG[5:2] are only updated at each new interrupt and occur according to Table 20-2. For example, bit 2 (SEC) will always be set when one second has passed. It will also be set when one minute has passed since the completion of one minute also marks the completion of one second (from 59 seconds to 60 seconds). The same holds true for hours and days: each of them will also correspond to the passing of a second.

Conversely, bit 5 (DAY) will always be set when a day has passed. It might also be set when an hour, minute, or second has passed. However, this only occurs when the elapsed hour, minute, or second corresponds to the start of a new day.

**Table 20-2. Interrupt Trigger Events**

| | One day has passed | One hour has passed | One minute has passed | One second has passed |
|---|---|---|---|---|
| STATUS_REG[5] (DAY) | 1 | 0/1[(1)] | 0/1[(1)] | 0/1[(1)] |
| STATUS_REG[4] (HOUR) | 1 | 1 | 0/1[(1)] | 0/1[(1)] |
| STATUS_REG[3] (MIN) | 1 | 1 | 1 | 0/1[(1)] |
| STATUS_REG[2] (SEC) | 1 | 1 | 1 | 1 |

(1) This event is only triggered when the elapsed time unit (for example, Day) corresponds to the passage of another unit (for example, Seconds). For example, when the clock ticks from 00:23:59:59 (days : hours : minutes : seconds) to 01:00:00:00.

#### 20.2.3.2.2 Alarm Interrupt (alarm_intr)

The alarm interrupt can be generated when the time set into TC ALARM registers is exactly the same as in the TC registers. This interrupt is then generated if the IT_ALARM bit of the interrupts register is set. This interrupt is low-level sensitive. RTC_STATUS_REG[6] indicates that IRQ_ALARM_CHIP has occurred. This interrupt is disabled by writing '1' into the RTC_STATUS_REG[6].

To set up an alarm:

- Modify the ALARM_SECONDS, ALARM_MINUTES, ALARM_HOURS, ALARM_DAY, ALARM_WEEK, ALARM_MONTH, and ALARM_YEAR registers to the exact time you want an alarm to generate.
- Set the IT_ALARM bit in the RTC_INTERRUPTS register to enable the alarm interrupt.


Copyright © 2011–2016, Texas Instruments Incorporated

## 20.2.4 Programming/Usage Guide

### 20.2.4.1 Time/Calendar Data Format

The time and calendar data in the RTC is stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. Although most of the time/calendar registers have 4 bits assigned to each BCD digit, some of the register fields are shorter since the range of valid numbers may be limited. For example, only 3 bits are required to represent the day of the week (WEEKS_REG) since only BCD numbers 1 through 7 are required. The following time and calendar registers are supported (BCD Format):

Note that the ALARM registers which share the names above also share the same BCD formatting.

- SECOND - Second Count (00-59)
- MINUTE - Minute Count (00-59)
- HOUR - Hour Count (12HR: 01-12; 24HR: 00-23)
- DAY - Day of the Month Count (01-31)
- WEEK - Day of the Week (0-6: SUN = 0)
- MONTH - Month Count (01-12; JAN = 1)
- YEAR - Year Count (00-99)

### 20.2.4.2 Register Access

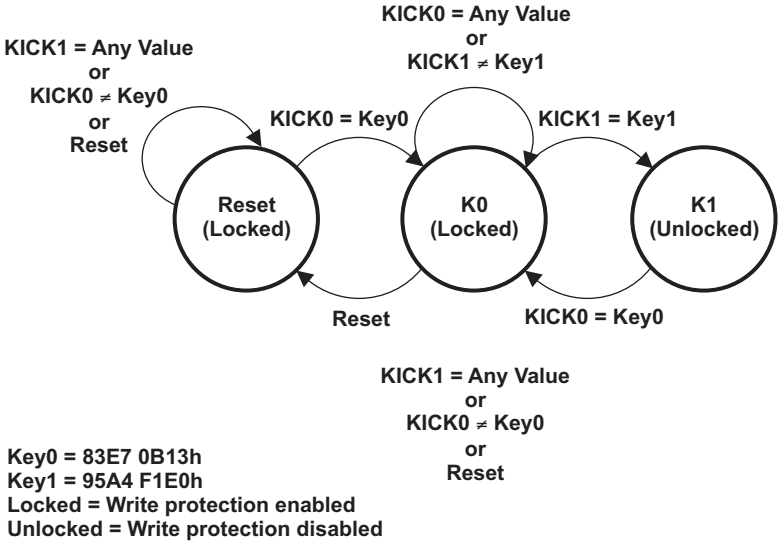
The three register types are as follows and each has its own access constraints:

- TC and TC alarm registers
- General registers
- Compensation registers

### 20.2.4.3 OCP MMR Spurious WRT Protection

The module also contains a kicker mechanism (Figure 20-3) to prevent any spurious writes from changing the register values. This mechanism requires two MMR writes to the Kick0 and Kick1 registers with exact data values before the kicker lock mechanism is released. Once released, the MMRs are writeable. The Kick0 data is 83E7 0B13h; the Kick1 data is 95A4 F1E0h. Note that it remains in an unlocked state until an OCP reset or invalid data pattern is written to one of the Kick0 or Kick1 registers.

**Figure 20-3. Kick Register State Machine Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

- S0 is the Reset/Idle state
- S1 is an OCP wrt cycle of 83E7 0B13h at Kick0 completed state
- S2 is the UNLOCK MMR WRT state
- S0 -> S1 when OCP wrt cycle of 83E7 0B13h at Kick0
- S1 -> S2 when OCP wrt cycle of 95A4 F1E0h at Kick1
- S1 -> S0 when OCP reset event
- S2 -> S0 when OCP reset event OR OCP wrt cycle of NOT 83E7 0B13h at Kick0 OR OCP wrt cycle at Kick1
- S2 ->S1 when OCP wrt cycle of 83E7 0B13h at Kick0

#### 20.2.4.4 Reading the Timer/Calendar (TC) Registers

The TC registers have a read-show register. The reading of the Seconds register will update all of the TC registers. For example, the Year will only get updated on a reading of the Seconds register. The time/calendar registers are updated every second as the time changes. During a read of the SECONDS register, the RTC copies the current values of the time/date registers into shadow read registers. This isolation assures that the CPU can capture all the time/date values at the moment of the SECONDS read request and not be subject to changing register values from time updates.

If desired, the RTC also provides a one-time-triggered minute-rounding feature to round the MINUTE:SECOND registers to the nearest minute (with zero seconds). This feature is enabled by setting the ROUND_30S bit in the control register (CTRL); the RTC automatically rounds the time values to the nearest minute upon the next read of the SECONDS register.

**NOTE:** Software should always read the Seconds register first. However, the software does not have to poll any status bit to determine when to read the TC registers. Table 20-3 defines the TC set that gets shadowed.

**Table 20-3. RTC Register Names and Values**

| Time Unit | Range | Remarks |
|---|---|---|
| Year | 00 to 99 | |
| Month | 01 to 12 | |
| Day | 01 to 31 | Months 1, 3, 5, 7, 8, 10, 12 |
| | 01 to 30 | Months 4, 6, 9, 11 |
| | 01 to 29 | Month 2 (leap year) |
| | 01 to 28 | Month 2 (common year) |
| Week | 00 to 06 | Day of week |
| Hour | 00 to 23 | 24 hour mode |
| | 01 to 12 | AM/PM mode |
| Minute | 00 to 59 | |
| Seconds | 00 to 59 | |

##### *20.2.4.4.1 Rounding Seconds*

Time can be rounded to the closest minute, by setting the ROUND_30S bit of the control register. When this bit is set, TC values are set to the closest minute value at the next second. The ROUND_30S bit will be automatically cleared when rounding time is performed.

**Example:**

- If current time is 10H59M45S, round operation will change time to 11H00M00S.
- If current time is 10H59M29S, round operation will change time to 10H59M00S.


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.2.4.5 Modifying the TC Registers

To write correct data from/to the TC and TC alarm registers and read the TC alarm registers, the ARM must first read the BUSY bit of the STATUS register until BUSY is equal to zero. Once the BUSY flag is zero, there is a 15 µs access period in which the ARM can program the TC and TC alarm registers. Once the 15 µs access period passes, the BUSY flag has to be read again from the STATUS register as described previously. If the ARM accesses the TC registers outside of the access period, then the access is not guaranteed.

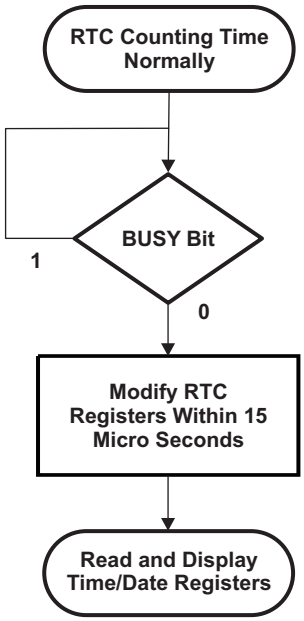
The ARM can access the STATUS_REG and CTRL_REG at any time, with the exception of CTRL_REG[5] which can only be changed when the RTC is stopped. The ARM can stop the RTC by clearing the STOP_RTC bit of the control register. After clearing this bit, the RUN bit in the STATUS_REG (bit 1) needs to be checked to verify the RTC has in fact stopped. Once this is confirmed, the TC values can be updated. After the values have been updated, the RTC can be re-started by resetting the STOP_RTC bit.

**NOTE:** After writing to a TC register, the user must wait 4 OCP clock cycles before reading the value from the register. If this wait time is not observed and the TC register is accessed, then old data will be read from the register.

**CAUTION**

In order to remove any possibility of interrupting the register's read process, thus introducing a potential risk of violating the authorized 15-microsecond access period, it is strongly recommended that you disable all incoming interrupts during the register read process.

**Figure 20-4. Flow Control for Updating RTC Registers**




#### *20.2.4.5.1 General Registers*

The ARM can access the STATUS_REG and the CTRL_REG at any time (except the CTRL_REG[5] bit which must be changed only when the RTC is stopped). For the INTERRUPTS_REG, the ARM should respect the available access period to prevent spurious interrupt.


Copyright © 2011–2016, Texas Instruments Incorporated

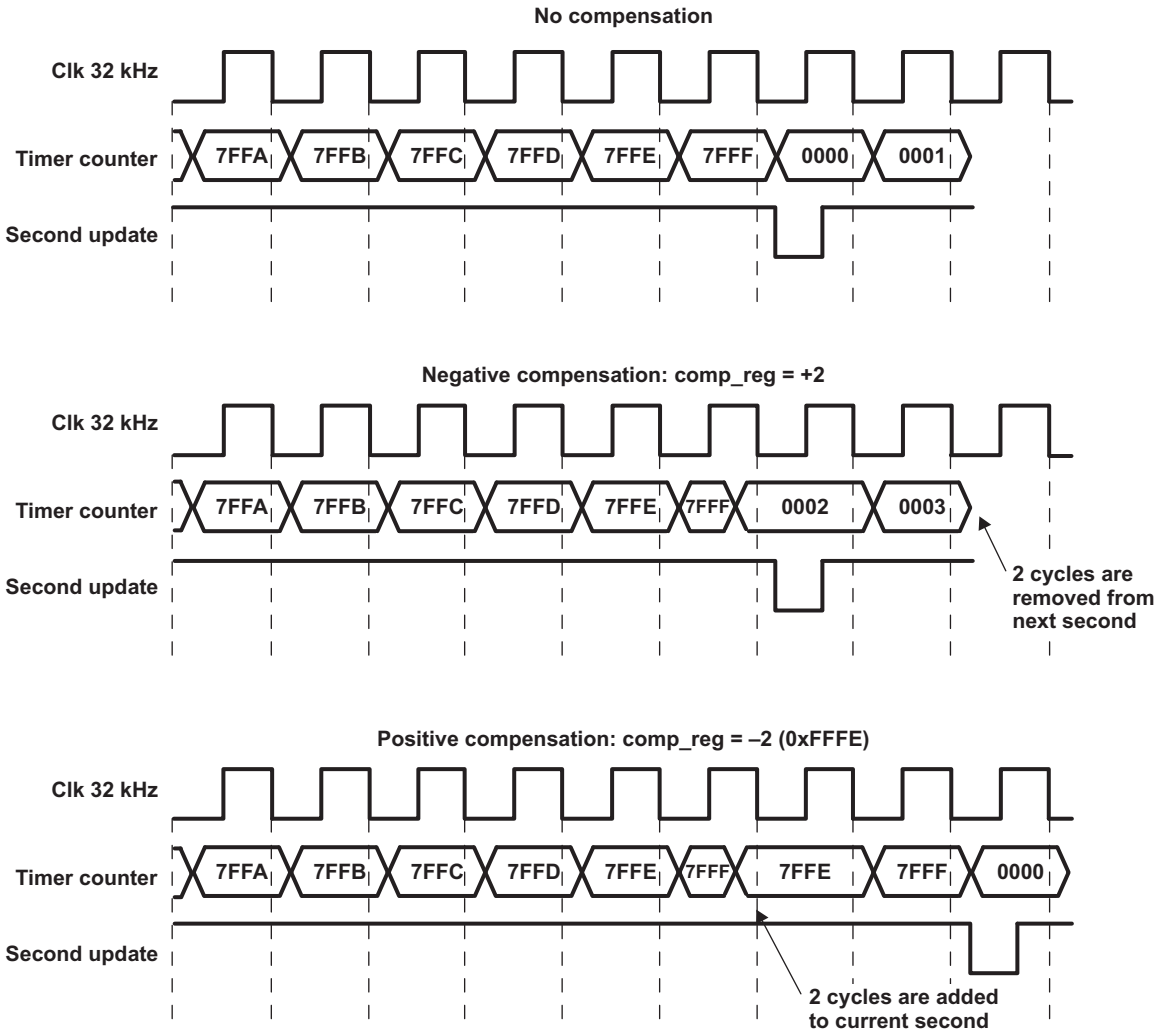
The RTC_DISABLE bit of the CTRL register must only be used to completely disable the RTC function. When this bit is set, the 32 kHz clock is gated, and the RTC is frozen. From this point, resetting this bit to zero can lead to unexpected behavior. In order to save power, this bit should only be used if the RTC function is unwanted in the application.

### 20.2.4.6 Crystal Compensation

To compensate for any inaccuracy of the 32 kHz oscillator, the ARM can perform a calibration of the oscillator frequency, calculate the drift compensation versus one-hour period, and load the compensation registers with the drift compensation value. Auto compensation is enabled by AUTO_COMP_EN bit in the RTC_CTRL register. If the COMP_REG value is positive, compensation occurs after the second change event. COMP_REG cycles are removed from the next second. If the COMP_REG value is negative, compensation occurs before the second change event. COMP_REG cycles are added to the current second. This enables compensation with a 1 32-kHz period accuracy each hour. The waveform below summarizes positive and negative compensation effect.

Access to the COMP_MSB_REG and COMP_LSB_REG registers must respect the available access period. These registers should not be updated during compensation (first second of each hour), but it is alright to update them during the second preceding a compensation event. For example, the ARM could load the compensation value into these registers after each hour event, during an available access period.

**Figure 20-5. Compensation Illustration**





Copyright © 2011–2016, Texas Instruments Incorporated

### 20.2.5 Scratch Registers

The RTC provides three general-purpose registers (SCRATCHx_REG) that can be used to store 32-bit words -- these registers have no functional purpose for the RTC. Software using the RTC may find the SCRATCHx registers to be useful in indicating RTC states. For example, the SCRATCHx_REG registers may be used to indicate write-protection lock status or unintentional power downs. To indicate write-protection, the software should write a unique value to one of the SCRATCHx_REG registers when write-protection is disabled and another unique value when write-protection is enabled again. In this way, the lock-status of the registers can be determined quickly by reading the SCRATCH register. To indicate unintentional power downs, the software should write a unique value to one of the SCRATCHx_REG registers when RTC is configured and enabled. If the RTC is unintentionally powered down, the value written to the SCRATCH register is cleared. For more information, see the registers section.

### 20.2.6 Power Management

The RTC supports the power idle protocol. It has two SWakeup ports: one for the alarm event and one for a timer event.

When the RTC is in IDLE mode, the OCP clock is turned off and the 32 kHz clock remains on. The time and calendar continue to count in IDLE mode. When the RTC is placed back in FUNCTIONAL mode, the TC registers can be read.

The Alarm SWakeup event can be used to wakeup the RTC when it is in IDLE state. In order to do so, the alarm needs to be set and enabled before RTC enters the IDLE state. Once this is done, the SWakeup will occur when the alarm event triggers.

**NOTE:** Since SWakeup is not periodic, using it to wake up the RTC when in IDLE state is not recommended. Please use Alarm SWakeup instead.

### 20.2.7 Reset Considerations

When the device is initially powered on, the RTC may issue spurious interrupt signals to the CPU. To avoid issues, a software reset should be performed on the RTC module before the CPU interrupt controller is initialized. Also, see the register summary table in the registers section for reset states of each register.


Copyright © 2011–2016, Texas Instruments Incorporated

## 20.3 Registers

Table 20-4 lists the memory-mapped registers for the RTC. See your device-specific data manual for the memory address of these registers.

**Table 20-4. Real-Time Clock (RTC) Registers**

| Address Offset | Acronym | Register Description |
|---|---|---|
| 0h | SECONDS_REG | Seconds Register |
| 4h | MINUTES_REG | Minutes Register |
| 8h | HOURS_REG | Hours Register |
| Ch | DAYS_REG | Day of the Month Register |
| 10h | MONTHS_REG | Month Register |
| 14h | YEARS_REG | Year Register |
| 18h | WEEK_REG | Day of the Week Register |
| 20h | ALARM_SECONDS_REG | Alarm Seconds Register |
| 24h | ALARM_MINUTES_REG | Alarm Minutes Register |
| 28h | ALARM_HOURS_REG | Alarm Hours Register |
| 2Ch | ALARM_DAYS_REG | Alarm Days Register |
| 30h | ALARM_MONTHS_REG | Alarm Months Register |
| 34h | ALARM_YEARS_REG | Alarm Years Register |
| 40h | RTC_CTRL_REG | Control Register |
| 44h | RTC_STATUS_REG | Status Register |
| 48h | RTC_INTERRUPTS_REG | Interrupt Enable Register |
| 4Ch | RTC_COMP_LSB_REG | Compensation (LSB) Register |
| 50h | RTC_COMP_MSB_REG | Compensation (MSB) Register |
| 54h | RTC_OSC_REG | Oscillator Register |
| 60h | RTC_SCRATCH0_REG | Scratch 0 Register (General-Purpose) |
| 64h | RTC_SCRATCH1_REG | Scratch 1 Register (General-Purpose) |
| 68h | RTC_SCRATCH2_REG | Scratch 2 Register (General-Purpose) |
| 6Ch | KICK0R | Kick 0 Register (Write Protect) |
| 70h | KICK1R | Kick 1 Register (Write Protect) |
| 74h | RTC_REVISION | Revision Register |
| 78h | RTC_SYSCONFIG | System Configuration Register |
| 7Ah | RTC_IRQWAKEEN_0 | Wakeup Enable Register |


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.1 Seconds Register (SECONDS_REG)

The SECONDS_REG is used to program the required seconds value of the current time. Seconds are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. If the seconds value is 45, then the value of SEC0 is 5 and value of SEC1 is 4.

**Figure 20-6. Seconds Register (SECONDS_REG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | SEC1 | | SEC0 | |
| R-0 | | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-5. Seconds Register (SECONDS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved. |
| 6-4 | SEC1 | 0-5h | 2nd digit of seconds, Range is 0 to 5 |
| 3-0 | SEC0 | 0-9h | 1st digit of seconds, Range is 0 to 9 |

### 20.3.2 Minutes Register (MINUTES_REG)

The MINUTES_REG is used to program the minutes value of the current time. Minutes are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. If the minutes value is 32, then the value of MIN0 is 2 and value of MIN1 is 3.

**Figure 20-7. Minutes Register (MINUTES_REG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | MIN1 | | MIN0 | |
| R-0 | | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-6. Minutes Register (MINUTES_REG) Field Descriptions**

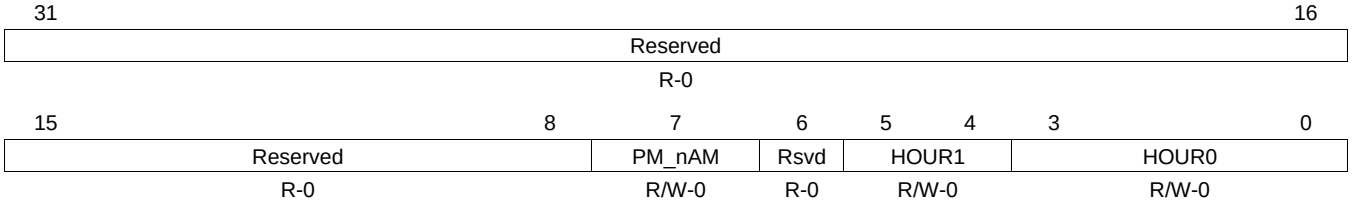
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6-4 | MIN1 | 0-5h | 2nd digit of minutes, Range is 0 to 5 |
| 3-0 | MIN0 | 0-9h | 1st digit of minutes, Range is 0 to 9 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.3 Hours Register (HOURS_REG)

The HOURS_REG is used to program the hours value of the current time. Hours are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. In 24Hr time mode if you want to set the hour as 18, then HOUR0 is set as 8 and HOUR1 is set as 1.

**Figure 20-8. Hours Register (HOURS_REG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

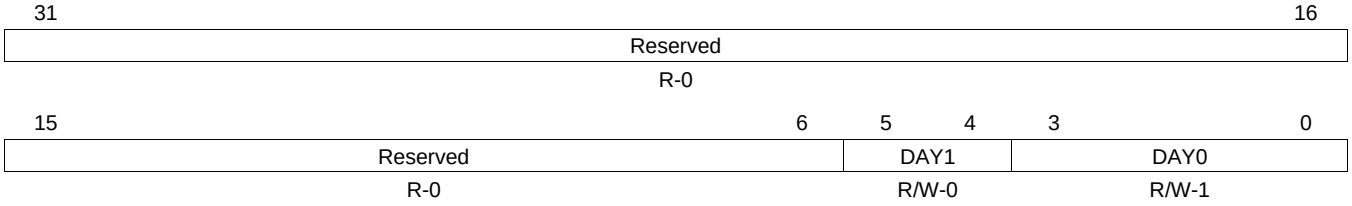
**Table 20-7. Hours Register (HOURS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7 | PM_nAM | | Only used in PM_AM mode (otherwise 0) |
| | | 0 | AM |
| | | 1 | PM |
| 6 | Reserved | 0 | Reserved |
| 5-4 | HOUR1 | 0-2h | 2nd digit of hours, Range is 0 to 2 |
| 3-0 | HOUR0 | 0-9h | 1st digit of hours, Range is 0 to 9 |

### 20.3.4 Days of the Month Register (DAYS_REG)

The DAYS_REG is used to program the day of the month value of the current date. Days are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. If the day value of the date is 28, DAY0 is set as 8 and DAY1 is set as 2.

**Figure 20-9. Days of the Month Register (DAYS_REG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-8. Day of the Month (DAYS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5-4 | DAY1 | 0-3h | 2nd digit of days, Range is 0 to 3 |
| 3-0 | DAY0 | 0-9h | 1st digit of days, Range is 0 to 9 |


Copyright © 2011–2016, Texas Instruments Incorporated

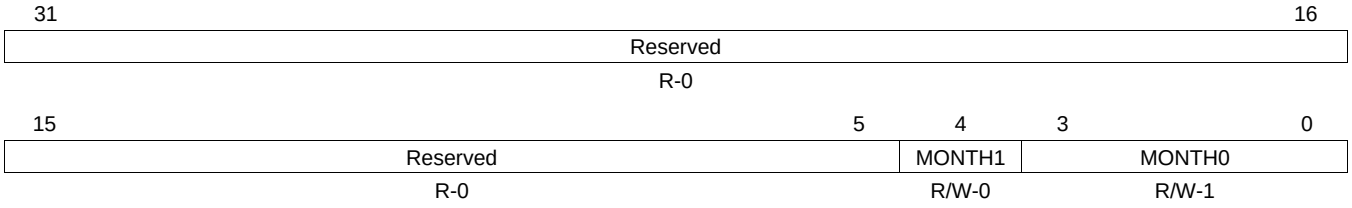
### 20.3.5 Month Register (MONTHS_REG)

The MONTHS_REG is used to set the month in the year value of the current date. Months are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

**NOTE:** Usual notation is taken for month value:

- 01 => January
- 02 => February
- ...
- 12 => December

**Figure 20-10. Month Register (MONTHS_REG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

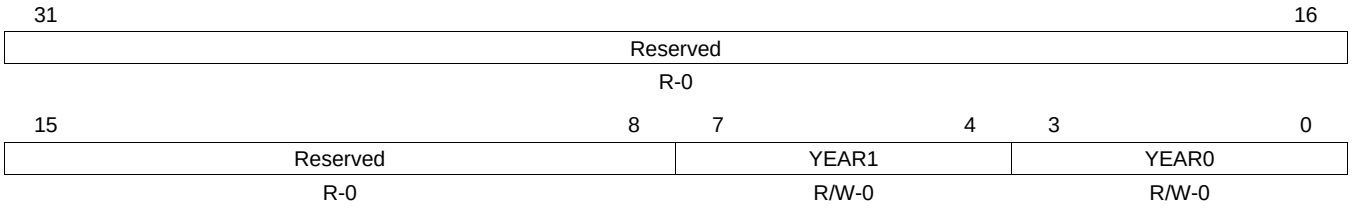
**Table 20-9. Month Register (MONTHS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Reserved |
| 4 | MONTH1 | 0-1h | 2nd digit of months, Range is 0 to 1 |
| 3-0 | MONTH0 | 0-9h | 1st digit of months, Range is 0 to 9 |

### 20.3.6 Year Register (YEARS_REG)

The YEARS_REG is used to program the year value of the current date. The year value is represented by only the last 2 digits and is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The year 1979 is programmed as 79 with YEAR0 set as 9 and YEAR1 set as 7.

**Figure 20-11. Year Register (YEARS_REG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-10. Year Register (YEARS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-4 | YEAR1 | 0-9h | 2nd digit of years, Range is 0 to 9 |
| 3-0 | YEAR0 | 0-9h | 1st digit of years, Range is 0 to 9 |

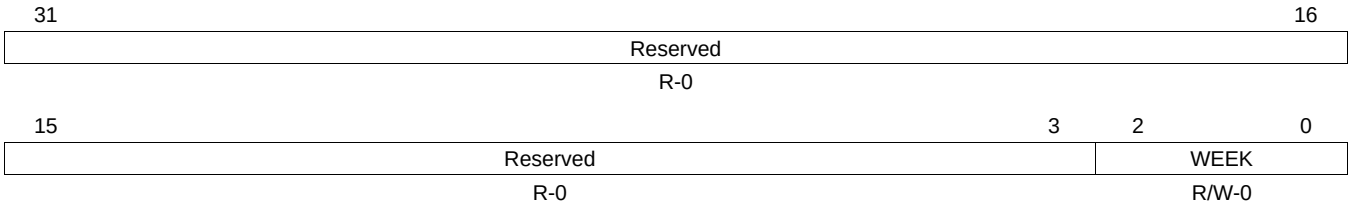

Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.7 Day of the Week Register (WEEKS_REG)

The WEEKS_REG is used to program the day of the week value of the current date. The day of the week is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

**NOTE:** Sunday is treated as 0, Monday 1, and ending at Saturday with 6.

**Figure 20-12. Day of the Week Register (WEEKS_REG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

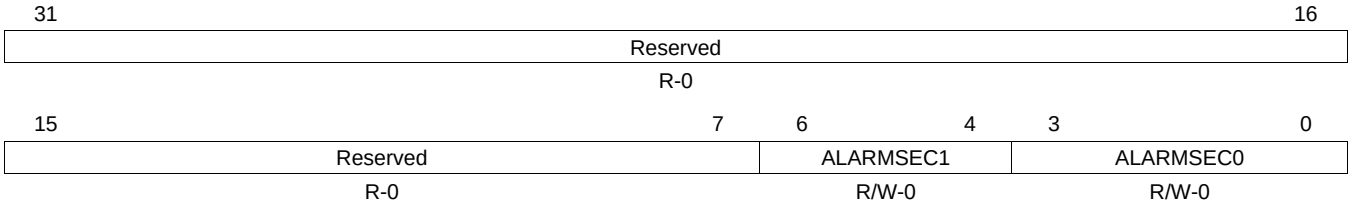
**Table 20-11. Day of the Week (WEEKS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved |
| 2-0 | WEEK | 0-6h | 1st digit of days in a week, Range from 0 (Sunday) to 6 (Saturday) |

### 20.3.8 Alarm Second Register (ALARM_SECONDS_REG)

The ALARM_SECONDS_REG is used to program the second value for the alarm interrupt. Seconds are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

**Figure 20-13. Alarm Second Register (ALARM_SECONDS_REG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-12. Alarm Second Register (ALARM_SECONDS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6-4 | ALARMSEC1 | 0-5h | 2nd digit of seconds, Range is 0 to 5 |
| 3-0 | ALARMSEC0 | 0-9h | 1st digit of seconds, Range is 0 to 9 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.9 Alarm Minute Register (ALARM_MINUTES_REG)

The ALARM_MINUTES_REG is used to program the minute value for the alarm interrupt. Minutes are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

**Figure 20-14. Alarm Minute Register (ALARM_MINUTES_REG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 7 | 6 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | ALARM_MIN1 | | ALARM_MIN0 | |
| R-0 | | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-13. Alarm Minute Register (ALARM_MINUTES_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6-4 | ALARM_MIN1 | 0-5h | 2nd digit of minutes, Range is 0 to 5 |
| 3-0 | ALARM_MIN0 | 0-9h | 1st digit of minutes, Range is 0 to 9 |

### 20.3.10 Alarm Hour Register (ALARM_HOURS_REG)

The ALARM_HOURS_REG is used to program the hour value for the alarm interrupt. Hours are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

**Figure 20-15. Alarm Hour Register (ALARM_HOURS_REG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | ALARM_PM_nAM | Rsvd | ALARM_HOUR1 | | ALARM_HOUR0 | |
| R-0 | | R/W-0 | R-0 | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-14. Alarm Hour Register (ALARM_HOURS_REG) Field Descriptions**

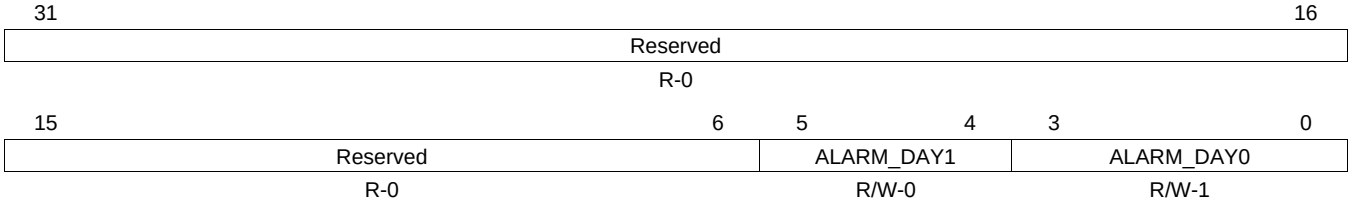
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7 | ALARM_PM_nAM | | Only used in PM_AM mode (otherwise 0) |
| | | 0 | AM |
| | | 1 | PM |
| 6 | Reserved | 0 | Reserved. |
| 5-4 | ALARM_HOUR1 | 0-2h | 2nd digit of hours, Range is 0 to 2 |
| 3-0 | ALARM_HOUR0 | 0-9h | 1st digit of hours, Range is 0 to 9 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.11 Alarm Day of the Month Register (ALARM_DAYS_REG)

The ALARM_DAYS_REG is used to program the day of the month value for the alarm interrupt. Days are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

**Figure 20-16. Alarm Day of the Month (ALARM_DAYS_REG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

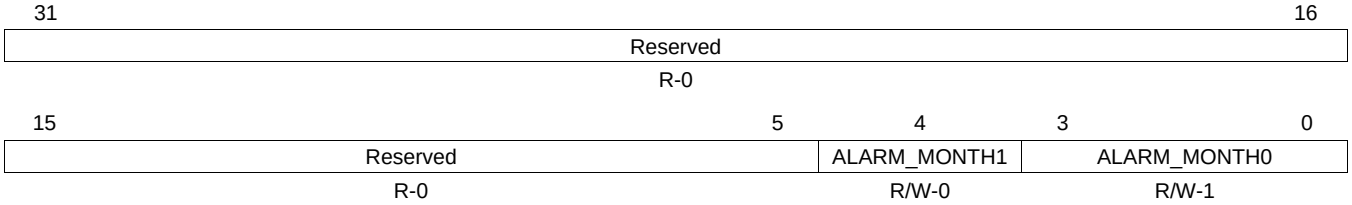
**Table 20-15. Alarm Day of the Month Register (ALARM_DAYS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5-4 | ALARM_DAY1 | 0-3h | 2nd digit for days, Range from 0 to 3 |
| 3-0 | ALARM_DAY0 | 0-9h | 1st digit for days, Range from 0 to 9 |

### 20.3.12 Alarm Month Register (ALARM_MONTHS_REG)

The ALARM_MONTHS_REG is used to program the month in the year value for the alarm interrupt. The month is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

**Figure 20-17. Alarm Month Register (ALARM_MONTHS_REG)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-16. Alarm Month Register (ALARM_MONTHS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | 0 | Reserved |
| 4 | ALARM_MONTH1 | 0-1h | 2nd digit of months, Range from 0 to 1 |
| 3-0 | ALARM_MONTH0 | 0-9h | 1st digit of months, Range from 0 to 9 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.13 Alarm Year Register (ALARM_YEARS_REG)

The ALARM_YEARS_REG is used to program the year for the alarm interrupt. Only the last two digits are used to represent the year and is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

**Figure 20-18. Alarm Year Register (ALARM_YEARS_REG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | ALARM_YEAR1 | | ALARM_YEAR0 | |
| R-0 | | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-17. Alarm Year Register (ALARM_YEARS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-4 | ALARM_YEAR1 | 0-9h | 2nd digit of years, Range from 0 to 9 |
| 3-0 | ALARM_YEAR0 | 0-9h | 1st digit of years, Range from 0 to 9 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.14 Control Register (CTRL_REG)

The CTRL_REG contains the controls to enable/disable RTC, set the 12/24 hour time mode, to enable the 30 second rounding feature, and to STOP/START RTC.

**Figure 20-19. Control Register (CTRL_REG)**

| 31 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | RTC_disable | SET_32_COUNTER | TEST_MODE | MODE_12_24 | AUTO_COMP | ROUND_30S | STOP_RTC |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-18. Control Register (CTRL_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved. |
| 6 | RTC_disable | | Disable RTC module and gate 32-kHz reference clock. |
| | | 0 | RTC enable |
| | | 1 | RTC disable (no 32 kHz clock) |
| 5 | SET_32_COUNTER | | Set the 32-kHz counter with the value stored in the compensation registers when the SET_32_COUNTER bit is set. |
| | | 0 | No action. |
| | | 1 | Set the 32Khz counter with compensation registers value |
| 4 | TEST_MODE | | Test mode. |
| | | 0 | Functional mode |
| | | 1 | Test mode (Auto compensation is enabled when the 32Khz counter reaches its end) |
| 3 | MODE_12_24 | | Enable 12-hour mode for HOURS and ALARMHOURS registers. |
| | | 0 | 24-hr mode |
| | | 1 | 12-hour mode |
| 2 | AUTO_COMP | | Enable oscillator compensation mode. |
| | | 0 | No auto compensation |
| | | 1 | Auto compensation enabled |
| 1 | ROUND_30S | | Enable one-time rounding to nearest minute on next time register read. |
| | | 0 | No update |
| | | 1 | Time is rounded to the nearest minute |
| 0 | STOP_RTC | | Stop the RTC 32-kHz counter. |
| | | 0 | RTC is frozen |
| | | 1 | RTC is running |


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:**

- The SET_32_counter must only be used when the RTC is frozen.
- The RTC_DISABLE bit of the CTRL register must only be used to completely disable the RTC function. When this bit is set, the 32 kHz clock is gated, and the RTC is frozen. From this point, resetting this bit to zero can lead to unexpected behavior. This bit should only be used if the RTC function is unwanted in the application, in order to save power.
- MODE_12_24: It is possible to switch between the two modes at any time without disturbing the RTC; read or write is always performed with the current mode.
- Auto compensation is enabled by AUTO_COMP_EN bit in the RTC_CTRL register. If the COMP_REG value is positive, compensation occurs after the second change event. COMP_REG cycles are removed from the next second. If the COMP_REG value is negative, compensation occurs before the second change event. COMP_REG cycles are added to the current second. This enables it to compensate with one 32-kHz period accuracy each hour.
- ROUND_30S is a toggle bit; ARM can only write 1 and RTC clears it. If the ARM sets the ROUND_30S bit and then reads it, the ARM read 1 until the round-to-the-closet-minute is performed at the next second.
- ARM can stop the RTC by clearing STOP_RTC bit of the control register (owing to internal resynchronization, the RUN bit of the status must be checked to be ensure that the RTC is frozen), then update TC values, and re-start the RTC by resetting STOP_RTC bit.


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.15 Status Register (STATUS_REG)

The RTC STATUS_REG contains bits that signal the status of interrupts, events to the processor. Status for the alarm interrupt and timer events are notified by the register.

**Figure 20-20. Status Register (STATUS_REG)**

| 31 | | | | | | | | 16 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | |
| R-0 | | | | | | | | |
| 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | ALARM | 1D_EVENT | 1H_EVENT | 1M_EVENT | 1S_EVENT | RUN | BUSY |
| R-0 | | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 20-19. Status Register (STATUS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Reserved |
| 6 | ALARM | 0-1 | Indicates that an alarm interrupt has been generated |
| 5 | 1D_EVENT | 0-1 | One day has occurred |
| 4 | 1H_EVENT | 0-1 | One hour has occurred |
| 3 | 1M_EVENT | 0-1 | One minute has occurred |
| 2 | 1S_EVENT | 0-1 | One second has occurred |
| 1 | RUN | | RTC is frozen or is running. |
| | | 0 | RTC is frozen |
| | | 1 | RTC is running |
| 0 | BUSY | | Status of RTC module. |
| | | 0 | Updating event in more than 15 µs |
| | | 1 | Updating event |

**NOTE:**

- The alarm interrupt keeps its low level until the ARM writes 1 in the ALARM bit of the RTC_STATUS_REG register.
- ALARM: This bit will indicate the status of the alarm interrupt. Writing a 1 to the bit clears the interrupt.
- 1D_EVENT1: This bit will indicate if a day event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller.
- 1H_EVENT1: This bit will indicate if an hour event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller.
- 1M_EVENT1: This bit will indicate if a minute event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller.
- 1S_EVENT1: This bit will indicate if a second event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller.
- RUN: This bit will indicate if RTC is frozen or it is running. The RUN bit shows the real state of the RTC. Indeed, because the STOP_RTC signal is resynchronized on 32-kHz clock the action of this bit is delayed.
- BUSY: This bit will give the status of RTC module. The Time and alarm registers can be modified only when this bit is 0.
- The timer interrupt is a negative edge sensitive low-level pulse (1 OCP cycle duration).


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.16 Interrupt Register (INTERRUPTS_REG)

The INTERRUPTS_REG is used to enable or disable RTC from generating interrupts. The timer interrupt and alarm interrupt can be controlled using this register.

**Figure 20-21. Interrupt Register (INTERRUPTS_REG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | IT_ALARM | IT_TIMER | EVERY | |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 20-20. Interrupt Register (INTERRUPTS_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved |
| 3 | IT_ALARM | 0 | Enable one interrupt when the alarm value is reached (TC ALARM registers) by the TC registers |
| 2 | IT_TIMER | | Enable periodic interrupt. |
| | | 0 | Interrupt disabled |
| | | 1 | Interrupt enabled |
| 1-0 | EVERY | 0-3h | Interrupt period. |
| | | 0 | Every second |
| | | 1h | Every minute |
| | | 2h | Every hour |
| | | 3h | Every day |

**NOTE:**

- The ARM must respect the BUSY period to prevent spurious interrupt. To set a period timer interrupt, the respective period value must be set in the EVERY field. For example, to set a periodic timer interrupt for every hour, the EVERY field has to be set to 2. Along with this the IT_TIMER bit also has to be set for the periodic interrupt to be generated.
- IT_ALARM bit has to be set to generate an alarm interrupt.


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.17 Compensation (LSB) Register (COMP_LSB_REG)

The COMP_LSB_REG is used to program the LSB value of the 32 kHz periods to be added to the 32 kHz counter every hour. This is used to compensate the oscillator drift. The COMP_LSB_REG works together with the compensation (MSB) register (COMP_MSB_REG). The AUTOCOMP bit in the control register (CTRL_REG) must be enabled for compensation to take place.

**Figure 20-22. Compensation (LSB) Register (COMP_LSB_REG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RTC_COMP_LSB | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-21. Compensation (LSB) Register (COMP_LSB_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | RTC_COMP_LSB | 0-FFh | Indicates number of 32-kHz periods to be added into the 32-kHz counter every hour |

**NOTE:** This register must be written in two's complement. That means that to add one 32-kHz oscillator period every hour, the ARM must write FFFFh into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. To remove one 32-kHz oscillator period every hour, the ARM must write 0001h into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. The 7FFFh value is forbidden.


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.18 Compensation (MSB) Register (COMP_MSB_REG)

The COMP_MSB_REG is used to program the MSB value of the 32 kHz periods to be added to the 32 kHz counter every hour. This is used to compensate the oscillator drift. The COMP_MSB_REG works together with the compensation (LSB) register (COMP_LSB_REG) to set the hourly oscillator compensation value. The AUTOCOMP bit in the control register (CTRL_REG) must be enabled for compensation to take place.

**Figure 20-23. Compensation (MSB) Register (COMP_MSB_REG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RTC_COMP_MSB | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-22. Compensation (MSB) Register (COMP_MSB_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved |
| 7-0 | RTC_COMP_MSB | 0-FFh | Indicates number of 32-kHz periods to be added into the 32-kHz counter every hour |

**NOTE:** This register must be written in two's complement. That means that to add one 32-kHz oscillator period every hour, the ARM must write FFFFh into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. To remove one 32-kHz oscillator period every hour, the ARM must write 0001h into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. The 7FFFh value is forbidden.


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.19 Oscillator Register (OSC_REG)

The OSC_REG is used to program the oscillator resistance value and the SW_RESET bit can be used to reset the RTC. This is the only way to reset the RTC.

**Figure 20-24. Oscillator Register (OSC_REG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | SWRESET | OSC32KPWRDNR | SWRESPROG | |
| R-0 | | R/W-0 | R/W-0(H/W) | R/W-7h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 20-23. Oscillator Register (OSC_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved |
| 5 | SWRESET | 0-1 | Software reset bit |
| 4 | OSC32KPWRDNR | 0-1 | Control of 32 kHz Oscillator powerdown |
| 3-0 | SWRESPROG | 0-Fh | Value of the oscillator resistance |

**NOTE:** SW_RESET bit is set to 1 to reset the RTC module. This bit is self-clearing, and is always read as 0. CPU must care of interrupt handling before RTC is reset. After setting the SW_RESET bit, you must not access any of the RTC registers for 3 32kHz clock cycles (roughly 2200 OCP cycles).

### 20.3.20 Scratch Registers (SCRATCHx_REG)

The SCRATCH_REG is used to hold some required values for the RTC register.

**Figure 20-25. Scratch Registers (SCRATCHx_REG)**

| 31 | 0 |
|---|---|
| RTCSCRATCHx | |
| R/W-0 | |

LEGEND: R/W = Read/Write; *-n* = value after reset

**Table 20-24. Scratch Registers (SCRATCHx_REG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | RTCSCRATCHx | 0-FFFF FFFFh | Scratch registers, available to program |


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.21 Kick Registers (KICK0R, KICK1R)

The kick registers (KICKnR) are used to enable and disable write protection on the RTC registers. Out of reset, the RTC registers are write-protected. To disable write protection, correct keys must be written to the KICKnR registers.

#### Kick0 Register (KICK0R)

The Kick0 register allows writing to unlock the kick0 data. To disable RTC register write protection, the value of 83E7 0B13h must be written to KICK0R, followed by the value of 95A4 F1E0h written to KICK1R. RTC register write protection is enabled when any value is written to KICK0R.

**Figure 20-26. Kick0 Register (KICK0R)**

| 31 | 0 |
|---|---|
| KICK0 | |
| W-0 | |

LEGEND: W = Write only; -*n* = value after reset

**Table 20-25. Kick0 Register (KICK0R) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | KICK0 | 0 | Kick0 data |

#### Kick1 Register (KICK1R)

The Kick1 register allows writing to unlock the kick1 data and the kicker mechanism to write to other MMRs. To disable RTC register write protection, the value of 83E7 0B13h must be written to KICK0R, followed by the value of 95A4 F1E0h written to KICK1R.

**Figure 20-27. Kick1 Register (KICK1R)**

| 31 | 0 |
|---|---|
| KICK1 | |
| W-0 | |

LEGEND: W = Write only; -*n* = value after reset

**Table 20-26. Kick1 Register (KICK1R) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | KICK1 | 0 | Kick1 data |


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.22 RTC Revision Register (RTC_REVISION)

**Figure 20-28. RTC Revision Register (RTC_REVISION)**

| 31–30 | 29–28 | 27–16 |
|---|---|---|
| SCHEME | Reserved | FUNC |
| R-1 | R-0 | R-EB0h |

| 15–11 | 10–8 | 7–6 | 5–0 |
|---|---|---|---|
| R_RTL | X_MAJOR | CUSTOM | Y_MINOR |
| R-1 | R-1 | R-0 | R-4h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-27. RTC Revision Register (RTC_REVISION) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | SCHEME | 0-3h | Used to distinguish between old scheme and current |
| 29-28 | Reserved | 0 | Reserved |
| 27-16 | FUNC | 0-FFFh | Function indicates a software compatible module family |
| 15-11 | R_RTL | 0-1Fh | RTL Version (R) |
| 10-8 | X_MAJOR | 0-7h | Major Revision |
| 7-6 | CUSTOM | 0-3h | Indicates a special version for a particular device |
| 5-0 | Y_MINOR | 0-3Fh | Minor Revision (Y) |

### 20.3.23 System Configuration Register (RTC_SYSCONFIG)

**Figure 20-29. System Configuration Register (RTC_SYSCONFIG)**

| 31–16 |
|---|
| Reserved |
| R-0 |

| 15–2 | 1–0 |
|---|---|
| Reserved | IDLEMODE |
| R-0 | R/W-2h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-28. System Configuration Register (RTC_SYSCONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1-0 | IDLEMODE | 0-3h | Configuration of the local target state management mode, By definition target can handle read/write transaction as long as it is out of IDLE state. |
| | | 0 | Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e., regardless of the IP module's internal requirements; Backup mode, for debug only. |
| | | 1h | No-idle mode: local target never enters idle state, Backup mode, for debug only. |
| | | 2h | Smart-idle mode: local target's state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements, IP module shall not generate (IRQ- or DMA-request-related) wakeup events. |
| | | 3h | Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges the system's idle requests, depending on the IP module's internal requirements, IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state, Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 20.3.24 Wakeup Enable Register (RTC_IRQWAKEEN)

**Figure 20-30. Wakeup Enable Register (RTC_IRQWAKEEN)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | ALARM_WAKEEN | TIMER_WAKEEN |
| R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 20-29. Wakeup Enable Register (RTC_IRQWAKEEN) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1 | ALARM_WAKEEN | | Wakeup generation for event Alarm. |
| | | 0 | Wakeup disabled |
| | | 1 | Wakeup enabled |
| 0 | TIMER_WAKEEN | | Wakeup generation for event Timer. |
| | | 0 | Wakeup disabled |
| | | 1 | Wakeup enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

# Serial ATA (SATA) Controller

This chapter describes the Serial ATA Controller (SATA) in the device.

**Topic** **Page**

**21.1 Introduction** ..... **3395**
**21.2 Architecture** ..... **3399**
**21.3 Use Cases** ..... **3405**
**21.4 Registers (Controller and PHY)** ..... **3424**
**21.5 SATA PLL Registers** ..... **3475**
**21.6 SATA Subsystem Registers** ..... **3484**


Copyright © 2011–2016, Texas Instruments Incorporated

## 21.1 Introduction

The serial ATA is the successor of the parallel ATA/ATAPI controller that has served as the choice of the communication medium between a portable computer (PC) and a hard-disk drive for several decades. During these times, the PATA interface has gone through changes to sustain the demands of the newly emerging applications needs and eventually, it reached its technical limitation, creating a throughput bound. The PATA controller reached a point where it required major changes to satisfy the upcoming applications requirements and this led to its successor, the birth of the SATA controller. The SATA controller was designed to use the same logical command structures that PATA uses but with a whole new physical characteristic. The SATA controller makes use of two pairs of high-speed differential conductors as opposed to the parallel 16-bit, low-speed interface. This device has a built-in SATA controller with a single HBA port operating in AHCI mode, and is used to interface to data storage devices at both 1.5 Gbits/Sec and 3.0 Gbits/Sec line speeds per HBA port. AHCI describes a system memory structure which contains a generic area for control and status, and a table of entries describing a command list where each command list entry contains information necessary to program a SATA device, and a pointer to a descriptor table for transferring data between system memory and the device.

### 21.1.1 Purpose of the Peripheral

The SATA controller addresses the drawback of the PATA interface architecture, throughput, and protocol perspectives. PATA required 40/80 wire parallel cable with a length requirement not exceeding 18 inches. With the latest/final PATA H/W design, PATA's maximum transfer rate saturated to a 133 Mbytes/Sec of transfer. In addition to newly added features, like hot swapping and native command queuing, the SATA controller abandoned the parallel physical interface and transitioned to a high-speed serial format using two differential pairs supporting up to 3 Mbits/Sec transfer rate, translating to a 300 Mbytes/Sec raw throughput for the single HBA port.

With the intent on handling roles of the PATA controller and addressing the future needs, the SATA controller is architected with the option of operating in a Legacy mode (which behaves similar to the PATA controller from the protocol/driver perspective), and a new AHCI mode that is different from the Legacy mode. The AHCI mode has a new interface whose standard is not compatible with PATA and does not support the Legacy mode of operation. The communication between a device and software moves from the PATA task file registers access made via byte-wide accesses to a structure based, command Frame Information Structure (FIS), located in system memory that is fetched by the HBA. This reduces command setup time significantly, allowing for multiplexing more SATA devices to a single HBA port. A port multiplier (PM) is required for this task and allows as many as 15 devices to attach to a single HBA port. Software no longer communicates directly to a device via task file registers. In other words, all data transfers between the device and system memory occur through the HBA acting as a bus master to system memory and this transfer happens without intervention of user software. Whether the transaction is of a DMA type or a PIO type (the use of the PIO command type is strongly discouraged and all transfers should be performed using DMA unless a transaction is only performed via PIO command, which in this case is still done via the AHCI master port not the CPU), the HBA fetches and stores data to memory, offloading the CPU. No data port exists for moving data in and out of the system memory similar to the PATA offerings. Software written for AHCI is not allowed to utilize any of the legacy mechanisms to program devices.

The SATA controller uses a less massive thinner flexible cable that can be up to 3 feet (1 meter) in length allowing for easier routing and better air ventilation inside a case. Its power budget is significantly reduced to 250 mV compared to the required 5V and/or 3.3V power of PATA.

Like its predecessor, the SATA controller is most commonly used by PCs, portable devices, and embedded devices to interface a host processor with external data storage or CD/audio devices. It also has the support for hot swapping capability and the ability of interfacing to a port multiplier (PM) to increase the number of devices that can be attached to a single HBA port by as many as fifteen devices. Note that in the case where a PM is used, the bandwidth offered by a single HBA port will be shared amongst the total number of devices attached to the port multiplier.

### 21.1.2 Features Supported

The main features of the SATA controller are:

- Support for Synopsys DWH Serial ATA 1.5Gbps and 3Gbps speeds core


Copyright © 2011–2016, Texas Instruments Incorporated

- Supports AHCI Controller (specification version 1.1)
- Supports one HBA port
- Integrated TI SERDES PHY (Mercury)
- Built in/integrated Rx and Tx data buffers
- Supports all SATA power management features
- Internal DMA engine per port
- Hardware-assisted native command queuing (NCQ) for up to 32 entries
- 32-bit addressing
- Supports port multiplier with command-based switching
- Activity LED support

### 21.1.3 Features Not Supported

Features not supported in this SATA controller are:

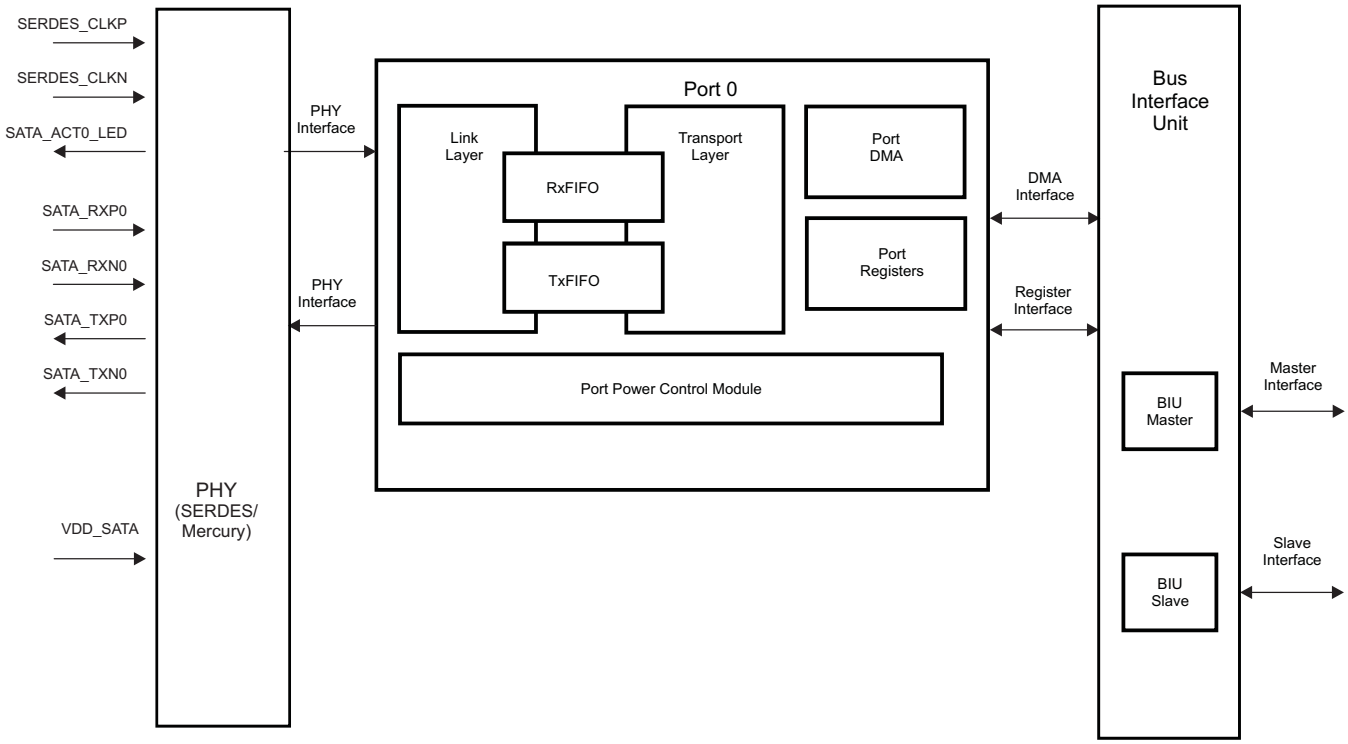
- Legacy mode of operation
- Master/slave type of configuration
- Far-end analog loopback
- Message signaled interrupts
- 64-bit addressing
- Dedicated Mechanical Presence Switch Pin (GPIO usage is expected to supplement this task if necessary)
- Dedicated Cold Presence Detect Pin (GPIO usage is expected to supplement this task if necessary)


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.1.4 Functional Block Diagram

The SATASS is a fully contained Serial ATA host with built in DMA. It uses the AHCI standard for communication with a SATA device. It has no support for the Legacy mode of operation. Figure 21-1 shows a high level block diagram of the SATA Subsystem (core and the integrated TI PHY (SERDES)).

**Figure 21-1. SATA Core Block Diagram**




### 21.1.5 Industry Standard(s) Compliance

The SATA Subsystem complies with the following industry standards:

- SATA revision 2.6 Gold standard
- AHCI revision 1.1 specification

### 21.1.6 Non-Industry Standard(s) Compliance

The SATA Subsytem complies with the following proprietary standards:

- Synopsys Design Ware Core DWC AHCI Controller Version 2.10
- TI SERDES PHY: Mercury GS70 Revision 1.2.


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.1.7 Terminology Used in this Document

The following is a brief explanation of some terms used in this document:

**AHCI —** Advanced Host Controller Interface (necessary for implementing newly supported features like native command queuing). The device supports only this mode of operation, i.e., has no support for Legacy mode. It also reduces CPU/Software overhead when moving data in and out of the system memory. Note that system software is responsible to ensure that queued and non-queued commands are not mixed in the command list.

**ATA/ATAPI—** Advanced Technology (AT) attachment/ATA packet interface

**CF—** Compact Flash

**Device—** External SATA or ATAPI device attached to the SATA controller

**DMA—** DMA within the ATA controller, not the processor EDMA system

**DWORD—** DWORD is 32 bits of data.

**Command List—** Command List is a memory buffer for as much as 32 commands. Each command is entered in a command slot. This is a required feature when supporting command queuing.

**Command Slot—** A command slot is a subset of the command list. It is the memory buffer for a single command. A total of 32 command slots exist.

**D2H—** D2H is an acronym for “device to HBA” and is mostly used to indicate the direction of the transmission of FIS which is from device to the HBA (host).

**FIS—** FIS is an acronym for “Frame Information Structure.” A frame is made of a data payload with computed CRC and a start and end primitives.

**H2D—** H2D is an acronym for “HBA to device” and is mostly used to indicate the direction of the transmission of FIS which is from HBA (host) to device.

**HBA —** HBA is an acronym for “host bus adapter.” It is the SATA controller that implements the AHCI specification to communicate between system memory and Serial ATA devices.

**Legacy Mode—** This mode of operation allows the user's application software or driver to view the SATA controller in a similar fashion as a PATA controller. However, the device does not support the Legacy mode of operation; it supports the AHCI mode of operation.

**OOB —** Out of Band. OOB signaling is used for during device detection and when recovering from power states.

**PM —** Port Multiplier. A port multiplier allows extending an HBA port connection capability to connect to multiple SATA Devices, a maximum of 15 Devices. Note that the operating bandwidth is shared amongst all the Devices when using this type of configuration.

**PMP —** Port Multiplier Port : A Port of a Port Multiplier (PM).

**PORT—** Refers to HBA Port, mainly within this document. However, the PORT is also be used to refer to a port of a PORT Multiplier.

**PRD —** PRD is an acronym for “physical region descriptor.” A PRD table is a data structure used by DMA engines that comply with the ATA/ATAPI Host Adapters standard. The PRD describes memory regions to be used as the source or destination of data during DMA transfers. A PRD table is often referred to as a scatter/gather list.

**SATA Controller—** Serial ATA controller, also HBA.

**System Memory—** Memory that is external from the SATA controller but is accessible by the built-in SATA controller DMA or the CPU.


Copyright © 2011–2016, Texas Instruments Incorporated

## 21.2 Architecture

The Serial ATA Controller supports an advanced host controller interface (AHCI) which is a standard interface. The single host port SATA controller does not support the Legacy mode of operation. Since the controller complies with the AHCI standard (version 1.1) the details of its operation are captured within the AHCI version 1.1 specification. Please consult the specification for the general behavior of the AHCI SATA Core operation. The SATA controller supported within the device has one HBA port.

### *21.2.1 Clock Control*

The SATA controller makes use of one internal 200 MHz clock (PRCM SYSCLK4 output clock) for both functional and keep alive functions. During a low powerdown event, the keep-alive clock is necessary to insure that portions of the controller interface remain up and running at all times to keep the controller context in tact, as well as to clock the hardware logic necessary used in detecting wakeup events from power-down modes. Access to the SATA controller and resources are implemented through SYSCLK4 derived from the output of the main input reference clock source. SYCLK4 to the SATA controller is gated by the PRCM (power, reset, and clock management) at the controller boundary and is required to be enabled prior to accessing the SATA controller. However, the keep-alive clock is always ON, i.e., it can not be gated at the controller boundary, and can not be turned off as long as the device is in normal operation state.

The SATA PHY requires multiple clock sources for operation. The main device reference 20 MHz clock is one of them and is required at all times. The other clock can be derived from the derivative of the 20MHz reference clock using the PHY PLL or it can be sourced from an external differential clock source. If sourced externally, it requires a high-quality, low-jitter external clock. A 100 MHz clock is preferred in such a case, since this external clock is shared with the PCIe SS that has this requirement..

The selection for the input clock source for the SERDES, i.e., the 20 MHz reference cock or the 100 MHz differential clock, is configured via one of the SATA PLL Configuration register fields (CFGPLL0. SEL_IN_FREQ bit field located @ 0x4814_0720) where 0/1 implies external SERDES differential input/ reference 20 MHz clock input source. Note that the SATA PLL configuration registers are accessed from the device configuration register space and these registers lie within the control module space which is found in a different location from where the the SATA peripheral and PHY registers reside.


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.2.2 Signal Description

The device has bonded out the data lines (a set of differential data lines) and a device activity indicator. Signals needed to detect device detection and attached device power indicators need to be implemented via GPIO if desired and no dedicated signals for these tasks exist. The power pins and logic necessary to power-up a cold SATA device should be handled external to the device.

Table 21-1 summarizes the available SATA module signals.

**Table 21-1. SATA Interface Signal Descriptions**

| Terminal Name | Direction from the HBA Perspective (In/Out) | Description |
|---|---|---|
| SATA_RXP0 | Input | Receive Data Positive Differential Signal for Port 0 |
| SATA_RXN0 | Input | Receive Data Negative Differential Signal for Port 0 |
| SATA_TXP0 | Input | Transmit Data Positive Differential Signal for Port 0 |
| SATA_TXN0 | Input | Transmit Data Positive Differential Signal for Port 0 |
| SERDES_CLKP | Input | PHY Reference Positive Differential Signal |
| SERDES_CLKN | Input | PHY Reference Negative Differential Signal |
| SATA_ACT0_LED | Output(1) | Device Activity Indicator for HBA Port 0 |
| VDD_SATA (2) | | |

(1) Multiplexed with other peripherals.

### 21.2.3 DMA

The AHCI HBA port contain two DMA engines. One of the DMA engine is used to fetch command from the command list. The other DMA is used to move FISes in and out of system memory. These DMAs are part of the AHCI controller. All data movement in and out of the device is performed through the DMA and no CPU-driven data movement option is available.

The DMA used to move FISs in and out of system memory has a register, P0DMACR, to control the burst transfer. This DMA is used to transfer all information between system memory and the attached SATA device, as well as configuration and status FISs.

The user can program the maximum burst size that will be issued on the system bus independently for both reads and writes. The DMA will issue transactions equaling the programmed size or smaller (in DWORD increments). This can be used to optimize burst size for over-all system throughput efficiency. Refer to Section 21.4.32 for details on legal values. Note that programming a burst size of greater than a transaction size (below), while is not invalid, is meaningless because the DMA maximizes out at the maximum transaction size.

The user can also program the transaction size for both receive and transmit (see Section 21.4.32). The transaction size is the minimum amount of data that the DMA will work on. For example, if there is an FIS coming from the device to the host, the DMA will not begin transferring data into system memory until there is at least RX_TRANSACTION_SIZE (RXTS) data in the receive FIFO. During transmit, the DMA will read data from system memory in TX_TRANSACTION_SIZE (TXTS) increments to put into the transmit FIFO. Note that transactions may be broken up into multiple bursts based on burst size, crossing of a 1k boundary, or end-of-frame.

### 21.2.4 Transport Layer

The transport layer handles all of the transport layer functions of the SATA protocol. During reception, it receives a FIS from the Link layer via the Rx FIFO, decodes the type, and routes it to the proper location via the port DMA. During transmission, it transfers a FIS constructed by the port DMA to the link layer via the Tx FIFO. It also passes link layer errors and checks for transport layer errors to pass up to the system.


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.2.5 FIFOs

The transport layer also contains the Tx and Rx FIFOs. These FIFOs are used as asynchronous data buffers between the serial domain and the bus clock domain. The size of these FIFOs affects the subsystems ability to buffer data before flow control must be asserted. It also affects the maximum programmable transaction and burst sizes that can be programmed into the port DMA. The Tx FIFO size is 64 DWORDS (256 bytes) deep while the Rx FIFO size is 128 DWORDS (512 bytes) deep.

### 21.2.6 Link Layer

The link layer maintains the link and supports all SATA link layer functionality including:

- Out-of-band (OOB) transmit signaling
- Frame negotiation and arbitration
- Envelope framing/de-framing
- CRC calculation (receive and transmit)
- 8b/10b encoding/decoding
- Flow control
- Frame acknowledgment and status
- Data width conversion
- Data scrambling/descrambling
- Primitive transmission
- Primitive detection and dropping
- Power management

### 21.2.7 PHY

The SATASS includes an integrated TI SERDES macro as a PHY. The PHY handles all of the serialization/de-serialization, symbol alignment, and Rx OOB signal detection.

### 21.2.8 Pin Multiplexing

Extensive pin multiplexing is used to accommodate the largest number of peripheral functions in the smallest possible package; therefore, some of the SATA controller peripheral signals share pins with other peripherals; SERDES_CLKP/N is shared with PCI Express and SATA_AC0_LED is shared with multiple peripherals. When using the SATA_AC0_LED bit for SATA function, the corresponding PINMUX register is required to be configured for SATA usage. Since the same SERDES is used by SATA and PCI Express peripherals, it is recommended the use of a 100MHz differential input clock source that is ideal for both peripherals. Refer to the device-specific data manual to determine how pin multiplexing affects the SATA.

### 21.2.9 Power Management

The SATA controller can be placed in reduced power modes to conserve power during periods of no activity or no use. The main power management of the peripheral is controlled by the Power, Reset, and Clock Management (PRCM) unit. The PRCM acts as a master controller for power management of all of the peripherals on the processor. For detailed information on power management procedures using the PRCM, see the *System Reference Guide*.

During times that the SATA peripheral is in use, the SATASS supports the industry standard power down modes (both Partial and Slumber low power modes) as provided within the SATA specification. These modes allow for power savings by powering down part of the SERDES PHY and by providing the ability to gate off the clocks to the link layer. The Port Power Control Module is used to enter and exit this power down modes (that is power down mode is controlled at Port level) which may have the normal functional clocks gated off.


Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** When SATA communication is in the logical idle state, that is, when no disk activity takes place, the communication remains active with both the host and the device sending a logical sync primitive and scrambled data at the speed negotiated continuously. For power sensitive applications, the power consumed during the disk inactivity stage might be undesirable and it might be a desired task to place the communication interface into an electrical idle (Partial or Slumber) state until data transfer activity is needed in order to conserve power.

### 21.2.10 Reset

The SATA controller reset is handled via PRCM. Please consult the PRCM document for details on how to use the PRCM module. Other types of resets (HBA Reset, Port Reset, and Software Reset) supported by the SATA controller are part of the AHCI specification. See the AHCI Standard Specification 1.1 for details on HBA Reset, Port Rest, and Software Reset.

### 21.2.11 Interfacing to Single and Multiple Devices

The SATA controller supports one HBA port. This means that the HBA port can be used to interface directly to a single SATA device or to multiple SATA Devices via a port multiplier. Note that on a multiple target setup, the available bandwidth will be shared amongst all the attached devices.

Note that Port Multipliers and Power Supplies for external SATA devices, when needed, should be furnished external to the device. No dedicated signals for controlling power and device detection are not bonded out and usage of GPIO pins can be used to supplement this task. However, the controller supports the capability to spin-up-attached devices independently.

#### 21.2.11.1 Interfacing to a Single Device

If need to interface directly to a single device, there is no need of populating a port multiplier. The software needs to ensure that the PMP field within the Command Header and the PM_PORT field of the FIS be cleared at all times.

If the external SATA device is not Self Powered, it is the responsibility of the System Designer to populate and furnish the right power supply needed. Consult the SATA Specification for details.

#### 21.2.11.2 Interfacing to Multiple Devices

Interfacing to multiple devices is pretty much identical to interfacing with a single device except the addition of a hardware Port Multiplier and initialization on software side. User software is required to perform the additional task of detecting and configuring the PM prior to accessing the attached devices. The user software is required to populate the Port Multiplier Port (PMP) field of the FIS in order to select one of the fifteen devices that could be connected and accessed by the HBA. The details on this are fully captured within the AHCI specification Version 1.1.

### 21.2.12 Initialization

Proper initialization of the HBA is required after power up to ensure proper operation of the SATA controller peripheral. The initialization process starts by performing a write to one-time write only registers (this initialization step is documented as Firmware initialization within the AHCI specification) where the values programmed depend on the features that the applications support. Note that the features that are enabled by the Firmware initialization are subsets of the features supported by the SATA subsystem. This Firmware initialization is similar to what a PC BIOS does and allows the user to enable/disable some supported features by using software.

The software can then continue with the normal initialization that is required by the software. The details and sequence of initialization is documented within the AHCI specification. In general, everything that is achieved by the Software Initialization has to do with configuring and furnishing resources needed by the AHCI controller and these tasks include PHY Initialization, allocating structures and memories for Command Slots, FIS, and Data Memories and concludes by enabling the receive FIS DMA. The software will then spin-up the device and ensure that a proper Device Detection and Speed Negotiation has completed prior to enabling the Command DMA.


Copyright © 2011–2016, Texas Instruments Incorporated

Note that DMA Configuration/Initialization should take place after Device Detection and Speed Negotiation. If done earlier, the default value is used (which is the recommended setting) since RESET removes the user programmed values. The only time it is advisable to change the DMA Configuration is when you need to prioritize System Resource access. This still has to be done after "PHY Ready" status is set (in other words, after the device detection and speed negotiation has completed). It also requires that the Command DMA is not running (P0CMD.ST = 0) when modifying the value of the DMA Configuration fields.

#### 21.2.12.1 Initialization (Firmware and Software)

Software reads the HBA capabilities register (CAP), ports implemented register (PI), AHCI version register (VS), global parameter 1 register (GPARAM1R), global parameter 2 register (GPARAM2R), and the port parameter register (PPARAMR) to obtain information about the subsystem's capabilities. The software should then take the following steps to configure each port for operation:

1. Do all firmware capability writes.
2. Setup all appropriate structures in memory as per the AHCI specification.
3. Set the port command list base address register (P0CLB).
4. Set the port FIS base address register (P0FB).
5. Set appropriate bits in the port command register (P0CMD).
6. Program the port serial ATA control register (P0SCTL).
7. Wait for Device Detection and Speed Negotiation to end.
8. Program the port DMA control register (P0DMACR).
9. Enable the appropriate interrupts.
10. Enable FIS reception in P0CMD.
11. Spin-up the device(s), if necessary.

#### 21.2.12.2 Issuing a Command

Once the host and device are configured, perform the following steps to issue a command:

1. Create the appropriate FIS in system memory.
2. Create the PRD.
3. Queue the command to the command queue list (location specified by the port command list base address register (P0CLB).

For detailed information, see the AHCI Specification Version 1.1.


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.2.13 Interrupt Support

The AHCI controller supports both standard interrupt sourcing, where interrupts are generated when enabled events occur, or a different type of method of generating interrupts that minimize interrupt loading by either generating interrupts in a batch or periodically. The latter method is handled using Command Completion Coalescing method and the details is captured within the AHCI Specification Version 1.1.

#### 21.2.13.1 Command Completion Coalescing

Command Completion Coalescing (CCC) is a feature designed to reduce the interrupt and command completion overhead in a heavily loaded system. The feature enables the number of interrupts taken per completion to be reduced significantly, while ensuring a minimum quality of service for command completions. When software specified number of commands have completed or a software specified timeout has expired, an interrupt is generated by hardware to allow software to process completed commands. The command completion coalescing ports register (CCC_PORTS) should be programmed by setting the corresponding bit of the Port that is to be included in command completion coalescing feature. Note that the device supports 1 HBA port.

For a detailed explanation of the CCC initialization and usage, see the AHCI Specification 1.1 Section 11.6.

##### 21.2.13.1.1 CCC Interrupt Based on Timer Expiration

When CCC is enabled and the desired method to receive an interrupt is based on a timer elapse condition, then the user needs to communicate a resolution for a 1ms time by programming the TIMER1MS register with the OCP cycle count derived from the OCP clock frequency sourced to the SATA controller.

As an example, if a user desires for interrupt to be generated every 15ms, for OCP bus clock frequency of 200 MHz, the 1ms cycle count should be programmed with a value of 250,000, i.e., 250 MHz/1000 = 200000d, and CCC_CTL.TV is programmed with a non-Zero millisecond value (15 in this case). When CCC_CTL.EN is set to 1 (CCC is enabled), the CCC will periodically generate an interrupt every 15 ms or every 15 * 200000 = 3,000,000 OCP cycles.

**NOTE:** Make sure the EN bit in the command completion coalescing control register (CCC_CTL) is cleared to 0 (i.e., CCC is disabled, prior to programming this field).

##### 21.2.13.1.2 CCC Interrupt Based on Completion Count

When CCC is enabled and the desired method to receive an interrupt is based on a completion count, that is, the CC bit in the command completion coalescing control register (CCC_CTL) is programmed with a non-zero value and the CCC interrupt is enabled (EN bit in CCC_CTL is set to 1), an interrupt is sourced from the SATA controller when the programmed desired number of interrupt is received.

**NOTE:** Make sure the EN bit in the command completion coalescing control register (CCC_CTL) is cleared to 0 (i.e., CCC is disabled, prior to programming this field).

#### 21.2.13.2 Non CCC Interrupt Configuration

For a standard interrupt handling method where every event that is enabled generates an interrupt, is handled as follows. For more information, see the AHCI Specification.

After insuring that CCC is disabled, the EN bit in the command completion coalescing control register (CCC_CTL) is 0, in order for the SATA Core to source interrupts, the interrupt should be enabled at both global level (the IE bit in the global HBA control register (GHC) is set to 1) and port level by enabling the bit fields for the desired interrupt. An enable bit at a Port level controls corresponding interrupt dispatch to the processor interrupt handling resource. So long as the CPU interrupt handler is configured properly, the CPU receives the interrupt when the enabled event occurs.


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.2.14 EDMA Event Support

The SATA controller makes use of its own built-in DMA and has no need for the use of the processor EDMA.

## 21.3 Use Cases

The following sections include some sample program snippets that can be used as a guide for software development. The example demonstrates one of the ways of creating the necessary structures; properly aligned system memory resources, initialization, as well as performing basic DMA Read/Write transfer using a couple of the Command Slots.

Section 21.3.2 contains examples in relations to System Memory resource allocations, Structures, and Subroutines used by the Initialization and Read/Write Transfer functions. The remaining sections include examples of basic Initialization, DMA Write transfer, and DMA Read transfer examples using Port 0.

### 21.3.1 Controller Clock, PLL and SERDES Initialization

The SATA PLL and SERDES configuration requires a predefined sequence of register access and initialization. The following initialization sequence is recommended to be followed prior to starting SATA controller initialization.

```
*SATA_PLLCFG0 = 0x00000004;
wait_for_cpu_cycles(2000);
*SATA_PLLCFG1 = 0xC12C003C;
wait_for_cpu_cycles(2000);
*SATA_PLLCFG3 = 0x004008E0;
wait_for_cpu_cycles(850);
*SATA_PLLCFG0 |= 0x00000010;
 wait_for_cpu_cycles(85);
*SATA_PLLCFG0 |= 0x00000002;
 wait_for_cpu_cycles(60);
wait_for_cpu_cycles(2000);

* CM_ALWON2_SATA_CLKCTRL |= 0x2;
 while ((*CM_ALWON2_SATA_CLKCTRL & 0x0F) != 0x2);

* CM_DEFAULT_L3_MED_CLKSTCTRL |= 0x2; // Enable SATA Clock
while ((*CM_DEFAULT_L3_MED_CLKSTCTRL & 0x0F) != 0x102);



*CFGTX0_OFFSET = 0x01003622;
*CFGTX1_OFFSET = 0x40000002;
*CFGTX2_OFFSET = 0x00C201F8;
*CFGTX3_OFFSET = 0x073CE39E;
*CFGRX0_OFFSET = 0x008FCC22;
*CFGRX1_OFFSET = 0x008E0500;
*CFGRX2_OFFSET = 0x7BDEF000;
*CFGRX3_OFFSET = 0xC0007077;

*SATA_PLLCFG0 |= 0xC0007061;

while (*SATA_PLLSTATUS & 0x1) ! = 0x1); // wait until PLL lock
```


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.3.2 General Utilities: Structures and Subroutines Sample Program Uses

```
/* Allocating memory for Command List. The structure pointed to by this
   address range is 1K-bytes in length and must be 1K-byte aligned.
   Note that each command header occupies 32 bytes of memory and 32
   command headers require 1024 bytes of memory.
*/
#pragma DATA_SECTION(CmdLists,     OCMCRAM0);
#pragma DATA_ALIGN(CmdLists, 1024);
CmdListHeader CmdLists[32]={0};

/* Indicates the 32-bit physical address of the command table, which
   contains
       the command FIS,
       ATAPI Command,
       andPRD table.
   This address must be aligned to a 128-bytes of memory,
*/
#pragma DATA_SECTION(CmdTable,     OCMCRAM0);
#pragma DATA_ALIGN(CmdTable, 128);
CommandTable CmdTable[LISTLENGTH];

/* Indicates the 32-bit base physical address for received FISes. The
   structure pointed to by this address range is 256 bytes in length
   and must be 256-byte aligned.
*/
#pragma DATA_SECTION(RcvFis,     OCMCRAM0);
#pragma DATA_ALIGN(RcvFis, 256);
ReceiveFis RcvFis;

#pragma DATA_SECTION(prdTableDataBuff,     OCMCRAM0);
unsigned char prdTableDataBuff[LISTLENGTH][PRDLENGTH][DATABUFFERLEN];


#define NUMOFPORTS       (2)    // Supports Two HBA Ports. However it can support up to
                //  15 additional Ports, per HBA Port, for a combined total of 30 Port Multiplier
                //  Ports (PMP) capable of attaching to 30 devices.
#define LISTLENGTH       (2)    // Max Command Header Per Port is 32

#define WRITE_CMD_SLOT   (0)    // Value used here should be <= LISTLENGTH-1
#define READ_CMD_SLOT    (1)    // Value used here should be <= LISTLENGTH-1

    // WARNING. PRDLENGTH can not be greater than 8 for this program.
    //  See Note captured by the area when memory has been reserved for
    //  within sata_utilities.c for Command Table "CmdTable" for
    //  more information.
#if 1
    #define _MAX_DATA_TRANSFER_    // Define this in project file when needed.
#endif

#ifndef _MAX_DATA_TRANSFER_           // 512 Bytes Data Size within 2 PRD Descriptors.
    #define PRDLENGTH     (2)       // Max PRD Length is 65535 per port.
    #define DATABUFFERLEN (256)     // DMA Data Buffer Length
#else // Max Data Size Transfer 8K Bytes within 2 PRD Descriptors
    #define PRDLENGTH     (2)       // Max PRD Length is 65535 per port.
    #define DATABUFFERLEN (2*4096)     // DMA Data Buffer Length
#endif

#if ((PRDLENGTH > 8) | (WRITE_CMD_SLOT > LISTLENGTH-1) | (READ_CMD_SLT > LISTLENGTH-1))
#error PRDLENGTH ENTRY ERROR - PROGRAM HARD CODED FOR MAX VALUE OF 8 - CMD SLOT ENTRY ERROR
#endif


#define DESIRED_SPEED           (GEN1)          // GOASFASTASDEVICE, GEN1, GEN2
#define DEVICE_LBA_ADDRESS      (0x00000002) // Dev28bitLbaAddress = 28-Bit LBA Address
#define WAIT_500_MILLISECONDS   (50)     // This should be set to 500 once the ONE_MS_VALUE is
programmed correctly.
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
#define WAIT_1_MILLISECOND      (1)
#define ONE_MS_VALUE        (1)        // Number of CPU Cycles needed to generate a millisecond
wait time.
#define DMA_BURST_LENGTH        (0x9)           // [0x0 - 0x9] Burst=2^(–1) i.e., 0x8=> 2^(9-
1)=256
#define DMA_TRANSACTION_SIZE    (0x4)         // [0x0 - 0xA] TransSize=2^n i.e., 0xA=> 2^10=1024

////////////////////////////////////////////////////////////////////////////////////
// Maximum of 32 commands slots per port exist where each command occupies 8 DWs (64 Bytes).
// The structure 'CmdListHeader' defines a single command header definition.
// The start of the first Command List &CmdListHeader[0] needs to be programmed onto P0CLB.
//
// Command List Base Address should be 1K Byte Aligned.

typedef struct {
    Uint32 CmdLen:5;   //bits[4:0]
    Uint32 Atapi:1;    //bit[5]
    Uint32 Write:1;    //bit[6]
    Uint32 Prefetch:1; //bit[7]
    Uint32 Reset:1;    //bit[8]
    Uint32 Bist:1;     //bit[9]
    Uint32 Rok:1;      //bit[10]
    Uint32 Rsv:1;      //bit[11]
    Uint32 Pmp:4;      //bits[15:12]
    Uint32 Prdtl:16;   //bits[31:16]
}CmdListHeaderW0;


typedef struct {
    Uint32 PrdByteCnt; //bits[31:0]
}CmdListHeaderW1;


typedef struct {
//    Uint32 CmdTableAddLowRsv:7; //bit[6:0]
//    Uint32 CmdTableAddLow:25;   //bits[31:7]
    Uint32 CmdTableAddLow;   //bits[31:7]
}CmdListHeaderW2;

typedef struct {
    Uint32 CmdTableAddHigh;     //bits[31:0]
}CmdListHeaderW3;


typedef struct {
    CmdListHeaderW0 DW0;
    CmdListHeaderW1 DW1;
    CmdListHeaderW2 DW2;
    CmdListHeaderW3 DW3;
    Uint32          DW4;
    Uint32          DW5;
    Uint32          DW6;
    Uint32          DW7;
} CmdListHeader;



typedef struct {
    Uint32 B0FisType:8;//bits[7:0]
    Uint32 BYTE1:8;       //bits[15:8]
    Uint32 B2Cmd:8;    //bits[23:16]
    Uint32 B3Feature:8;//bits[31:24]
}CmdFisWord0;

typedef struct {
    Uint32 B0LbaLow:8; //bits[7:0]
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
    Uint32 B1LbaMid:8; //bits[15:8]
    Uint32 B2LbaHigh:8;//bits[23:16]
    Uint32 B3Device:8; //bits[31:24]
}CmdFisWord1;

typedef struct {
    Uint32 B0LbaLowExp:8; //bits[7:0]
    Uint32 B1LbaMidExp:8; //bits[15:8]
    Uint32 B2LbaHighExp:8;//bits[23:16]
    Uint32 B3FeatureExp:8; //bits[31:24]
}CmdFisWord2;

typedef struct {
    Uint32 B0SecCnt:8;    //bits[7:0]
    Uint32 B1SecCntExp:8; //bits[15:8]
    Uint32 B2Rsv:8;       //bits[23:16]
    Uint32 B3Control:8;   //bits[31:24]
}CmdFisWord3;


typedef struct {
    Uint32 DWResv; //bits[31:0]
}CmdFisWord4;

typedef struct {
    CmdFisWord0 DW0;
    CmdFisWord1 DW1;
    CmdFisWord2 DW2;
    CmdFisWord3 DW3;
    CmdFisWord4 DW4;
    Uint32      DW5;
    Uint32      DW6;
    Uint32      DW7;
    Uint32      DW8;
    Uint32      DW9;
    Uint32      DW10;
    Uint32      DW11;
    Uint32      DW12;
    Uint32      DW13;
    Uint32      DW14;
    Uint32      DW15;
}CommandFIS;

//-----------Command FIS end ATAPI Command ------

// ATAPI Command Data Structure
typedef struct {
    Uint32 ATAPI[4];
}Atapi;

//-----------ATAPI Command end PRDT -------------
// Physical Region Descriptor Table Data Structure
typedef struct {
    Uint32 DbaLow;   //bits[31:0]
}DbaAddressLow;

typedef struct {
    Uint32 DbaHigh; //bits[31:0]
}DbaAddressHigh;

typedef struct {
    Uint32 DW2Reserved; //bits[31:0]
}PrdtRsv;

typedef struct {
    Uint32 DataBC:22;  //bits[21:0]
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
}DataByteCnt;


typedef struct {
    DbaAddressLow  DW0;
    DbaAddressHigh DW1;
    PrdtRsv        DW2;
    DataByteCnt    DW3;
}PRDT;
//-----------PRDT end ---------------------------

//-----------Command Table Data Structure -------
// Since Command Table has to be 128 bytes = 0x80 bytes aligned if supporting more
//  than a single Command Header, then need to make sure that the Array you are
//  creating for all associated Command Tables match the 128 bytes alignment.
//  In order to do so, the number of PRD Table length you are allocating should be
//  multiples of 8.
typedef struct {
    CommandFIS cfis;
    Atapi      atapi;
    Uint32     Rsv[12];
    PRDT       prdTable[16];    // Have forced this size to 8 in order to meet the minimum
                          //  required size for Command Table.
}CommandTable;

//-----------Command Table Data Structure end ---


////////////////////////////////////////////////////////////////////////////////
// Receive FIS requires the Receive FIS to be 256 byte aligned. P0FB should be programmed
//  with this restriction.
//
// RECEIVE FIS Data Structure
// Members: DMA Setup FIS (DSFIS)
//          PIO Setup FIS (PSFIS)
//          D2H Register FIS (RFIS)
//          Set Device Bits FIS (SDBFIS)
//          Unknown FIS (UFIS)

//-----------DMA Setup FIS-----------------------

typedef struct {
    Uint32 B0FisType:8;//bits[7:0]
    Uint32 BYTE1:8;       //bits[15:8]
    Uint32 B2Rsv:8;    //bits[23:16]
    Uint32 B3Rsv:8;    //bits[31:24]
}DsfisW0;



typedef struct {
    DsfisW0 DW0;
    Uint32 DW1DmaBuffLow;
    Uint32 DW2DmaBuffHigh;
    Uint32 DW3Rsv;
    Uint32 DW4DmaBuffOffset;
    Uint32 DW5DmaXfrCnt;
    Uint32 DW6Rsv;
}DMASetupFis;

//-----------DMA Setup FIS end PIO Setup FIS ----

typedef struct {
    Uint32 B0FisType:8;//bits[7:0]
    Uint32 BYTE1:8;       //bits[15:8]
    Uint32 B2Status:8;    //bits[23:16]
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
    Uint32 B3Errror:8;//bits[31:24]
}PioSetupDW0;

typedef struct {
    Uint32 B0LbaLow:8; //bits[7:0]
    Uint32 B1LbaMid:8; //bits[15:8]
    Uint32 B2LbaHigh:8;//bits[23:16]
    Uint32 B3Device:8; //bits[31:24]
}PioSetupDW1;

typedef struct {
    Uint32 B0LbaLowExp:8; //bits[7:0]
    Uint32 B1LbaMidExp:8; //bits[15:8]
    Uint32 B2LbaHighExp:8;//bits[23:16]
    Uint32 B3Rsv:8;       //bits[31:24]
}PioSetupDW2;


typedef struct {
    Uint32 B0SecCnt:8;    //bits[7:0]
    Uint32 B1SecCntExp:8; //bits[15:8]
    Uint32 B2Rsv:8;       //bits[23:16]
    Uint32 B3Estatus:8;   //bits[31:24]
}PioSetupDW3;

typedef struct {
    Uint32 HW0XferCnt:16; //bits[15:0]
    Uint32 HW1Rsv:16;     //bits[31:16]
}PioSetupDW4;


typedef struct {
    PioSetupDW0 DW0;
    PioSetupDW1 DW1;
    PioSetupDW2 DW2;
    PioSetupDW3 DW3;
    PioSetupDW4 DW4;
}PIOSetupFis;

//-----------PIO Setup FIS end D2H Reg FIS-------


typedef struct {
    Uint32 B0FisType:8;//bits[7:0]
    Uint32 BYTE1:8;       //bits[15:8]
    Uint32 B2Status:8;    //bits[23:16]
    Uint32 B3Errror:8;//bits[31:24]
}D2HRegDW0;

typedef struct {
    Uint32 B0LbaLow:8; //bits[7:0]
    Uint32 B1LbaMid:8; //bits[15:8]
    Uint32 B2LbaHigh:8;//bits[23:16]
    Uint32 B3Device:8; //bits[31:24]
}D2HRegDW1;

typedef struct {
    Uint32 B0LbaLowExp:8; //bits[7:0]
    Uint32 B1LbaMidExp:8; //bits[15:8]
    Uint32 B2LbaHighExp:8;//bits[23:16]
    Uint32 B3Rsv:8;       //bits[31:24]
}D2HRegDW2;


typedef struct {
    Uint32 B0SecCnt:8;    //bits[7:0]
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
    Uint32 B1SecCntExp:8; //bits[15:8]
    Uint32 HW1Rsv:16;     //bits[31:16]
}D2HRegDW3;

typedef struct {
    Uint32 W0Rsv;         //bits[31:0]
}D2HRegDW4;


typedef struct {
    D2HRegDW0 DW0;
    D2HRegDW1 DW1;
    D2HRegDW2 DW2;
    D2HRegDW3 DW3;
    D2HRegDW4 DW4;
}D2HRegFis;

//-----------D2H Reg FIS end Set Device Bits FIS-
// The Set Device Bit FIS definition does not contain the 2nd Word required
//  for Native Command Queueing. This second word is the SACTVE register and
//  the AHCI takes care of updating P0SACT register at its location.

typedef struct {
    Uint32 B0FisType:8;//bits[7:0]
    Uint32 BYTE1:8;       //bits[15:8]
    Uint32 B2Status:8; //bits[23:16]
    Uint32 B3Errror:8; //bits[31:24]
}SetDevBitsDW0;

typedef struct {
    Uint32 W1Rsv;     //bits[31:0]
}SetDevBitsDW1;

typedef struct {
    SetDevBitsDW0 DW0;
    SetDevBitsDW1 DW1;
}SetDevBitsFis;

//-----------Set Device Bits FIS end Unkonwn FIS-

typedef struct {
    Uint32 UserDefined; //bits[31:0]
}UnknownDWx;


typedef struct {
    UnknownDWx DW[16]; // 16 Words (Max 64 Bytes allowed)
}UnknownFis;

//-----------Unkonw FIS end----------------------

//-----------Receive Register FIS Structure------

typedef struct {
    DMASetupFis   DSFIS;
    Uint32        Rsv1;
    PIOSetupFis   PSFIS;
    Uint32        Rsv2[3];
    D2HRegFis     RFIS;
    Uint32        Rsv3;
    SetDevBitsFis SDBFIS;
    UnknownFis    UFIS;
}ReceiveFis;

/
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
typedef struct {
    Uint8 cfisType;
    Uint8 cfisByte1;
    Uint8 cfisCmd;
    Uint8 cfisFeature;
    Uint8 cfisDw1SecNumLbaLow;
    Uint8 cfisDw1CylLowLbaMid;
    Uint8 cfisDw1CylHighLbahigh;
    Uint8 cfisDw1Dev;
    Uint8 cfisDw2SecNumLbaLowExp;
    Uint8 cfisDw2CylLowLbaMidExp;
    Uint8 cfisDw2CylHighLbahighExp;
    Uint8 cfisDw2FeatureExp;
    Uint8 cfisDw3SecCnt;
    Uint8 cfisDw3SecCntExp;
    Uint8 cfisDw3Ctrl;
}cmdFis;

typedef struct {
    Uint8 dsfisType;
    Uint8 dsfisByte1;
    Uint32 dsfisDw1DmaBuffLow;
    Uint32 dsfisDw2DmaBuffHigh;
    Uint32 dsfisDw4DmaBuffOffset;
    Uint32 dsfisDw5DmaXferCnt;
}dsFis;

typedef struct {
    Uint8 psfisType;
    Uint8 psfisByte1;
    Uint8 psfisStatus;
    Uint8 psfisError;
    Uint8 psfisDw1SecNumLbaLow;
    Uint8 psfisDw1CylLowLbaMid;
    Uint8 psfisDw1CylHighLbahigh;
    Uint8 psfisDw1Dev;
    Uint8 psfisDw3SecCnt;
    Uint8 psfisDw3Estatus;
    Uint16 psfisDw4XferCnt;
}piosFis;

typedef struct {
    Uint8 regfisType;
    Uint8 regfisByte1;
    Uint8 regfisStatus;
    Uint8 regfisError;
    Uint8 regfisDw1SecNumLbaLow;
    Uint8 regfisDw1CylLowLbaMid;
    Uint8 regfisDw1CylHighLbahigh;
    Uint8 regfisDw1Dev;
    Uint8 regfisDw3SecCnt;
}regFis;

typedef struct {
    Uint8 sdbfisType;
    Uint8 sdbfisByte1;
    Uint8 sdbfisStatus;
    Uint8 sdbfisError;
}sdbFis;

typedef struct {
    Uint32 ufisWord[16];
}uFis;
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
typedef struct {
    Uint32  capSMPS:1;
    Uint32  capSSS:1;
    Uint32  piPi:2;
    Uint32  p0cmdCpd:1;
    Uint32  p0cmdEsp:1;
    Uint32  p0cmdMpsp:1;
    Uint32  p0cmdHpcp:1;
    Uint32  rsv:24;
}FirmwareCtrlFeatures;



void initMemory(Uint32 *startAddress, Uint32 length, Uint32 seedWord, Uint32 modifyVal) {
    Uint32 I;
    *startAddress++ = seedWord;
    for (I=0; i<length-1; I++) {
        seedWord += modifyVal;
        *startAddress++ = seedWord;
    }
}

void clearCmdList(void) {
    //clear Host to Device (Command FIS) Space
    initMemory((Uint32*)CmdLists, (LISTLENGTH*(sizeof(CmdListHeader)/4)), 0, 0);
}

void clearCmdTables(void) {
    Uint16 cmdSlot;
    for (cmdSlot=0; cmdSlot<LISTLENGTH; cmdSlot++) {
        //Clear Command FIS and ATAPI Command Spaces for Command Header X; LISTLENGTH < X < 0
        initMemory((Uint32 *)&CmdTable[cmdSlot], (sizeof(CommandFIS)/4)+(sizeof(Atapi)/4), 0, 0);
        //Clear PRD Descriptor Locations for Command Header X; LISTLENGTH < X < 0
        initMemory((Uint32*)((Uint32)&CmdTable[cmdSlot]+0x80),
(sata_input_filePageSize*(sizeof(PRDT)/4)*sata_input_prdLength), 0, 0);
    }
}

void clearRcvFis() {
    //clear Receive DMA Setup FIS Space.
    initMemory((Uint32*)&RcvFis, (sizeof(DMASetupFis)/4), 0, 0);
    //clear Receive PIO Setup FIS Space.
    initMemory((Uint32*)((Uint32)&RcvFis+0x20), (sizeof(PIOSetupFis)/4), 0, 0);
    //clear Receive Device to Host (D2H) Register FIS Space.
    initMemory((Uint32*)((Uint32)&RcvFis+0x40), (sizeof(D2HRegFis)/4), 0, 0);
    //clear Set Device Bits FIS Space.
    initMemory((Uint32*)((Uint32)&RcvFis+0x58), (sizeof(SetDevBitsFis)/4), 0, 0);
    //clear Unknow FIS Space.
    initMemory((Uint32*)((Uint32)&RcvFis+0x60), (sizeof(UnknownFis)/4), 0, 0);
}


void clearDmaBuffers(void) {
    //Clear PRD Data Buffer Memory
    initMemory((Uint32 *)prdTableDataBuff, (LISTLENGTH*PRDLENGTH*DATABUFFERLEN/4), 0, 0);
}


void performFirmwareInit(void) {
/* Firmware Initialization*/
//    Make sure you perform a Single Write in one operation of all HwInit Fields
//   initialization defined within a single register
    sataRegs->CAP |= ((swCtrlFeatures.capSMPS << 28) |
                      (swCtrlFeatures.capSSS  << 27)
                     );
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
// Configure PI[31:0]
    sataRegs->PI  |=  (swCtrlFeatures.piPi  << 0);

// Configure P0CMD[ESP,CPD,MPSP,HPCP=21,20,19,18]
    sataRegs->P0CMD |= ((swCtrlFeatures.p0cmdEsp << 21) |
                        (swCtrlFeatures.p0cmdCpd << 20) |
                        (swCtrlFeatures.p0cmdMpsp << 19)|
                        (swCtrlFeatures.p0cmdHpcp << 18)
                       );

/* Software Initialization*/

    initBaseAddresses(); // Initialize Command List (P0CLB) and Receive FIS (P0FS)

    // Configure Line Speed.
    setSataSpeed(DESIRED_SPEED);  //GOASFASTASDEVICE=0,GEN1=1,GEN2=2

    enableRcvFis();    // Enable Receive DMA


// The below are general Semaphores/Flags used and want to make sure they are initialized.
    // 28 Bit LBA Address of Device. 0xFFFFFFFF is used by test S/W to indicate that it is not
initialized
    Dev28bitLbaAddress = 0xFFFFFFFF; // S/W needs to initialize this variable prior to calling


    //_INT_DRIVEN_TEST_ is defined within the Project File
    #ifdef _INT_DRIVEN_TEST_
        intHandlingMethod = USE_INT_HANDLER;
    #else
        intHandlingMethod = USE_POLLING;
    #endif

}


char spinUpDeviceAndWaitForInitToComplete(void) {
    // Make sure that the HBA is in a Listen Mode prior to Spinning Up Device
    //  Following Configuration is not allowed.
    //      [P0SCTL.DET, P0CMD.SUD] = [1,1] NOT Allowed.
    if((sataRegs->P0SCTL & AHCI_PxSCTL_PxSSTS_DET) != 0)
        sataRegs->P0SCTL &= ~(0xf << AHCI_PxSCTL_PxSSTS_DET_SHIFT);

    // Clear P0SERR.DIAG.X (RWC bit field) so that the P0TFD is updated by HBA.
    sataRegs->P0SERR |= 0x04000000;

    // Spin Up Device.
    sataRegs->P0CMD |= (1 << AHCI_PxCMD_SUD_SHIFT);

    // Wait for Device Detection or/and Speed Negotiation to take place and finish.
    while ((sataRegs->P0SSTS & AHCI_PxSCTL_PxSSTS_DET) !=0x3);

    // Device would send its status and and default Task file regs content (signature)
    // when finished with Power Up: Look for Device ready status.
    while ((sataRegs->P0TFD & AHCI_PxTFD_STS_BSY_DRQ_ERR) != 0);

    // Make sure that the expected Device signature is received.
    if (sataRegs->P0SIG != AHCI_P0SIG_SIG_ATA_DEV_GOOD_STAT) // LBAhigh:LBAmid:LBAlow:SECcnt
        return(1);                                           // =0x00000101

    return(0);

}

void initIntAndClearFlags(void) {
    // Make sure Interrupt is disabled (Disable at Port Level followed by Global Level).
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
    // Clear Interrupt at Port Level
    enableDisableInt(PORTint, DISABLE, 0xFFFFFFFF); // clearInt(int type, intState, specificField)
                                            // int type=GLOBALint or PORTint
                                            // intState=DISABLE or ENABLE
                                            // specificField=bit field to Enable or Disable
                                            //  is used for the RWC feature for PORTint
    // Disable interrrupt at Global Level
    enableDisableInt(GLOBALint, DISABLE, 0);    // clearInt(intType, intState fields2clr)
                                            // int type=GLOBALint or PORTint
                                            // intState=DISABLE or ENABLE
                                            // fields2clr=dontcare for GLOBALint
                                            //  is used for the RWC feature for PORTint

    // Need to clear interrupts at Port Level followed by Global Level.
    // Ensure all pending Port Error and Status are cleared.
    clearIntOrErrorDiag(ERRORFIELDS, sataRegs->P0SERR);    // Clear P0SERR Register
    // Ensure all pending Port Interrupts and The Single Global Interrupt are cleared.
    clearIntOrErrorDiag(INTFIELDS, sataRegs->P0IS); // Clear P0IS and IS Regs
}


void invokeHBAReset() {
    // HBA Reset will not affect the following Registers settings of PxFB and PxCLB
    //  regs and HwInit fields of Port Registers are not affected.
    // To Do: Check if the Global Registers are affected. Spec mentions not affected.

    // Note: COMRESET OOB will not be sent to attached Device because this device supports
    //  Staggered Spinup capability and P0CMD.SUD is cleared to Zero when HBA Reset
    //  takes place. Software needs to invoke this if needed.

    // Most likely user want to ensure HBA comes up in its default operation state
    //  or has hung and is unable to idle the port when needing to perform an HBA
    //  reset. Regardless, there is no need to attempt to idle the HBA from
    //  running
    sataRegs->GHC |= (1 << AHCI_GHC_HR_SHIFT);

    // Max Spec time is 1 Second for Reset to complete.
    while((sataRegs->GHC & AHCI_GHC_HR) != 0) {
        waitForXms(WAIT_500_MILLISECONDS);
        waitForXms(WAIT_500_MILLISECONDS);
    }
}

char placeHbaInIdle(void) {
    // To Place HBA In IDLE, need to make sure both DMAs (Cmd List and Rcv FIS) are not running.
    //  Order of Disabling the DMA is important.

// Ensure that the Cmd List DMA is not running
    //  If is running, clear ST and wait for 500ms. Then Check CR.
    if (sataRegs->P0CMD & AHCI_PxCMD_ST) {
        sataRegs->P0CMD &= ~(1 << AHCI_PxCMD_ST_SHIFT);
        waitForXms(WAIT_500_MILLISECONDS);
    }// Wait another 500 Milliseconds for CR to clear. This is twice more than required.
    if (sataRegs->P0CMD & AHCI_PxCMD_CR)
        waitForXms(WAIT_500_MILLISECONDS);

    // If P0CMD.CR is still set, HBA probably has hunged. No need to continue.
    //  Need to perform HBA Reset.
    if (sataRegs->P0CMD & AHCI_PxCMD_CR)
        return(1);

// Ensure that the Receive FIS DMA is running.
    //  If is running, clear FRE and wait for 500ms. Then Check FR.
    if (sataRegs->P0CMD & AHCI_PxCMD_FRE) {
            sataRegs->P0CMD &= ~(1 << AHCI_PxCMD_FRE_SHIFT);
            waitForXms(WAIT_500_MILLISECONDS);
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
        }// Wait unti FR is Cleared.
    while (sataRegs->P0CMD & AHCI_PxCMD_FR)
        waitForXms(WAIT_500_MILLISECONDS);

    // If P0CMD.FRE is still set, HBA probably has hunged. No need to continue.
    //  Need to perform HBA Reset.
    if (sataRegs->P0CMD & AHCI_PxCMD_FRE)
        return(1);

    return(0);
}


void associateSysMem2Hba(Uint16 cmdSlot) {
    associtateCmdSlotWithCmdTable(cmdSlot);    // Assign Sys Mem allocated for Cmd Table to Cmd
List Slot
    //associtatePrdsWithCmdTable(cmdSlot);     // Assign PRD info to Cmd Table
    associtatePrdsWithCmdTable(cmdSlot, sata_input_filePageSize);
}

void associtateCmdSlotWithCmdTable(Uint16 cmdSlot) {
    CmdLists[cmdSlot].DW2.CmdTableAddLow=((unsigned int)&CmdTable[cmdSlot] & 0xFFFFFF80);
    CmdLists[cmdSlot].DW3.CmdTableAddHigh=0x0;
}

void associtatePrdsWithCmdTable(Uint16 cmdSlot, Uint16 fSize) {
    Uint16 fileSIZE, prdLength;
    for (fileSIZE=0; fileSIZE<fSize; fileSIZE++) {
        for (prdLength=0; prdLength<sata_input_prdLength; prdLength++) {
            // Command Header 0 PRD Descriptors 0 & 1 are Initialized.

CmdTable[cmdSlot].prdTable[(sata_input_prdLength*fileSIZE)+prdLength].DW0.DbaLow=(unsigned
int)&prdTableDataBuff[cmdSlot][prdLength];
            CmdTable[cmdSlot].prdTable[(sata_input_prdLength*fileSIZE)+prdLength].DW1.DbaHigh=0x0;

CmdTable[cmdSlot].prdTable[(sata_input_prdLength*fileSIZE)+prdLength].DW3.DataBC=sata_input_prd_da
taBuffLen-1;
        }
    }
}


void setSataSpeed(unsigned char iSpeed) {
    sataRegs->P0SCTL |= (iSpeed << AHCI_PxSCTL_PxSSTS_SPD_SHIFT);
    waitForXms(5); // This might not be necessary: wait a bit
}

char setupCfisEntriesForDataRdWr(CmdListHeader *CmdListNum, dataXferDir readOrWrite, xferProtocol
xferType) {
    // ****************************************************************************
    //$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ **** Initializing the Command Header *******
    // Other part of the Command List Structure, with the exception of Word 0 for the
    //  Command Slots[0] and [1] are already initialized when invoking sata_init_and_spin_up()
    //  function via associateMem2HBA() function.
    // Configure Word 0 of Command List
    CmdListNum->DW0.CmdLen=5;   // This is the length of H2D FIS. This might need changing
                                //   based on the Command issued to Device. Need to Check.
    CmdListNum->DW0.Atapi=0;    // Command is destined to HDD Like Device.
    CmdListNum->DW0.Prefetch=1; // Doesn't hurt prefetching so do it.
                                // WARNING: Do Not Prefetch if using:
                                //          => Command Queuing
                                //          => Port Multiplier
    CmdListNum->DW0.Reset=0;    // This is normally set to Zero unless a Soft Reset is required.
    CmdListNum-
>DW0.Bist=0;     // This is for entering test mode and should be cleared for normal operation.
    CmdListNum-
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
>DW0.Rok=0;      // For Normal operation require to Clear this bit so P0TFD and P0CI are modified
by HBA as appropriate.
                                //  Rok should be set for S/W Reset Command.
    CmdListNum-
>DW0.Pmp=0x0;    // Used only if an external Port Multiplier is attached and selects the Port of
the Port Multiplier.
    //$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    // The above DW0 fields usually would not change for Normal operation.

    if (readOrWrite == DATA_DIR_WR) // The Write setting here is based on the Data FIS direction.
        CmdListNum->DW0.Write=1;    //  Write=1/0=>Write/Read;
    else if (readOrWrite == DATA_DIR_RD) // The Write setting here is based on the Data FIS
direction.
        CmdListNum->DW0.Write=0;    //  Write=1/0=>Write/Read;
    else return(1);

//    CmdListNum-
>DW0.Prdtl=sata_input_prdLength;    // Need to update this when using DMA for Data transfer.
    CmdListNum-
>DW0.Prdtl=sata_input_filePageSize*sata_input_prdLength;    // Need to update this when using DMA
for Data transfer.

    // ********************************************************************************
    //$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ **** Initializing the Command FIS (H2D FIS) ******

    // Cmd FIS is made of a 20 Bytes size FIS.
    // FIS Fieds are Initialized into the alloted data buffers.
    // Strucutre type 'cmdFis' holds 8 of the 20 bytes of the FIS. These
    //  seven elements of the cfis are good enough for majority of command
    //  building tasks.
    // If need to access other cfis members, need to access the cfis member
    //  directly. Example of how to access the FIS Type, cfis Byte0.
    //     CmdTable[n].cfis.DW0.B0FisType=0x27;
    // FIS Type is hard coded within buildCmdFis function and Reserved Locations
    //  are also cleared to Zeros, so no need of iniitalizing this bytes is necessary here.

    // Make sure the Device 28 Bit LBA Address is initialize prior to calling this function.
    if (Dev28bitLbaAddress == 0xFFFFFFFF)
        return(1);

    myCmdFis.cfisByte1             = CMDFIS_BYTE1_C_IS_CMD_UPDATE;    // Bit7 of Byte1 is set for
Command Wr and Cleared for Control Wr
    //myCmdFis.cfisByte1             = CMDFIS_BYTE1_C_IS_CTRL_UPDATE; // Bit7 of Byte1 is Cleared
for Command Control Wr.

    if (readOrWrite == DATA_DIR_WR) {
        if (xferType == DMA_PROTOCOL) {
            myCmdFis.cfisCmd               = ATA_CMD_WRITE_DMA; // Uses 28-Bit Addressing.
            //myCmdFis.cfisCmd               = ATA_CMD_WRITE_DMA_EXT; // Uses 48-Bit Addressing.
        } else {
            myCmdFis.cfisCmd               = ATA_CMD_WRITE_SECTOR; // PIO Write Command: Uses 28-
Bit Addressing
        }
    }
    else {
        if (xferType == DMA_PROTOCOL) {
            myCmdFis.cfisCmd               = ATA_CMD_READ_DMA; // Uses 28-Bit Addressing.
            //myCmdFis.cfisCmd               = ATA_CMD_READ_DMA_EXT; // Uses 48-Bit Addressing.
        } else {
            if (PioCmd == ATA_CMD_IDENTIFY_DEVICE)
                myCmdFis.cfisCmd               = ATA_CMD_IDENTIFY_DEVICE; // PIO Read Command:
Uses 28-Bit Addressing.
            else
                myCmdFis.cfisCmd               = ATA_CMD_READ_SECTOR; // PIO Read Command: Uses
28-Bit Addressing.
        }
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
    }

    myCmdFis.cfisFeature          = 0x00;
    myCmdFis.cfisDw1SecNumLbaLow  = (Uint8) Dev28bitLbaAddress;
    myCmdFis.cfisDw1CylLowLbaMid  = (Uint8)(Dev28bitLbaAddress>>8);
    myCmdFis.cfisDw1CylHighLbahigh = (Uint8)(Dev28bitLbaAddress>>16);
    myCmdFis.cfisDw1Dev           = ( DEVICE_REG_USE_LBA_ADDRESSING |
                                    (Uint8)(Dev28bitLbaAddress>>24)
                                    );
//    myCmdFis.cfisDw3SecCnt        = (sata_input_prdLength*sata_input_prd_dataBuffLen)/512;
    myCmdFis.cfisDw3SecCnt        =
(sata_input_filePageSize*sata_input_prd_dataBuffLen*sata_input_prdLength)/512;
    myCmdFis.cfisDw3Ctrl          = 0x00;

    // The below require to be initialized at least once and can be ignored especially if
    //  not using 48-Bit Addressing. If use 48-Bit Addressing, then require constant
    //  maintenance prior to invoking a command.
    myCmdFis.cfisDw2SecNumLbaLowExp=0x00;
    myCmdFis.cfisDw2CylLowLbaMidExp=0x00;
    myCmdFis.cfisDw2CylHighLbahighExp=0x00;
    myCmdFis.cfisDw2FeatureExp=0x00;
    myCmdFis.cfisDw3SecCntExp=0x00;

    // Invalidate for future use.
    Dev28bitLbaAddress = 0xFFFFFFFF;

    return(0);
}

void buildCmdFis(CommandTable *CmdSlotNum) {
//    +----------+----------+--------------+-----------+
// DW0|  FEATURE | COMMAND  | c r r r port |FISTYPE 27h|
//    +----------+----------+--------------+-----------+
// DW1|  DEVICE  | LBA HIGH |  LBA MID     |  LBA LOW  |
//    +----------+----------+--------------+-----------+
// DW2|FETUREExp|LBAHIGHexp|  LBAMIDexp   | LBALOWexp |
//    +----------+----------+--------------+-----------+
// DW3| CONTROL  | RESERVED | SEC CNTexp   |  SEC CNT  |
//    +----------+----------+--------------+-----------+
// DW4| RESERVED | RESERVED |  RESERVED    | RESERVED  |
//    +----------+----------+--------------+-----------+
    CmdSlotNum->cfis.DW0.B0FisType=0x27;
    CmdSlotNum-
>cfis.DW0.BYTE1=myCmdFis.cfisByte1;                //Make Sure the 'C' bit field is correctly
set or cleared.
    CmdSlotNum->cfis.DW0.B2Cmd=myCmdFis.cfisCmd;
    CmdSlotNum->cfis.DW0.B3Feature=myCmdFis.cfisFeature;

    CmdSlotNum->cfis.DW1.B0LbaLow=myCmdFis.cfisDw1SecNumLbaLow;
    CmdSlotNum->cfis.DW1.B1LbaMid=myCmdFis.cfisDw1CylLowLbaMid;
    CmdSlotNum->cfis.DW1.B2LbaHigh=myCmdFis.cfisDw1CylHighLbahigh;
    CmdSlotNum->cfis.DW1.B3Device=myCmdFis.cfisDw1Dev;          //Make Sure 48-Bit or 28-
Bit Addressing is indicated here.


    CmdSlotNum->cfis.DW2.B0LbaLowExp=myCmdFis.cfisDw2SecNumLbaLowExp;    //0x0;
    CmdSlotNum->cfis.DW2.B1LbaMidExp=myCmdFis.cfisDw2CylLowLbaMidExp;    //0x0;
    CmdSlotNum->cfis.DW2.B2LbaHighExp=myCmdFis.cfisDw2CylHighLbahighExp; //0x0;
    CmdSlotNum->cfis.DW2.B3FeatureExp=myCmdFis.cfisDw2FeatureExp;        //0x0;

    CmdSlotNum->cfis.DW3.B0SecCnt=myCmdFis.cfisDw3SecCnt;
    CmdSlotNum->cfis.DW3.B1SecCntExp=myCmdFis.cfisDw3SecCntExp;          //0x0;
    CmdSlotNum->cfis.DW3.B2Rsv=0x0;
    CmdSlotNum->cfis.DW3.B3Control=myCmdFis.cfisDw3Ctrl;

    CmdSlotNum->cfis.DW4.DWResv=0x0;
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
}

char startCmdListProcessing(void) {
    // Make sure that a device is present and HBA has established communications.
    while ((sataRegs->P0SSTS & AHCI_PxSCTL_PxSSTS_DET) !=0x3);

    // Clear P0SERR.DIAG.X (RWC bit field) so that the P0TFD is updated by HBA.
    // Make sure it is cleared.
    sataRegs->P0SERR |= 0x04000000;

    // Make sure the Command List is not Running.

    if (sataRegs->P0CMD & AHCI_PxCMD_CR)
        return(1);
    //  Task file regs and look for Device ready status.
    while ((sataRegs->P0TFD & AHCI_PxTFD_STS_BSY_DRQ_ERR) != 0);

    // Make sure the the Receive FIS DMA is running.
    if ((sataRegs->P0CMD & (AHCI_PxCMD_FRE | AHCI_PxCMD_FRE)) !=
                            (AHCI_PxCMD_FRE | AHCI_PxCMD_FRE))
        return(1);

    // Enable the Cmd List DMA Engine.
    sataRegs->P0CMD |= AHCI_PxCMD_ST;

    // Wait here a bit until the Command List DMA Engine has started to run
    while ((sataRegs->P0CMD & AHCI_PxCMD_CR) == 0)
        waitForXms(1);

    return(0);
}

char submitCmd(Uint8 commandType, Uint8 commandSlot) {
    // Make sure both the Command List and Receive FIS DMAs are eanbled and running prior to
    //  submiting command
    Uint16 I;
    if ((sataRegs->P0CMD & (AHCI_PxCMD_FRE | AHCI_PxCMD_FRE | AHCI_PxCMD_CR | AHCI_PxCMD_ST)) !=
                            (AHCI_PxCMD_FRE | AHCI_PxCMD_FRE | AHCI_PxCMD_CR | AHCI_PxCMD_ST))
        return(1);

    switch (commandType) {
            case NON_QUEUED_CMD:
                sataRegs->P0CI |= (0x1 << commandSlot);
                break;

            case QUEUED_CMD:
                sataRegs->P0SACT |= (0x1 << commandSlot);
                sataRegs->P0CI |= (0x1 << commandSlot);
                break;

            default:
                break;
    }
    return(0);
}

// Basic Interrupt Handler Function.
void sataIsr(void) {
    intIsrCnt++;    // Count interrupt.
    intIsrFlag=1;

    // Ensure all pending Port Interrupts and The Single Global Interrupt are cleared.
    clearIntOrErrorDiag(INTFIELDS, sataRegs->P0IS); // Clear P0IS and IS Regs

}
```


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.3.3 Example on Initialization and Spinning Up Device

```
//char hetero_doTest(void) {
char sata_setup() {

    progStatus1='F';
    progStatus2='F';


    // Firmware HwInit Fields Configuration values.
    //  Need to configure this prior to calling sata_init_and_spin_up();
    swCtrlFeatures.capSMPS=1;   // Input Pin exist for external activity detection presence.
    swCtrlFeatures.capSSS=1;    // Always set to 1 in order to avoid spin up when HBA is powered.
    swCtrlFeatures.piPi=3;      // Supports a two HBA Ports (o and 1). Corresponding bit fields
of Pl need to be set.
    swCtrlFeatures.p0cmdEsp=0;  // The state of this bit is based on the support for eSATA.
CAP.SXS setting is the Logical OR of all Ports PxCMD.ESP. If any of the PxCMD.ESP is set, the
CAP.SXS will be set too.
        swCtrlFeatures.p0cmdHpcp=0; // Since ESP is mutually exclusive with HPCP (as mentioned in
spec) then HPCP should be set to 0.


    if(chceckSysMemorySize())
        for(;;);    // If program stays here, need to fix alignment issue.

    // Clear all allocated System Memory
    clearCmdList();        // Clear Cmd List allocated within Sys Mem
    clearCmdTables();    // Clear Cmd Tables allocated within Sys Mem
    clearRcvFis();        // Clear Receive FIS allocated within Sys Mem
    clearDmaBuffers();    // Clear all DMA Buffers

    // Make sure that both DMAs (Cmd List and Rcv FIS) are not running.
    if (placeHbaInIdle())
        invokeHBAReset();    // If unable to shut one or both DMAs, Perform HBA Reset.

    performFirmwareInit();

    //Start the Disk Drive (spin it up)
    if(spinUpDeviceAndWaitForInitToComplete())
        while(1); // Stay here if device signature does not match.

    //DMA Settings get affected by RESET
    cfgDmaSetting();

    initIntAndClearFlags();    // Disable CCC, Initialize 1ms Time, Clear Int and Flags


    if(intHandlingMethod == USE_INT_HANDLER) { //     USE_POLLING or USE_INT_HANDLER
        // Setup Interrupt Handler
        intIsrFlag=0;
        sata_intc_setup();    // Configure Interrupt Handler


        enableDisableInt(PORTint, ENABLE, 0xFFC000FF); // enableDisableInt(int type, intState,
specificField)
                                                       // int type=GLOBALint or PORTint
                                                       // intState=DISABLE or ENABLE
        enableDisableInt(GLOBALint, ENABLE, 0); // enableDisableInt(int type, intState,
specificField)
                                                       // int type=GLOBALint or PORTint
                                                       // intState=DISABLE or ENABLE
                                                       // specificField = Don't Care for
GLOBALint
        debugInt=0;
    } else
        enableDisableInt(PORTint, ENABLE, 0xFFC000FF); // enableDisableInt(int type, intState,
specificField)
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
                                                        // int type=GLOBALint or PORTint
                                                        // intState=DISABLE or ENABLE

    // Initialize Golden Data
    initMemory((Uint32 *)&prdTableDataBuff[0],
(sata_input_prdLength*sata_input_prd_dataBuffLen/4), 0x12345678, 0x01010101);
    // Invalidate Read Data Buffer
    initMemory((Uint32 *)&prdTableDataBuff[1],
(sata_input_prdLength*sata_input_prd_dataBuffLen/4), 0xDEADDEAD, 0x00000000);

    return(0);
}
```


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.3.4 Example of DMA Write Transfer

```
void performDmaWrite(void) {
// WRITE DMA
/*
  When performing the Non-Queued Command "Write DMA" the following captures the FISes that are
   communicated between Host (HBA) and Device (SpeedBridge or SATA Drive).

  HBA ==> Device    H2D FIS (Command FIS that has the "Write DMA" Command within H2D.Command
field).
  Device ==> HBA    DMA Activate FIS (Device notifies HBA that it's ready to receive data).
  HBA ==> Device    Data FIS (Actual Data Requested and pointed by the DMA/PRD Contents within
Command Table).
  Device ==> HBA    D2H FIS (Completion Status from Device)


  If need to capture Interrupt (just have only the Flag set) at:
    Global Level (IS Register), need to enable P0IE.Interrupt field.
    Port Level (No Enable bit needs to be set. P0IS will be set if D2H FIS DW0.Byte1.bit6 is set).

  If need to generate Interrupt at:
    Global Level (enable interrupt by setting GHC.IE bit field and P0IE.xxx should be set to
enable particular interrupt). Port Level interrupt handling is a subset of Global Level.
    Port Level (So long as Global Level interrupt is not enabled, GHC.IE is set, Interrupt will
not be sent to CPU just from Port Level only).
*/
// Write to Disk

    cmdSlot2Use=0;
    associateSysMem2Hba(cmdSlot2Use);    // Associate Sys Memory to CmdList and PRDs to Cmd Table

    //Initialize PRD Data Buffer Memory
//    initMemory((Uint32 *)&prdTableDataBuff[cmdSlot2Use],
(sata_input_prdLength*DATABUFFERLEN/4), 0x12345678, 0x01010101);

    // Configure Device 28 bit LBA Address (Start Address for Rd/Wr Transfer);
    Dev28bitLbaAddress              = sata_input_startAddress; // 28-Bit LBA Address

    // If problem exist when setuping CmdList, stay here. If not continue
    setupCfisEntriesForDataRdWr(&CmdLists[cmdSlot2Use], DATA_DIR_WR, DMA_PROTOCOL); // Usage:
setupCfisEntriesForDataRdWr(CmdListHeader *, DataDir, xferProtocol)
                                                                    // DataDir = DATA_DIR_RD or
DATA_DIR_WR, xferProtocol = PIO/DMA_PROTOCOL
    // Write cfis Data
    buildCmdFis(&CmdTable[cmdSlot2Use]);

    //getCmdFis(&CmdTable[0]);

    startCmdListProcessing(); // Stay here if unable to start processing command list.

    submitCmd (NON_QUEUED_CMD, cmdSlot2Use);    // Usage: CommandType (QUEUED(NON_QUEUED)_CMD,
Command Slot);

    if(intHandlingMethod == USE_INT_HANDLER) { //     USE_POLLING or USE_INT_HANDLER
        while(intIsrFlag==0);
        intIsrFlag=0;
    }
    else
        while(sataRegs-
>IS == 0);                           // Stay here until an interrupt is received.


    // If P0IS.DPS is set, A PRD with the I bit set has transferred all of its data.
    // This bitfield might not bit set for Non-Queued DMA. Applicable for Queued DMA
    //while(getRegStatus((Uint32*)&sataRegs-
>P0IS, AHCI_P0IS_DPS) == 0); // getRegStatus(ptr2int,field Mask)
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
    while(getRegStatus((Uint32*)&sataRegs-
>P0CI, (1<<cmdSlot2Use)) == (1<<cmdSlot2Use));    // Wait Until P0CI[ChHeader] to be cleared by
HBA

    // Clears both P0IS and IS register. Only Clears RWC bit fields. So, it is OK for this task.
    clearIntOrErrorDiag(INTFIELDS, sataRegs->P0IS);      // Clear P0IS Register
}
```

### 21.3.5 Example of DMA Read Transfer

```
void performDmaRead() {
// READ DMA
/*
  When performing the Non-Queued Command "Read DMA" the following captures the FISes that are
   communicated between Host (HBA) and Device (SpeedBridge or SATA Drive).

  HBA ==> Device    H2D FIS (Command FIS that has the "Read DMA" Command within H2D.Command
field).
  Device ==> HBA    Data FIS (Actual Data Requested and pointed by the DMA/PRD Contents within
Command Table).
  Device ==> HBA    D2H FIS (Completion Status from Device)

  If need to capture Interrupt (just have only the Flag set) at:
    Global Level (IS Register), need to enable P0IE.Interrupt field.
    Port Level (No Enable bit needs to be set. P0IS will be set if D2H FIS DW0.Byte1.bit6 is set).

  If need to generate Interrupt at:
    Global Level (enable interrupt by setting GHC.IE bit field and P0IE.xxx should be set to
enable particular interrupt). Port Level interrupt handling is a subset of Global Level.
    Port Level (So long as Global Level interrupt is not enabled, GHC.IE is set, Interrupt will
not be sent to CPU just from Port Level only).
*/
// Read from Disk
    cmdSlot2Use=1;
    associateSysMem2Hba(cmdSlot2Use);    // Associate Sys Memory to CmdList and PRDs to Cmd Table

    //Initialize PRD Data Buffer Memory
//    initMemory((Uint32 *)&prdTableDataBuff[cmdSlot2Use],
(sata_input_prdLength*DATABUFFERLEN/4), 0xDEADBEEF, 0x00000000);

    // Configure Device 28 bit LBA Address (Start Address for Rd/Wr Transfer);
    Dev28bitLbaAddress              = sata_input_startAddress; // 28-Bit LBA Address

    setupCfisEntriesForDataRdWr(&CmdLists[cmdSlot2Use], DATA_DIR_RD, DMA_PROTOCOL);    // Usage:
setupCfisEntriesForDataRdWr(CmdListHeader *, DataDir, xferProtocol)
                                                                    // DataDir = DATA_DIR_RD or
DATA_DIR_WR, xferProtocol = PIO/DMA_PROTOCOL
    // Write cfis Data
    buildCmdFis(&CmdTable[cmdSlot2Use]);

    //getCmdFis(&CmdTable[cmdSlot2Use]);

    startCmdListProcessing(); // Stay here if unable to start processing command list.


    submitCmd (NON_QUEUED_CMD, cmdSlot2Use); // Usage: CommandType (QUEUED(NON_QUEUED)_CMD,
Command Slot);

    if(intHandlingMethod == USE_INT_HANDLER) { //     USE_POLLING or USE_INT_HANDLER
        while(intIsrFlag==0);
        intIsrFlag=0;
    }
    else
        while(sataRegs-
>IS == 0);                            // Stay here until an interrupt is received.
```


Copyright © 2011–2016, Texas Instruments Incorporated

```
    // If P0IS.DPS is set, A PRD with the I bit set has transferred all of its data.
    // This bitfield might not bit set for Non-Queued DMA. Applicable for Queued DMA
    //while(getRegStatus((Uint32*)&sataRegs-
>P0IS, AHCI_P0IS_DPS) == 0); // getRegStatus(ptr2int,field Mask)

    while(getRegStatus((Uint32*)&sataRegs-
>P0CI, (1<<cmdSlot2Use)) == (1<<cmdSlot2Use));    // Wait Until P0CI[ChHeader] to be cleared by
HBA

    // Clears both P0IS and IS register. Only Clears RWC bit fields. So, it is OK for this task.
    clearIntOrErrorDiag(INTFIELDS, sataRegs->P0IS);      // Clear P0IS Register

}
```

## 21.4 Registers (Controller and PHY)

Due to the support of a standard controller, that is compliant with the AHCI 1.1 specifications, the memory-map of the controller and the register descriptions matches the memory-map documented within the standard. For features that are not part of the standard or are implementer specific, (for example, DMA burst control or built-in self test or PHY configuration registers) the reserved locations are used to document the necessary registers required by the non-standard features.

The sub-system core contains register space for global host programming and register space for port programming (the device supports two host ports and confirms to the AHCI specification calls for the support of multiple Host ports and the register space partitioning comes from this perspective). All registers that start below offset address 100h are global and meant to apply to the HBA, i.e., the registers settings entered at the global region applies to the single HBA port. Note that if the device happens to have multiple ports, then the registers within the global space would apply to all ports implemented. The Port 0 control registers reside at offset 100h and above. Dedicated registers that pertains to the specific port reside within this space. There are as many register banks as there are ports (for this device only one bank of registers is available for use).

The subsystem core contains register space for global host programming and space for port programming (the AHCI specification calls for the support of multiple host ports and the register space partitioning comes from this perspective). All registers that start below offset address 100h are global and meant to apply to the entire HBA. The Port 0 control registers reside at offset 100h to 17Fh. Dedicated registers that pertain to the specific port reside within this space. There are as many register banks as there are ports (for this device, one bank of registers is available).

Table 21-2 lists the registers of the SATA subsystem. Note that a special class of registers whose reset state is listed as W/RO (write/read only) requires a one time initialization after power-up. These registers are written once after a hard reset by firmware, and then remain as read-only thereafter. These registers are not affected by software reset.

The base address for the SATA register file is located at: 0x4A14_0000.

**Table 21-2. SATA Controller Registers**

| Address Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 0h | CAP | HBA Capabilities Register | Section 21.4.1 |
| 4h | GHC | Global HBA Control Register | Section 21.4.2 |
| 8h | IS | Interrupt Status Register | Section 21.4.3 |
| Ch | PI | Ports Implemented Register | Section 21.4.4 |
| 10h | VS | AHCI Version Register | Section 21.4.5 |
| 14h | CCC_CTL | Command Completion Coalescing Control Register | Section 21.4.6 |
| 18h | CCC_PORTS | Command Completion Coalescing Ports Register | Section 21.4.7 |
| A0h | BISTAFR | BIST Active FIS Register | Section 21.4.8 |
| A4h | BISTCR | BIST Control Register | Section 21.4.9 |
| A8h | BISTFCTR | BIST FIS Count Register | Section 21.4.10 |
| ACh | BISTSR | BIST Status Register | Section 21.4.11 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-2. SATA Controller Registers (continued)**

| Address Offset | Acronym | Register Description | Section |
|---|---|---|---|
| B0h | BISTDECR | BIST DWORD Error Count Register | Section 21.4.12 |
| E0h | TIMER1MS | BIST DWORD Error Count Register | Section 21.4.13 |
| E8h | GPARAM1R | Global Parameter 1 Register | Section 21.4.14 |
| ECh | GPARAM2R | Global Parameter 2 Register | Section 21.4.15 |
| F0h | PPARAMR | Port Parameter Register | Section 21.4.16 |
| F4h | Reserved | Reserved | |
| F8h | VERSIONR | Version Register | Section 21.4.17 |
| FCh | IDR | ID Register | Section 21.4.18 |
| 100h | P0CLB | Port 0 Command List Base Address Register | Section 21.4.19 |
| 108h | P0FB | Port 0 FIS Base Address Register | Section 21.4.20 |
| 110h | P0IS | Port 0 Interrupt Status Register | Section 21.4.21 |
| 114h | P0IE | Port 0 Interrupt Enable Register | Section 21.4.22 |
| 118h | P0CMD | Port 0 Command Register | Section 21.4.23 |
| 120h | P0TFD | Port 0 Task File Data Register | Section 21.4.24 |
| 124h | P0SIG | Port 0 Signature Register | Section 21.4.25 |
| 128h | P0SSTS | Port 0 Serial ATA Status Register | Section 21.4.26 |
| 12Ch | P0SCTL | Port 0 Serial ATA Control Register | Section 21.4.27 |
| 130h | P0SERR | Port 0 Serial ATA Error Register | Section 21.4.28 |
| 134h | P0SACT | Port 0 Serial ATA Active Register | Section 21.4.29 |
| 138h | P0CI | Port 0 Command Issue Register | Section 21.4.30 |
| 13Ch | P0SNTF | Port 0 Serial ATA Notification Register | Section 21.4.31 |
| 170h | P0DMACR | Port 0 DMA Control Register | Section 21.4.32 |
| 174h - 1FC | - | Reserved | |
| 1100h | IDLE | Idle Register | Section 21.4.33 |
| 1104h | CFGRX0 | PHY Configuration Receive 0 Register | Section 21.4.34 |
| 1108h | CFGRX1 | PHY Configuration Receive 1 Register | Section 21.4.35 |
| 110Ch | CFGRX2 | PHY Configuration Receive 2 Register | Section 21.4.36 |
| 1110h | CFGRX3 | PHY Configuration Receive 3 Register | Section 21.4.37 |
| 1114h | CFGRX4 | PHY Configuration Receive 4 Register | Section 21.4.38 |
| 1118h | STSRX | Receive Bus PHY-to-Controller Status Register (used for debug purposes | Section 21.4.39 |
| 111Ch | CFGTX0 | PHY Configuration Transmit 0 Register | Section 21.4.40 |
| 1120h | CFGTX1 | PHY Configuration Transmit 1 Register | Section 21.4.41 |
| 1124h | CFGTX2 | PHY Configuration Transmit 2 Register | Section 21.4.42 |
| 1128h | CFGTX3 | PHY Configuration Transmit 3 Register | Section 21.4.43 |
| 112Ch | CFGTX4 | PHY Configuration Transmit 4 Register | Section 21.4.44 |
| 1130h | STSTX | Receive Bus Controller -to- PHY Status Register (used for debug purposes | Section 21.4.45 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.1 HBA Capabilities Register (CAP)

The HBA capabilities register (CAP) indicates basic capabilities of the DWC SATA AHCI to the driver software.

The CAP register is shown in Figure 21-2 and described in Table 21-3.

**Figure 21-2. HBA Capabilities Register (CAP)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S64A | SNCQ | SSNTF | SMPS | SSS | SALP | SAL | SCLO | ISS | | SNZO | SAM | SPM | Rsvd |
| R-0 | R-1 | R-1 | W/RO-0 | W/RO-0 | R-1 | R-1 | R-1 | R-2h | | R-0 | R-1 | R-1 | R-0 |

| 15 | 14 | 13 | 12 | 8 | 7 | 6 | 5 | 4 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| PMD | SSC | PSC | NCS | | CCCS | EMS | SXS | NP | |
| R-1 | R-1 | R-1 | R-1Fh | | R-1 | R-0 | R-0 | R-1 | |

LEGEND: R/W = Read/Write; R = Read only; W/RO: written once after hard reset by firmware, then remain as read-only; *-n* = value after reset

**Table 21-3. HBA Capabilities Register (CAP) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | S64A | 0 | Indicates Support for 64-Bit Addressing. The SATASS only supports 32-bit addressing so this bit is always 0. |
| 30 | SNCQ | 1 | Supports Native Command Queuing. SATASS supports SATA native command queuing by handling DMA Setup FIS natively. |
| 29 | SSNTF | 1 | Supports SNotification Register. SATASS supports P0SNTF (SNotification) register and its associated functionality. |
| 28 | SMPS | 0 | Supports Mechanical Presence Switch. This bit is set by firmware when the platform supports a mechanical presence switch for hot plug operation. This should be written with Zero value. Can use GPIO to supplement this task. |
| 27 | SSS | 0 | Supports Staggered Spin-Up. Only writable once after power up. Set this bit during Firmware Initialization prior to attempting to spin up device (P0CMD[SUD]). |
| 26 | SALP | 1 | Supports Aggressive Link Power Management. SATASS supports auto-generating (Port-initiated) Link Layer requests to the PARTIAL or SLUMBER power management states when there are no commands to process. |
| 25 | SAL | 1 | Supports Activity LED. |
| 24 | SCLO | 1 | Supports Command List Override. Supports the P0CMD.CLO bit functionality for Port Multiplier devices enumeration. |
| 23-20 | ISS | 2h | Interface Speed Support. SATASS Supports 1.5 and 3 Gbps. |
| 19 | SNZO | 0 | Supports Non-Zero DMA Offsets. Not supported. |
| 18 | SAM | 1 | Supports AHCI Mode Only. SATASS supports AHCI mode only and does not support legacy, task-file based register interface. |
| 17 | SPM | 1 | Supports Port Multiplier. SATASS supports command-based switching Port Multiplier on any of its Ports. |
| 16 | Reserved | 0 | Reserved. |
| 15 | PMD | 1 | PIO Multiple DRQ Block. SATASS supports multiple DRQ block data transfers for the PIO command protocol. |
| 14 | SSC | 1 | Slumber State Capable. SATASS supports transitions to the interface SLUMBER power management state. |
| 13 | PSC | 1 | Partial State Capable. SATASS supports transitions to the interface PARTIAL power management state. |
| 12-8 | NCS | 1Fh | Number of Command Slots. SATASS supports 32 command slots per Port. |
| 7 | CCCS | 1 | Command Completion Coalescing Supported. SATASS supports command completion coalescing. |
| 6 | EMS | 0 | Enclosure Management Supported. Enclosure Management is not supported. |
| 5 | SXS | 1 | Supports External SATA. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-3. HBA Capabilities Register (CAP) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 4-0 | NP | | Number of Ports. 0's based value indicating the number of Ports supported by the SATSS. |
| | | 0 | 1 Port |
| | | 1 | 2 Ports |
| | | 2h - Fh | Reserved |

### 21.4.2 Global HBA Control Register (GHC)

The global HBA control register (GHC) provides various global functions for the SATASS.

The GHC register is shown in Figure 21-3 and described in Table 21-4.

**Figure 21-3. Global HBA Control Register (GHC)**

| 31 | 30 | 16 |
|---|---|---|
| AE | Reserved | |
| R-1 | R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | IE | HR |
| R-0 | | R/W-0 | W-0 |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

**Table 21-4. Global HBA Control Register (GHC) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | AE | 1 | AHCI Enable. This bit is always set since SATASS supports only AHCI mode as indicated by the SAM bit in the HBA capabilities register (CAP) = 1. |
| 30-2 | Reserved | 0 | Reserved. |
| 1 | IE | | Interrupt Enable. This global bit enables interrupts from the SATASS. This field is reset on Global reset (GHC.HR = 1). |
| | | 0 | All interrupt sources from all the Ports are disabled (masked). |
| | | 1 | Interrupts are enabled and any SATASS interrupt event causes interrupt output assertion. |
| 0 | HR | 0 | HBA Reset. When set by the software, this bit causes an internal Global reset of the SATASS. All state machines that relate to data transfers and queuing return to an idle state, and all the Ports are reinitialized by sending COMRESET if staggered spin-up is not supported. If staggered spin-up is supported, then it is the responsibility of the software to spin-up each Port after this reset has completed. The SATASS clears this bit when the reset action is done. A software write of 0 has no effect. |

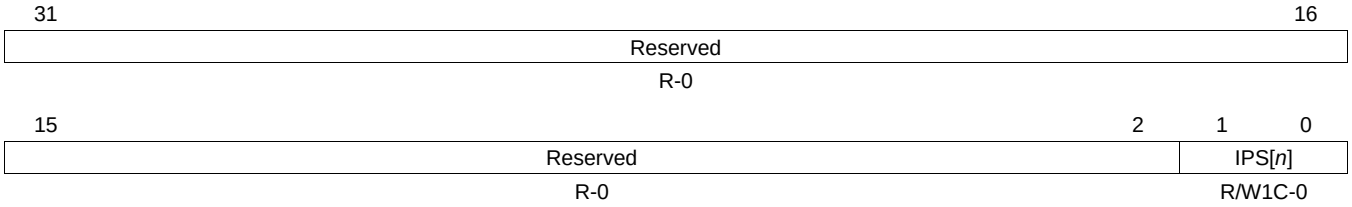

Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.3 Interrupt Status Register (IS)

The interrupt status register (IS) indicates which port inside of the subsystem has a pending interrupt.

The IS register is shown in Figure 21-4 and described in Table 21-5.

**Figure 21-4. Interrupt Status Register (IS)**




LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear; -*n* = value after reset

**Table 21-5. Interrupt Status Register (IS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved. |
| 1-0 | IPS[*n*] | 0-1 | Interrupt Pending Status (IPS[0] is for Port 0 and IPS[1] is for Port[1]).. If a bit *n* is set to 1 If set, the corresponding Port has an interrupt pending. Software can use this information to determine which Ports require service after an interrupt. The bits of this field are set by the Ports that have interrupt events pending in the port interrupt status register (P0IS) bits and enabled by the port interrupt enable register (P0IE) bits. Set bits are cleared by the software writing 1 to all bits to clear. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.4 Ports Implemented Register (PI)

The ports implemented register (PI) indicates which ports are exposed by the SATASS and are available for software to use. It is loaded by the BIOS. For example, the SATASS supports 2 Ports as indicated in the CAP.NP, only Port 1 could be available, while Port 0 is unavailable. During Firmware Initialization the software needs to indicate that Port 0 is not available by setting only PI[bit1] only. This means the Port Enabled is a subset of Port available. Note that a minimum of one port needs to be enabled.

The PI register is shown in Figure 21-5 and described in Table 21-6.

**Figure 21-5. Ports Implemented Register (PI)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | PI[*n*] | |
| R-0 | | W/RO-0 | |

LEGEND: R/W = Read/Write; R = Read only; W/RO: written once after hard reset by firmware, then remain as read-only; -*n* = value after reset

**Table 21-6. Ports Implemented Register (PI) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved. |
| 1-0 | PI[*n*] | 0-1 | Ports Implemented. This register is bit significant. If a bit *n* is set to 1, the corresponding Port is available for software to use. If a bit *n* is cleared to 0, the Port is not available for software to use. The maximum number of bits that can be set is 2 for a module with two HBA ports.<br>**Note:** The contents of this register are relevant to the command completion coalescing ports register (CCC_PORTS). |

### 21.4.5 AHCI Version Register (VS)

The AHCI version register (VS) indicates the version of the Synopsis AHCI core embedded in the SATASS.

The VS register is shown in Figure 21-6 and described in Table 21-7.

**Figure 21-6. AHCI Version Register (VS)**

| 31 | 16 |
|---|---|
| MJR | |
| R-1 | |

| 15 | 0 |
|---|---|
| MNR | |
| R-100h | |

LEGEND: R = Read only; -*n* = value after reset

**Table 21-7. AHCI Version Register (VS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | MJR | 1 | Major Revision Number. |
| 15-0 | MNR | 100h | Minor Revision Number. |

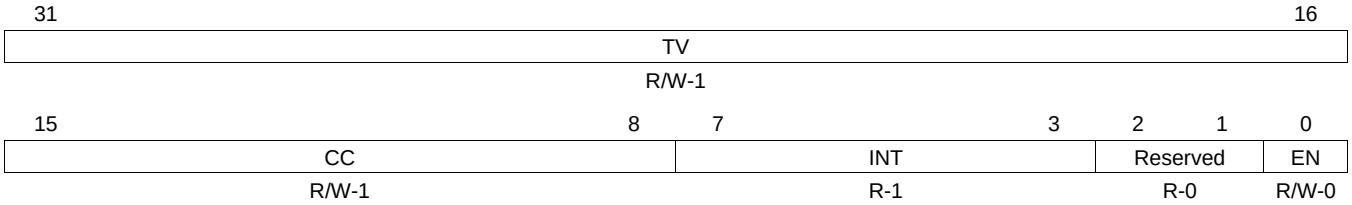

Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.6 Command Completion Coalescing Control Register (CCC_CTL)

The command completion coalescing control register (CCC_CTL) is used to configure the command completion coalescing (CCC) feature for the SATASS core. It is reset on Global reset.

The CCC_CTL register is shown in Figure 21-7 and described in Table 21-8.

**Figure 21-7. Command Completion Coalescing Control Register (CCC_CTL)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-8. Command Completion Coalescing Control Register (CCC_CTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | TV | 0-FFFFh | Time-out value. This bit field specifies the CCC time-out value in 1 ms intervals. Software loads this value prior to enabling CCC. This bit field is:<br>• Read/Write (R/W) when EN = 0<br>• Read only (R) when EN = 1<br>A time-out value of 0 is reserved and should not be used. |
| 15-8 | CC | 0-FFh | Command Completions. This bit field specifies the number of command completions that are necessary to cause a CCC interrupt. Software loads this value prior to enabling CCC. This bit field is:<br>• Read/Write (R/W) when EN = 0<br>• Read only (R) when EN = 1<br>A value of 0 disables CCC interrupts being generated based on the number of commands completed, that is, CCC interrupts are only generated based on the timer in this case. |
| 7-3 | INT | 0-1Fh | Interrupt. This bit field specifies the interrupt used by the CCC feature, using the number of ports configured for the core. For a single Port instantiation, INT should be programmed to 1.<br>When a CCC interrupt occurs, the IS.IPS[INT] bit is set to 1. |
| 2-1 | Reserved | 0 | Reserved. |
| 0 | EN | | CCC feature enable. When EN = 1, software can not change the bit fields: TV and CC. |
| | | 0 | CCC feature is disabled and no CCC interrupts are generated. |
| | | 1 | CCC feature is enabled and CCC interrupts may be generated based on the time-out or command completion conditions.<br>**Note:** When EN = 1, software can not change the fields : CCC_CTL.TV and CCC_CTL.CC. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.7 Command Completion Coalescing Ports Register (CCC_PORTS)

The command completion coalescing ports register (CCC_PORTS) specifies the Ports that are coalesced as part of the command completion coalescing (CCC) feature when CCC_CTL.EN = 1. It is reset on Global reset.

The CCC_PORTS register is shown in Figure 21-8 and described in Table 21-9.

**Figure 21-8. Command Completion Coalescing Ports Register (CCC_PORTS)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | PRT | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 21-9. Command Completion Coalescing Ports Register (CCC_PORTS) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved |
| 1-0 | PRT | | This field is bit significant. Each bit corresponds to a particular Port, where bit 0 corresponds to Port 0 and bit 1 corresponds to Port 1. Bits set to 1 must also have the corresponding bit set to 1 in the ports implemented register (PI). |
| | | 0 | The corresponding Port is not part of the CCC feature. |
| | | 1 | The corresponding Port is part of the CCC feature. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.8 BIST Active FIS Register (BISTAFR)

The BIST active FIS register (BISTAFR) is shown in Figure 21-9 and described in Table 21-10.

**Figure 21-9. BIST Active FIS Register (BISTAFR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| NCP | | PD | |
| R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 21-10. BIST Active FIS Register (BISTAFR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-8 | NCP | 0-FFh | Non-Compliant Pattern. Least significant byte of the received BIST Activate FIS second DWORD (bits [7:0]). This value defines the required pattern for the far-end transmit only modes (PD=80h or A0h). If none of these values are decoded, the simultaneous switching pattern is transmitted by default. |
| | | 0-49h | Reserved |
| | | 4Ah | High frequency test pattern (HFTP) |
| | | 4Bh-77h | Reserved |
| | | 78h | Mid frequency test pattern (MFTP) |
| | | 79h-7Dh | Reserved |
| | | 7Eh | Low frequency test pattern (LFTP) |
| | | 7Fh | Simultaneous switching outputs pattern (SSOP) |
| | | 80h-8Ah | Reserved |
| | | 8Bh | Lone Bit pattern (LBP) |
| | | 8Ch-AAh | Reserved |
| | | ABh | Low frequency spectral component pattern (LFSCP) |
| | | ACh-B4h | Reserved |
| | | B5h | High transition density pattern (HTDP) |
| | | B6h-F0h | Reserved |
| | | F1h | Low transition density pattern (LTDP) |
| | | F2h-FFh | Reserved |
| 7-0 | PD | 0-FFh | Pattern Definition. Indicates the pattern definition field of the received BIST Activate FIS (bits [23:16]) of the first DWORD. It is used to put the SATASS in one of the following BIST modes. All reserved values should not be used by the device; otherwise, the FIS is negatively acknowledged with R_ERRp. For far-end transmit only modes, the NCP bit field contains the required data pattern. |
| | | 0-7h | Reserved |
| | | 8h | Far-end analog (if PHY supports this mode) |
| | | 9h-Fh | Reserved |
| | | 10h | Far-end retimed |
| | | 11h-7Fh | Reserved |
| | | 80h | Far-end transmit only |
| | | 81h-9Fh | Reserved |
| | | A0h | Far-end transmit only with scrambler bypassed |
| | | A1h-FFh | Reserved |

### 21.4.9 BIST Control Register (BISTCR)

The BIST control register (BISTCR) is shown in Figure 21-10 and described in Table 21-11.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 21-10. BIST Control Register (BISTCR)**

| 31 | | | | | | | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 23 | | | | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | TXO | CNTCLR | NEALB |
| R-0 | | | | | W-0 | W-0 | R/W-0 |

| 15 | | | | 11 | 10 | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | LLC | | |
| R-0 | | | | | R/W-7h | | |

| 7 | 6 | 5 | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | ERREN | FLIP | PV | PATTERN | | | |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | | | |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

**Table 21-11. BIST Control Register (BISTCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-19 | Reserved | 0 | Reserved. |
| 18 | TXO | 0 | Transmit Only. This bit is used to initiate transmission of one of the non-compliant patterns defined by the BISTCR.PATTERN value when the device is disconnected. |
| 17 | CNTCLR | 0 | Counter Clear. This bit clears BIST error count registers. Writing a 1 clears BISTFCTR, BISTSR, and BISTDECR registers. |
| 16 | NEALB | | Near-end Analog Loopback. Places the Port PHY into near-end analog loopback mode. |
| | | 0 | Near-end analog loopback is not requested. |
| | | 1 | Initiates a near-end analog loopback request. BISTCR.CP field contains the appropriate pattern. This mode should be initiated either in the PARTIAL or SLUMBER power mode, or with the device disconnected from the Port PHY (Link NOCOMM state). BIST Activate FIS is not sent to the device in this mode. |
| 15-11 | Reserved | 0 | Reserved. |
| 10-8 | LLC | 0-7h | Link Layer Control. This bit field controls the Port Link Layer functions: scrambler, descrambler, and repeat primitive drop (RPD). In normal mode, the functions scrambler, descrambler, or RPD are changed only during Port reset (P0SCTL.DET = 1). Note the different meanings for normal and BIST modes of operation: |
| | | | **Bit 10 (RPD):** |
| | | 0 | Repeat primitive drop function is disabled in normal mode, enabled in BIST mode. |
| | | 1 | Repeat primitive drop function is enabled in normal mode, disabled in BIST mode. |
| | | | **Bit 9 (DESCRAM):** |
| | | 0 | Descrambler is disabled in normal mode, enabled in BIST mode. |
| | | 1 | Descrambler is enabled in normal mode, disabled in BIST mode. |
| | | | **Bit 8 (SCRAM):** |
| | | 0 | Scrambler is disabled in normal mode, enabled in BIST mode. |
| | | 1 | Scrambler is enabled in normal mode, disabled in BIST mode. |
| | | | The SCRAM bit is cleared (enabled) by the Port when the Port enters a responder far-end transmit BIST mode with scrambling enabled (BISTAFR.PD = 80h). |
| 7 | Reserved | 0 | Reserved. |
| 6 | ERREN | | Error Enable. Used to allow or filter (disable) PHY internal errors outside the FIS boundary to set corresponding port serial ATA error register (P0SERR) bits. |
| | | 0 | Filter errors outside the FIS, allow errors inside the FIS. |
| | | 1 | Allow errors outside or inside the FIS. |
| 5 | FLIP | 0-1 | Flip Disparity. Enables changing disparity of the current test pattern to the opposite every time its state is changed by software. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-11. BIST Control Register (BISTCR) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 4 | PV | | Pattern Version. Selects either short or long version of the SSOP, HTDP, LTDP, LFSCP, and COMP patterns. |
| | | 0 | Short pattern version |
| | | 1 | Long pattern version |
| 3-0 | PATTERN | 0-Fh | Defines one of the following SATA compliant patterns for far-end retimed/ far-end analog/near-end analog initiator modes, or non-compliant patterns for transmit-only responder mode when initiated by software writing to the BISTCR.TXO bit. |
| | | 0 | Simultaneous switching outputs pattern (SSOP) |
| | | 1h | High-transition density pattern |
| | | 2h | Low-transition density pattern |
| | | 3h | Low-frequency spectral component pattern (LFSCP) |
| | | 4h | Composite pattern (COMP) |
| | | 5h | Lone bit pattern (LBP) |
| | | 6h | Mid-frequency test pattern (MFTP) |
| | | 7h | High-frequency test pattern (HFTP) |
| | | 8h | Low-frequency test pattern (LFTP) |
| | | 9h-Fh | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.10 BIST FIS Count Register (BISTFCTR)

The BIST FIS count register (BISTFCTR) contains the received BIST FIS count in the loopback initiator far-end retimed, far-end analog, and near-end analog modes. It is updated each time a new BIST FIS is received. It is reset by Global reset, Port reset (COMRESET) or by setting the BISTCR.CNTCLR bit. This register does not roll over and freezes when the FFFF FFFFh value is reached. It takes approximately 65 hours of continuous BIST operation to reach this value.

The BISTFCTR register is shown in Figure 21-11 and described in Table 21-12.

**Figure 21-11. BIST FIS Count Register (BISTFCTR)**

| 31 | 0 |
|---|---|
| BISTFCTR | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 21-12. BIST FIS Count Register (BISTFCTR) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | BISTFCTR | 0-FFFF FFFFh | Received BIST FIS Count. |

### 21.4.11 BIST Status Register (BISTSR)

The BIST status register (BISTSR) contains errors detected in the received BIST FIS in the loopback initiator far-end retimed, far-end analog, and near-end analog modes. It is not meaningful in responder mode. It is updated each time a new BIST FIS is received. It is reset by Global reset, Port reset (COMRESET) or by setting the BISTCR.CNTCLR bit.

The BISTSR register is shown in Figure 21-12 and described in Table 21-13.

**Figure 21-12. BIST Status Register (BISTSR)**

| 31 | 24 | 23 | 16 |
|---|---|---|---|
| Reserved | | BRSTERR | |
| R-0 | | R-0 | |

| 15 | 0 |
|---|---|
| FRAMERR | |
| W-0 | |

LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 21-13. BIST Status Register (BISTSR) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-24 | Reserved | 0 | Reserved. |
| 23-16 | BRSTERR | 0-FFh | Burst Error. This bit field contains the burst error count. It is accumulated each time a burst error condition is detected: DWORD error is detected in the received frame and 1.5 seconds (27,000 frames) passed since the previous burst error was detected. The BRSTERR value does not roll over and freezes at FFh. This bit field is updated if BIST_MODE = DWORD. |
| 15-0 | FRAMERR | 0-FFFFh | Frame Error. This bit field contains the frame error count. It is accumulated (new value is added to the old value) each time a new BIST frame with a CRC error is received. The FRAMERR value does not roll over and freezes at FFFFh. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.12 BIST DWORD Error Count Register (BISTDECR)

The BIST DWORD error count register (BISTDECR) contains the number of DWORD errors detected in the received BIST frame in the loopback initiator far-end retimed, far-end analog, and near-end analog modes. It is updated each time a new BIST frame is received. It is reset by Global reset, Port reset (COMRESET) or by setting the BISTCR.CNTCLR bit. This register is updated only when the parameter BIST_MODE=DWORD.

The BISTDECR register is shown in Figure 21-13 and described in Table 21-14.

**Figure 21-13. BIST DWORD Error Count Register (BISTDECR)**

| 31 | 0 |
|---|---|
| DWERR | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 21-14. BIST DWORD Error Count Register (BISTDECR) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | DWERR | 0-FFFF FFFFh | DWORD Error Count. This bit field contains the DWORD error count. It is accumulated (new value is added to the old value) each time a new BIST frame is received. The DWERR value does not roll over and freezes if it exceeds FFFF F000h. |

### 21.4.13 BIST DWORD Error Count Register (TIMER1MS)

The BIST DWORD error count register (TIMER1MS) is used to generate 1ms tick for the command completion coalescing (CCC) logic based on the OCP clock frequency sourced to the SATA controller. Software must initialize this register with the required value after power up before using the CCC feature. This register is reset to 100,000d (corresponding to a VBUS Clock frequency of 100 MHz hclk) on power up and is not affected by Global reset.

The TIMER1MS register is shown in Figure 21-14 and described in Table 21-15.

**Figure 21-14. BIST DWORD Error Count Register (TIMER1MS)**

| 31 | 20 | 19 | 0 |
|---|---|---|---|
| Reserved | | TIMV | |
| R-0 | | R/W-186A0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-15. BIST DWORD Error Count Register (TIMER1MS) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-20 | Reserved | 0 | Reserved. |
| 19-0 | TIMV | 0-F FFFFh | 1ms Timer Value. This bit field contains the cycle count of the SATA VBUS Clock Cycle generating 1ms tick. This bit field is:<br>• Read/Write (R/W) when CCC_CTL.EN = 0<br>• Read only (R) when CCC_CTL.EN = 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.14 Global Parameter 1 Register (GPARAM1R)

The global parameter 1 register (GPARAM1R) is a read-only register that contains encoded information about the configuration of the embedded Synopsis AHCI SATA Core.

The GPARAM1R register is shown in Figure 21-15 and described in Table 21-16.

**Figure 21-15. Global Parameter 1 Register (GPARAM1R)**

| 31 | 30 | 29 | 28 | 27 | 26 | 24 |
|---|---|---|---|---|---|---|
| ALIGN_M | RX_BUFFER | PHY_DATA | | PHY_RST | PHY_CTRL | |
| R-1 | R-1 | R-0 | | R-0 | R-1Ah | |

| 23 | 21 | 20 | 15 |
|---|---|---|---|
| PHY_CTRL | | PHY_STAT | |
| R-1Ah | | R-2h | |

| 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| LATCH_M | BIST_M | PHY_TYPE | Reserved | RETURN_ERR | AHB_ENDIAN | |
| R-0 | R-0 | R-0 | R-0 | R-1 | R-2h | |

| 7 | 6 | 5 | 3 | 2 | 0 |
|---|---|---|---|---|---|
| S_HADDR | M_HADDR | S_HDATA | | M_HDATA | |
| R-0 | R-0 | R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 21-16. Global Parameter 1 Register (GPARAM1R) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | ALIGN_M | 1 | Rx Data Alignment. Data is always aligned. |
| 30 | RX_BUFFER | 1 | Rx Data Buffer. Core includes an Rx Data Buffer. |
| 29-28 | PHY_DATA | 0 | PHY Data Width. Indicates width = 0 (8-bits). |
| 27 | PHY_RST | 0 | PHY Reset Mode. Indicates that the PHY reset output is active-low. |
| 26-21 | PHY_CTRL | 1Ah | PHY Control Width. Indicates that there are 32-bits of PHY control. |
| 20-15 | PHY_STAT | 2h | PHY Status Width. Indicates that there are 32-bits of PHY status. |
| 14 | LATCH_M | 0 | Latch Mode. Indicates that the subsystem does not include latches. |
| 13 | BIST_M | 0 | BIST Loopback Checking Depth. Checks errors per FIS not DWORD. |
| 12 | PHY_TYPE | 0 | PHY Interface Type. Indicates a non-Synopsis PHY. |
| 11 | Reserved | 0 | Reserved. |
| 10 | RETURN_ERR | 1 | AHB Error Response. Indicates AHB Errors are returned. |
| 9-8 | AHB_ENDIAN | 2h | Bus Endianness. Indicates that endianness may be configured by input pin. |
| 7 | S_HADDR | 0 | Slave address bus width. Indicates 32-bit wide address bus. |
| 6 | M_HADDR | 0 | Master address bus width. Indicates 32-bit wide address bus. |
| 5-3 | S_HDATA | 0 | Slave Data Bus Width. Indicates 32-bit data bus. |
| 2-0 | M_HDATA | 0 | Master Data Bus Width. Indicates 32-bit data bus. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.15 Global Parameter 2 Register (GPARAM2R)

The global parameter 2 register (GPARAM2R) is a read-only register that contains encoded information about the configuration of the embedded Synopsis AHCI SATA Core.

The GPARAM2R register is shown in Figure 21-16 and described in Table 21-17.

**Figure 21-16. Global Parameter 2 Register (GPARAM2R)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | DEV_CP | DEV_MP | ENCODE_M | RXOOB_CLK_M | RXOOB_M | TX_OOB_M | RXOOB_CLK | | | | | | | | |
| R-0 | R-1 | R-1 | R-1 | R-0 | R-0 | R-1 | R-96h | | | | | | | | |

LEGEND: R = Read only; -*n* = value after reset

**Table 21-17. Global Parameter 2 Register (GPARAM2R) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-15 | Reserved | 0 | Reserved. |
| 14 | DEV_CP | 0 | Cold Presence Detect. CPD is supported in SATASS. |
| 13 | DEV_MP | 0 | Mechanical Presence Switch. MP is supported in SATASS. |
| 12 | ENCODE_M | 1 | 8b/10b Encoding/Decoding. Indicates 8-bit/10-bit encoding and decoding are done in the Synopsis core. |
| 11 | RXOOB_CLK_M | 0 | Rx OOB Clock Mode. Indicates that the Rx OOB Functions are on the Rx Clock. |
| 10 | RX_OOB_M | 1 | Rx OOB Mode. Indicates that Rx Out-of-Band signals are detected in the PHY. |
| 9 | TX_OOB_M | 1 | Tx OOB Mode. Indicates that Tx Out-of-Band signaling is done by the Synopsis Core. |
| 8-0 | RXOOB_CLK | 96h | Rx OOB Clock Frequency. Reflects the hexadecimal value of the RXOOB_CLK_FREQ parameter. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.16 Port Parameter Register (PPARAMR)

The port parameter register (PPARAMR) is a read-only register that contains encoded information about the selected Port configuration parameters settings. The Port is selected by the TESTR.PSEL field. Note, there are two HBA ports on this sub-system and the configuration setting applies to both ports.

The PPARAMR register is shown in Figure 21-17 and described in Table 21-18.

**Figure 21-17. Port Parameter Register (PPARAMR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 10 | 9 | 8 |
|---|---|---|---|
| Reserved | | TX_MEM_M | TX_MEM_S |
| R-0 | | R-0 | R-0 |

| 7 | 6 | 5 | 3 | 2 | 0 |
|---|---|---|---|---|---|
| RX_MEM_M | RX_MEM_S | X_FIFO_DEPTH | | RX_FIFO_DEPTH | |
| R-0 | R-0 | R-0x2 | | R-0x3 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 21-18. Port Parameter Register (PPARAMR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reserved. |
| 9 | TX_MEM_M | 0 | Tx FIFO Memory Read Port Type. Indicates that the Tx FIFO memory is asynchronous read. |
| 8 | TX_MEM_S | 0 | Tx FIFO Memory Type. Indicates that the Tx FIFO memory is outside of the core (but still inside the subsystem wrapper). |
| 7 | RX_MEM_M | 0 | Rx FIFO Memory Read Port Type. Indicates that the Rx FIFO memory is asynchronous read. |
| 6 | RX_MEM_S | 0 | Rx FIFO Memory Type. Indicates that the Rx FIFO memory is outside of the core (but still inside the subsystem wrapper). |
| 5-3 | TX_FIFO_DEPTH | 2h | Tx FIFO Depth. Indicates that the depth of the Tx FIFO is 64 DWORDS. |
| 2-0 | RX_FIFO_DEPTH | 3h | Rx FIFO Depth. Indicates that the depth of the Rx FIFO is 128 DWORDS. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.17 Version Register (VERSIONR)

The version register (VERSIONR) contains the 32-bit SATASS version.

The VERSIONR register is shown in Figure 21-18 and described in Table 21-19.

**Figure 21-18. Version Register (VERSIONR)**

LEGEND: R = Read only; -*n* = value after reset

**Table 21-19. Version Register (VERSIONR) Field Description**

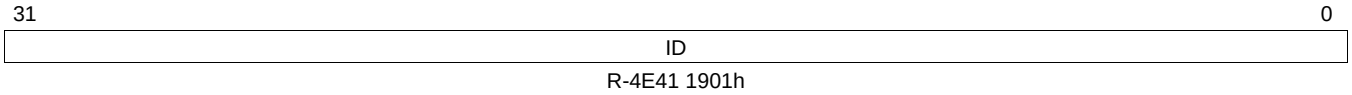
| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | VERSION | 3134 302Ah | 32-bit version of SATASS. |

### 21.4.18 ID Register (IDR)

The ID register (IDR) contains the 32-bit SATASS ID.

The IDR register is shown in Figure 21-19 and described in Table 21-20.

**Figure 21-19. ID Register (IDR)**




LEGEND: R = Read only; -*n* = value after reset

**Table 21-20. ID Register (IDR) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | ID | 4E41 1901h | 32-bit ID of SATASS. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.19 Port Command List Base Address Register (P0CLB)

The port command list base address register (P0CLB) contains the 32-bit base address of the command list.

The P0CLB register is shown in Figure 21-20 and described in Table 21-21.

**Figure 21-20. Port Command List Base Address Register (P0CLB)**

| 31 | 10 | 9 | 0 |
|---|---|---|---|
| CLB | | Reserved | |
| R/W-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-21. Port Command List Base Address Register (P0CLB) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | CLB | 0-3F FFFFh | Command List Base Address. Indicates the 32-bit base physical address for the command list for this port. This base is used when fetching commands to execute. The structure pointed to by this address range is 1Kbyte in length. This address must be 1Kbyte-aligned as indicated by bits [9:0] being read-only. |
| 9-0 | Reserved | 0 | Reserved. |

### 21.4.20 Port FIS Base Address Register (P0FB)

The port FIS base address register (P0FB) contains the 32-bit base address of the destination for incoming received FISes.

The P0FB register is shown in Figure 21-21 and described in Table 21-22.

**Figure 21-21. Port FIS Base Address Register (P0FB)**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| FB | | Reserved | |
| R/W-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-22. Port FIS Base Address Register (P0FB) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | FB | 0-FF FFFFh | FIS Base Address. Indicates the 32-bit base physical address for received FISes. The structure pointed to by this address range is 256 bytes in length. This address must be 256 byte-aligned as indicated by bits [7:0] being read-only. |
| 7-0 | Reserved | 0 | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.21 Port Interrupt Status Register (P0IS)

The port interrupt status register (P0IS) is used to generate SATASS interrupts when any of the bits are set. Bits in this register are set by some internal conditions, and cleared by software writing ones in the positions it wants to clear. This register is reset on Global reset.

The P0IS register is shown in Figure 21-22 and described in Table 21-23.

**Figure 21-22. Port Interrupt Status Register (P0IS)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Rsvd | TFES | HBFS | HBDS | IFS | INFS | Rsvd | OFS | IPMS | PRCS | Reserved | |
| R-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R-0 | R/W1C-0 | R/W1C-0 | R-0 | R-0 | |

| 15 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | DMPS | PCS | DPS | UFS | SDBS | DSS | PSS | DHRS |
| R-0 | | R/W1C-0 | R-0 | R/W1C-0 | R-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 |

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear; -*n* = value after reset

**Table 21-23. Port Interrupt Status Register (P0IS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved. Since no dedicated pins for CPD detection/control exist for the device, this bit is reserved. If need to support this feature, use of GPIO is recommended. |
| 30 | TFES | 0-1 | Task File Error Status. This bit is set whenever the P0TFD register is updated by the device and the error bit (P0TFD.STS[0]) is set. |
| 29 | HBFS | 0-1 | Host Bus Fatal Error Status. This bit is set when SATASS bus master detects an ERROR response from the slave. |
| 28 | HBDS | 0-1 | Host Bus Data Error Status. This bit is always cleared to 0. |
| 27 | IFS | 0-1 | Interface Fatal Error Status. This bit is set if any of the following conditions is detected:<br>• SYNC escape is received from the device during H2D Register or Data FIS transmission.<br>• One or more of the following errors are detected during Data FIS transfer: Protocol (P0SERR.ERR_P), CRC (P0SERR.DIAG_C), Handshake (P0SERR.DIAG_H), PHY Not Ready (P0SERR.ERR_C).<br>• Unknown FIS is received with good CRC, but the length exceeds 64 bytes.<br>• PRD table byte count is zero.<br>Port DMA transitions to a fatal state until software clears P0CMD.ST bit or resets the interface by way of Port or Global reset. |
| 26 | INFS | 0-1 | Interface Non-fatal Error Status. This bit is set if any of the following conditions is detected:<br>• One or more of the following errors are detected during non-data FIS transfer: Protocol (P0SERR.ERR_P), CRC (P0SERR.DIAG_C), Handshake (P0SERR.DIAG_H), PHY Not Ready (P0SERR.ERR_C).<br>• Command list underflow during read operation (DMA read) when software builds command table that has more total bytes than the transaction given to the device.<br>In both cases, Port operation continues normally. If error is detected during non-data FIS transmission, this FIS is retransmitted continuously until it succeeds or software times out and resets the interface. |
| 25 | Reserved | 0 | Reserved. |
| 24 | OFS | 0-1 | Overflow Status. This bit is set if command list overflow is detected during read or write operation when software builds a command table that has fewer total bytes than the transaction given to the device.<br>Port DMA transitions to a fatal state until software clears P0CMD.ST bit or resets the interface by way of Port or Global reset. |
| 23 | IPMS | 0-1 | Incorrect Port Multiplier Status. Indicates that the HBA received a FIS from a device whose Port Multiplier field did not match what was expected.<br>This bit may be set during enumeration of devices on a Port Multiplier due to the normal Port Multiplier enumeration process. The software should only use the IPMS bit after enumeration is complete on the Port Multiplier. |
| 22 | PRCS | 0-1 | PHY Ready Change Status. When set to 1, indicates the internal PHY Ready signal changed state. This bit reflects the state of the P0SERR.DIAG_N bit. To clear this bit, the software must clear the P0SERR.DIAG_N bit to 0. |
| 21-8 | Reserved | 0 | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-23. Port Interrupt Status Register (P0IS) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7 | DMPS | 0-1 | Device Mechanical Presence Status. This bit is set when the state of SATA_MP_SWITCH input pin changes as a result of a mechanical switch attached to this port being opened or closed. This bit is only valid if both CAP.SMPS and P0CMD.MPSP are set to 1. |
| 6 | PCS | | Port Connect Change Status. This bit reflects the state of the P0SERR.DIAG_X bit. This bit is only cleared when P0SERR.DIAG_X is cleared. |
| | | 0 | No change in Current Connect Status |
| | | 1 | Change in Current Connect Status |
| 5 | DPS | 0-1 | Descriptor Processed. A PRD with the I bit set has transferred all of its data.<br>Note: This is an opportunistic interrupt and should not be used to definitely indicate the end of a transfer. Two PRD interrupts could happen close in time together such that the second interrupt is missed when the first PRD interrupt is being cleared. |
| 4 | UFS | 0-1 | Unknown FIS Interrupt. When set to 1, indicates that an unknown FIS was received and has been copied into system memory. This bit is cleared to 0 by software clearing the P0SERR.DIAG_F bit to 0.<br>Note: The UFS bit does not directly reflect the P0SERR.DIAG_F bit. P0SERR.DIAG_F bit is set immediately when an unknown FIS is detected, whereas the UFS bit is set when that FIS is posted to memory. Software should wait to act on an unknown FIS until the UFS bit is set to 1 or the two bits may become out of sync. |
| 3 | SDBS | 0-1 | Set Device Bits Interrupt. A Set Device Bits FIS has been received with the I bit set and has been copied into system memory. |
| 2 | DSS | 0-1 | DMA Setup FIS Interrupt. A DMA Setup FIS has been received with the I bit set and has been copied into system memory. |
| 1 | PSS | 0-1 | PIO Setup FIS Interrupt. A PIO Setup FIS has been received with the I bit set, it has been copied into system memory, and the data related to that FIS has been transferred. |
| 0 | DHRS | 0-1 | Device to Host Register FIS Interrupt. A D2H Register FIS has been received with the I bit set, and has been copied into system memory. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.22 Port Interrupt Enable Register (P0IE)

The port interrupt enable register (P0IE) enables and disables the reporting of the corresponding interrupt to system software. When a bit is set (1), and the corresponding interrupt condition is active, then the SATASS intrq output is asserted. Interrupt sources that are disabled (0) are still reflected in the status registers. This register is symmetrical with the P0IS register. This register is reset on Global reset.

The P0IE register is shown in Figure 21-23 and described in Table 21-24.

**Figure 21-23. Port Interrupt Enable Register (P0IE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Rsvd | TFEE | HBFE | HBDE | IFE | INFE | Rsvd | OFE | IPME | PRCE | Reserved | |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | |

| 15 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | DMPE | PCE | DPE | UFE | SDBE | DSE | PSE | DHRE |
| R-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-24. Port Interrupt Enable Register (P0IE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | 0 | Reserved. No CPD capability exist since CPD related pins are not bonded out for the device. If need this feature, use of GPIO is recommended. |
| 30 | TFEE | 0-1 | Task File Error Enable. When set to 1, GHC.IE = 1, and P0IS.HBFS = 1, the intrq output is asserted. |
| 29 | HBFE | 0-1 | Host Bus Fatal Error Enable. When set to 1, GHC.IE = 1, and P0IS.HBFS = 1, the interq output is asserted. |
| 28 | HBDE | 0-1 | Host Bus Data Error Enable. When set to 1, GHC.IE = 1, and P0IS.HBDS = 1, the intrq output is asserted. |
| 27 | IFE | 0-1 | Interface Fatal Error Enable. When set to 1, GHC.IE = 1, and P0IS.IFS = 1, the intrq output is asserted. |
| 26 | INFE | 0-1 | Interface Non-fatal Error Enable. When set to 1, GHC.IE = 1, and P0IS.INFS = 1, the intrq output is asserted. |
| 25 | Reserved | 0 | Reserved. |
| 24 | OFE | 0-1 | Overflow Enable. When set to 1, GHC.IE = 1, and P0IS.OFS = 1, the intrq output is asserted. |
| 23 | IPME | 0-1 | Incorrect Port Multiplier Enable. When set to 1, GHC.IE = 1, and P0IS.IPMS = 1, the intrq output is asserted. |
| 22 | PRCE | 0-1 | PHY Ready Change Enable. When set to 1, GHC.IE = 1, and P0IS.PRCS = 1, the intrq output is asserted. |
| 21-8 | Reserved | 0 | Reserved. |
| 7 | DMPE | 0-1 | Device Mechanical Presence Enable. When set to 1, GHC.IE = 1, and P0IS.DMPS = 1, the intrq output is asserted. |
| 6 | PCE | 0-1 | Port Change Interrupt Enable. When set to 1, GHC.IE = 1, and P0IS.PCS = 1, the intrq output is asserted. |
| 5 | DPE | 0-1 | Descriptor Processed Interrupt Enable. When set to 1, GHC.IE = 1 and P0IS.DPS = 1, the intrq output is asserted. |
| 4 | UFE | 0-1 | Unknown FIS Interrupt Enable. When set to 1, GHC.IE = 1, and P0IS.UFS = 1, the intrq output is asserted. |
| 3 | SDBE | 0-1 | Set Device Bits FIS Interrupt Enable. When set to 1, GHC.IE = 1, and P0IS.SDBS = 1, the intrq output is asserted. |
| 2 | DSE | 0-1 | DMA Setup FIS Interrupt Enable. When set to 1, GHC.IE = 1, and P0IS.PSS = 1, the intrq output is asserted. |
| 1 | PSE | 0-1 | PIO Setup FIS Interrupt Enable. When set to 1, GHC.IE = 1, and P0IS.PSS = 1, the intrq output is asserted. |
| 0 | DHRE | 0-1 | Device Host Register FIS Interrupt Enable. When set to 1, GHC.IE = 1, and P0IS.DHRS = 1, the intrq output is asserted. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.23 Port Command Register (P0CMD)

The port command register (P0CMD) contains bits controlling various Port functions. All read/write bits are reset on Global reset.

The P0CMD register is shown in Figure 21-24 and described in Table 21-25.

**Figure 21-24. Port Command Register (P0CMD)**

| 31 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ICC | | ASP | ALPE | DLAE | ATAPI | Reserved | | ESP | Rsvd | MPSP | HPCP | PMA | CPD |
| R/W-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | | W/RO-0 | W/RO-0 | W/RO-0 | W/RO-0 | R/W-0 | R-0 |

| 15 | 14 | 13 | 12 | 8 | 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CR | FR | MPSS | CCS | | Reserved | | FRE | CLO | POD | SUD | ST |
| R-0 | R-0 | R-0 | R-0 | | R-0 | | R/W-0 | W-0 | R/W-0 | W/RO-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; W = Write; W/RO: written once after hard reset by firmware, then remain as read-only; *-n* = value after reset

**Table 21-25. Port Command Register (P0CMD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | ICC | 0-Fh | Interface Communication Control. Controls power management states of the interface. If the Link layer is currently in the L_IDLE state, writes to this bit field causes the Port to initiate a transition to the interface power management state requested. If the Link layer is not currently in the L_IDLE state, writes to this bit field have no effect.<br>When system software writes a nonreserved value other than No-Op (0), the Port performs the action and updates this bit field back to Idle (0). If software writes to this bit field to change the state to a state the link is already in (that is, interface is in the active state and a request is made to go to the active state), the Port takes no action and returns this bit field to Idle. If the interface is in a low power state and software wants to transition to a different low power state, software must first bring the link to active and then initiate the transition to the desired low power state. |
| | | 0 | No-Op/ Idle: When software reads this value, it indicates the Port is ready to accept a new interface control command, although the transition to the previously selected state may not yet have occurred. |
| | | 1h | Active: Causes the Port to request a transition of the interface into the active state. |
| | | 2h | Partial: Causes the Port to request a transition of the interface to the Partial state. The SATA device may reject the request and the interface remains in its current state. |
| | | 3h-5h | Reserved |
| | | 6h | Slumber: Causes the Port to request a transition of the interface to the Slumber state. The SATA device may reject the request and the interface remains in its current state. |
| | | 7h-Fh | Reserved |
| 27 | ASP | | Aggressive Slumber/Partial. |
| | | 0 | When cleared to 0 and P0CMD.ALPE = 1, the Port aggressively enters the PARTIAL state when it clears the P0CI register, and the P0SACT register is cleared when it clears the P0SACT register and P0CI is cleared. |
| | | 1 | When set to 1 and P0CMD.ALPE = 1, the Port aggressively enters the SLUBMER state when it clears the P0CI register and the P0SACT register is cleared, or when it clears the P0SACT register and P0CI is cleared. |
| 26 | ALPE | | Aggressive Link Power Management Enable. |
| | | 0 | Aggressive power management state transition is disabled. |
| | | 1 | Port aggressively enters a lower link power state (PARTIAL or SLUMBER) based on the setting of the P0CMD.ASP bit. |
| 25 | DLAE | 0-1 | Drive LED on ATAPI Enable. When set to 1, P0CMD.ATAPI = 1, and commands are active, the Port asserts P0_act_led output. |
| 24 | ATAPI | 0-1 | Device is ATAPI. When set to 1, the connected device is an ATAPI device. This bit is used by the Port to control whether or not to assert P0_act_led output when commands are active. |
| 23-22 | Reserved | 0 | Reserved. |
| 21 | ESP | | External SATA Port. The ESP bit is mutually exclusive with the P0CMD.HPCP bit. |
| | | 0 | Ports signal only connector is not externally accessible. |
| | | 1 | Ports signal only connector is externally accessible. When set to 1, CAP.SXS is also set to 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-25. Port Command Register (P0CMD) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 20 | Reserved | 0 | Reserved. Since no CPD pins are bonded out for the device, this bit should be written with a zero value. If need to implement CPD functionality, GPIO usage is recommended. |
| 19 | MPSP | | Mechanical Presence Switch Attached to Port. |
| | | 0 | Platform does not support a mechanical presence switch on this port. |
| | | 1 | Platform supports a mechanical presence switch attached to this port. When this bit is set to 1, P0CMD.HPCP should also be set to 1. |
| 18 | HPCP | | Hot Plug. |
| | | 0 | Ports signal and power connectors are not externally accessible. |
| | | 1 | Ports signal and power connectors are externally accessible via a joint signal-power connector for blindmate device hot plug. |
| 17 | PMA | | Port Multiplier Attached. |
| | | | Note: Software is responsible for detecting whether a Port Multiplier is present. There is no auto detection. |
| | | 0 | When cleared to 0 by software, indicates that a Port Multiplier is not attached to this Port. |
| | | 1 | When set to 1 by software, indicates that a Port Multiplier is attached to this Port. |
| 16 | CPD | 0 | Since no CPD pins are bonded out for the device, this bit is always zero. if need to implement CPD usage of GPIO is recommended. |
| 15 | CR | 0-1 | Command List Running. When this bit is set to 1, the command list DMA engine for this Port is running. See Synopsys spec for more detail. |
| 14 | FR | 0-1 | FIS Receive Running. When set to 1, the FIS Receive DMA engine for this port is running. Refer to the AHCI specification section 10.3.2 for details on when this bit is set and cleared by the Port. |
| 13 | MPSS | | Mechanical Presence Switch State. Reports the state of a mechanical presence switch attached to this port as indicated by the P0_mp_switch input state (assuming CAP.SMPS = 1 and P0CMD.MPSP = 1). If CAP.SMPS = 0, then this bit is cleared to 0. Software should only use this bit if both CAP.SMPS and P0CMD.MPSP are set to 1. |
| | | | Note: The reset of this bit may change based on the SATA_MP_SWITCH state at reset. |
| | | 0 | Switch is closed. |
| | | 1 | Switch is open. |
| 12-8 | CCS | 0-1Fh | Current Command Slot. This field is valid when P0CMD.ST is set to 1 and is set to the command slot value of the command that is currently being issued by the Port. When P0CMD.ST transitions from 1 to 0, this field is reset to 0x00. After P0CMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0. After the first command has been issued, the highest priority slot to issue from next is P0CMD.CCS+1. For example, after the Port has issued its first command, if CCS=0x00 and P0CI is set to 0x3, the next command that will be issued is from command slot 1. |
| 7-5 | Reserved | 0 | Reserved. |
| 4 | FRE | 0-1 | FIS Receive Enable. When set to 1, the Port may post received FISes into the FIS receive area pointed to by P0FB. When cleared, received FISes are not accepted by the Port, except for the first D2H register FIS after the initialization sequence, and no FISes are posted to the FIS receive area. |
| | | | System software must not set this but until P0FB has been programmed with a valid pointer to the FIS receive area, and if software wishes to move the base, this bit must first be cleared, and software must wait for the P0CMD.FR bit to be cleared. Refer to the Synopsis spec for important restrictions on changing P0CMD.FRE. |
| 3 | CLO | 1 | Command List Override. Setting this bit to 1 causes P0TFD.STS.BSY and P0TFD.STS.DRQ to be cleared to 0. This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the P0TFD.STS register. This bit is cleared to 0 when P0TFD.STS.BSY and P0TFD.STS.DRQ have been cleared to 0. A write to this register with a value of 0 has no effect. |
| | | | This bit should only be set to 1 immediately prior to setting P0CMD.ST bit to 1 from a previous value of 0. Setting this bit to 1 at any other time is not supported and results in indeterminate behavior. |
| 2 | POD | 0-1 | Power On Device. This bit is RW if cold presence detection is supported on this port as indicated by P0CMD.CPD=1. This bit is RO 1 if cold presence detection is not supported and P0CMD.CPD=0. Note: Since no CPD control/status related pins are bonded out on the device, this bit is read only. Usage of GPIO is recommended if need to supplement CPD feature. |
| 1 | SUD | 0-1 | Spin-Up Device. This bit is read/write if staggered spin-up is supported as indicated by the CAP.SSS = 1. This bit is read-only 1 if staggered spin-up is not supported and CAP.SSS = 0. On an edge detect from 0 to 1, the Port starts a COMRESET initialization sequence to the device. Clearing this bit causes no action on the interface. |
| | | | Note: the SUD bit is read-only 0 on power-up until CAP.SSS bit is written with the required value. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-25. Port Command Register (P0CMD) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | ST | 0-1 | Start. When set to 1, the Port processes the command list. When cleared, the Port does not process the command list. Whenever this bit is changed from a 0 to a 1, the Port starts processing the command list at entry 0. Whenever this bit is changed from a 1 to a 0, the P0CI register is cleared by the Port upon transition into an idle state. Refer to the AHCI specification for important restrictions on when this bit can be set to 1. |

### 21.4.24 Port Task File Data Register (P0TFD)

The port task file data register (P0TFD) contains Error and Status registers updated every time a new Register FIS.

The P0TFD register is shown in Figure 21-25 and described in Table 21-26.

**Figure 21-25. Port Task File Data Register (P0TFD)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| ERR | | STS | |
| R-0 | | R-7Fh | |

LEGEND: R = Read only; -*n* = value after reset

**Table 21-26. Port Task File Data Register (P0TFD) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-8 | ERR | 0-FFh | Error. Contains the latest copy of the task file error register. |
| 7-0 | STS | 0-FFh | Status. Contains the latest copy of the task file status register. |
| | | | Note: the HBA updates the entire 8-bit field not just the bits listed. |
| | | 0 | ERR – Indicates an error during the transfer |
| | | 1h-2h | Reserved |
| | | 3h | DRQ – Indicates a data transfer is requested |
| | | 4h-6h | Reserved |
| | | 7h | BSY – Indicates the interface is busy |
| | | 8h-FFh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.25 Port Signature Register (P0SIG)

The port signature register (P0SIG) contains the Device Signature information.

The P0SIG register is shown in Figure 21-26 and described in Table 21-27.

**Figure 21-26. Port Signature Register (P0SIG)**

| 31 | 0 |
|---|---|
| SIG | |
| R-FFFF FFFFh | |

LEGEND: R = Read only; -*n* = value after reset

**Table 21-27. Port Signature Register (P0SIG) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | SIG | 0-FFFF FFFFh | Signature. Contains the signature received from a device on the first D2H Register FIS. This register is updated once after a reset sequence. Reset on Global or Port reset. The bit order as is:<br>Bits 31-24: LBA High (Cylinder High) Register<br>Bits 23-16: LBA Mid (Cylinder Low) Register<br>Bits 15-8: LBA Low (Sector Number) Register<br>Bits 7-0: Sector Count Register |

### 21.4.26 Port Serial ATA Status Register (P0SSTS)

The port serial ATA status register (P0SSTS) conveys the current state of the interface and host. The Port updates it continuously and asynchronously. When the Port transmits a COMRESET to the device, this register is updated to its reset values (i.e., Global reset, Port reset, or COMINIT from the device).

The P0SSTS register is shown in Figure 21-27 and described in Table 21-28.

**Figure 21-27. Port Serial ATA Status Register (P0SSTS)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | IPM | | SPD | | DET | |
| R-0 | | R-0 | | R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 21-28. Port Serial ATA Status Register (P0SSTS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | Reserved | 0 | Reserved. |
| 11-8 | IPM | 0-Fh | Interface Power Management. Indicates the current interface state. |
| | | 0 | Device is not present or communication is not established. |
| | | 1h | Interface in active state |
| | | 2h | Interface in Partial power management state. |
| | | 3h-5h | Reserved |
| | | 6h | Interface in Slumber power management state. |
| | | 7h-Fh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-28. Port Serial ATA Status Register (P0SSTS) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7-4 | SPD | 0-Fh | Current Interface Speed. Indicates the negotiated interface communication speed. |
| | | 0 | Device is not present or communication is not established. |
| | | 1h | Generation 1 (1.5 Gbps) negotiated |
| | | 2h | Generation 2 (3 Gbps) negotiated |
| | | 3h-Fh | Reserved |
| 3-0 | DET | 0-Fh | Device Detection. Indicates the interface device detection and PHY state. |
| | | 0 | No device detected and PHY communication is not established. |
| | | 1h | Device presence detected but PHY communication is not established (COMINIT is detected). |
| | | 2h | Reserved |
| | | 3h | Device presence detected and PHY communication is established (PHY Ready is detected). |
| | | 4h | PHY in offline mode as a result of the interface being disabled or running in BIST loopback mode. |
| | | 5h-Fh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.27 Port Serial ATA Control Register (P0SCTL)

The port serial ATA control register (P0SCTL) is used by software to control SATA interface capabilities. Writes to this register result in an action being taken by the Port PHY interface. Reads from the register return the last value written to it. Reset on Global reset.

These bits are static and should not be changed frequently due to the clock crossing between the Transport and Link Layers. Software must wait for at least seven periods of the slower clock (clk_asic or OCP clock) before changing this register.

The P0SCTL register is shown in Figure 21-28 and described in Table 21-29.

**Figure 21-28. Port Serial ATA Control Register (P0SCTL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | IPM | | SPD | | DET | |
| R-0 | | R/W-0 | | R/W-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-29. Port Serial ATA Control Register (P0SCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | Reserved | 0 | Reserved. |
| 11-8 | IPM | 0-Fh | Interface Power Management Transitions Allowed. Indicates which power states the Port PHY interface is allowed to transition to. If an interface power management state is disabled, the Port does not initiate that state and any request from the device to enter that state is rejected via PMNAKp. |
| | | 0 | No interface power management state restrictions. |
| | | 1h | Transitions to the Partial state are disabled. |
| | | 2h | Transitions to the Slumber state are disabled. |
| | | 3h | Transitions to both Partial and Slumber states are disabled. |
| | | 4h-Fh | Reserved |
| 7-4 | SPD | 0-Fh | Speed Allowed. Indicates the highest allowable speed of the Port PHY interface.<br>Note: When host software must change this bit field value, the host must also reset the Port (DET = 1) at the same time to ensure proper speed negotiation. |
| | | 0 | No speed negotiation restrictions. |
| | | 1h | Limit speed negotiation to Generation 1 (1.5 Gbps) communication rate. |
| | | 2h | Limit speed negotiation to Generation 2 (3 Gbps) communication rate. |
| | | 3h-Fh | Reserved |
| 3-0 | DET | 0-Fh | Device Detection Initialization. Controls the Ports device detection and interface initialization.<br>Note: This bit field may only be modified when P0CMD.ST = 0; changing this bit field while P0CMD.ST = 1 results in undefined behavior. When P0CMD.ST is set to 1, this bit field should have a value of 0. |
| | | 0 | No device detection or initialization action is requested. |
| | | 1h | Perform interface initialization sequence to establish communication. This results in the interface being reset and communication re-initialized. |
| | | 2h-3h | Reserved |
| | | 4h | Disable the Serial ATA interface and put the Port PHY in offline mode. |
| | | 5h-Fh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.28 Port Serial ATA Error Register (P0SERR)

The port serial ATA error register (P0SERR) represents all the detected interface errors accumulated since the last time it was cleared. The set bits in the P0SERR register indicate that the corresponding error condition became true one or more times since the last time the bit was cleared. The set bits in this register are explicitly cleared by a write operation to the register, Global reset, or Port reset (COMRESET). The value written to clear the set error bits should have ones encoded in the bit positions corresponding to the bits that are to be cleared. All bits in the following table have a reset value of 0.

The P0SERR register is shown in Figure 21-29 and described in Table 21-30.

**Figure 21-29. Port Serial ATA Error Register (P0SERR)**

| 31 | | | | | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | DIAG_X | DIAG_F | DIAG_T |
| R-0 | | | | | | R/W1C-0 | R/W1C-0 | R/W1C-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | |
| DIAG_S | DIAG_H | DIAG_C | DIAG_D | DIAG_B | DIAG_W | DIAG_I | DIAG_N | |
| R/W1C-0 | R/W1C-0 | R/W1C-0 | R-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | |
| 15 | | | 12 | 11 | 10 | 9 | 8 | |
| Reserved | | | | ERR_E | ERR_P | ERR_C | ERR_T | |
| R-0 | | | | R/W1C-0 | R/W1C-0 | R/W1C-0 | R/W1C-0 | |
| 7 | | | | | 2 | 1 | 0 | |
| Reserved | | | | | | ERR_M | ERR_I | |
| R-0 | | | | | | R/W1C-0 | R/W1C-0 | |

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear; -*n* = value after reset

**Table 21-30. Port Serial ATA Error Register (P0SERR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-27 | Reserved | 0 | Reserved. |
| 26 | DIAG_X | 0-1 | Exchanged. This bit is set to 1 when PHY COMINIT signal is detected. This bit is reflected in the P0IS.PCS bit. |
| 25 | DIAG_F | 0-1 | Unknown FIS Type. Indicates that one or more FISes were received by the Transport layer with good CRC, but had a type field that was not recognized/known and the length was less than or equal to 64 bytes.<br>Note: If the Unknown FIS length exceeds 64 bytes, the DIAG_F bit is not set and the DIAG_T bit is set instead. |
| 24 | DIAG_T | 0-1 | Transport State Transition Error. Indicates that a Transport Layer protocol violation was detected. |
| 23 | DIAG_S | 0-1 | Link Sequence Error. Indicates that one or more Link state machine error conditions was encountered. One of the conditions that cause this bit to be set is the device doing SYNC escape during FIS transmission. |
| 22 | DIAG_H | 0-1 | Handshake Error. Indicates that one or more R-ERRp was received in response to frame transmission. Such errors may be the result of a CRC error detected by the device, a disparity or 8-bit/10-bit decoding error, or other error condition leading to a negative handshake on a transmitted frame. |
| 21 | DIAG_C | 0-1 | CRC Error. Indicates that one ore more CRC errors were detected by the Link layer during FIS reception. |
| 20 | DIAG_D | 0 | Disparity Error. This bit is always 0 since it is not used by the AHCI specification. |
| 19 | DIAG_B | 0-1 | 10B to 8B Decode Error. Indicates errors were detected by 10b8b decoder.<br>Note: This bit is set only when an error is detected on the received FIS data word. This bit is not set when an error is detected on the primitive, regardless of whether it is inside or outside the FIS. |
| 18 | DIAG_W | 0-1 | Comm Wake. This bit is set when the PHY COMWAKE signal is detected. |
| 17 | DIAG_I | 0-1 | PHY Internal Error. This bit is set when the PHY detects some internal error.<br>Note: The TI PHY does not support any errors so this bit will never be set. |
| 16 | DIAG_N | 0-1 | PHY Ready Change. Indicates that the PHY Ready signal changed state. This bit is reflected in the P0IS.PRCS bit. |
| 15-12 | Reserved | 0 | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-30. Port Serial ATA Error Register (P0SERR) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 11 | ERR_E | 0-1 | Internal Error. This bit is set to 1 when one or more AHB bus ERROR responses are detected on the master or the slave interfaces. |
| 10 | ERR_P | 0-1 | Protocol Error. This bit is set to 1 when any of the following conditions are detected:<br>• Transport state transition error (DIAG_T)<br>• Link sequence error (DIAG_S)<br>• RxFIFO overflow<br>• Link bad end error (WTRM instead of EOF is received) |
| 9 | ERR_C | 0-1 | Non-recovered Persistent Communication Error. This bit is set to 1 when the PHY Ready signal is negated due to the loss of communication with the device or problems with the interface, but not after transition from active to Partial or Slumber power management state. |
| 8 | ERR_T | 0-1 | Non-recovered Transient Data Integrity Error. This bit is set if any of the following P0SERR register bits is set during Data FIS transfer:<br>• ERR_P (Protocol)<br>• DIAG_C (CRC)<br>• DIAG_H (Handshake)<br>• ERR_C (PHY Ready negation) |
| 7-2 | Reserved | 0 | Reserved. |
| 1 | ERR_M | 0-1 | Recovered Communication Error. This bit is set to 1 when the PHY Ready condition is detected after interface initialization, but not after transition from partial or Slumber power management state to active state. |
| 0 | ERR_I | 0-1 | Recovered Data Integrity. This bit is set if any of the following P0SERR register bits is set during non-Data FIS transfer:<br>• ERR_P (Protocol)<br>• DIAG_C (CRC)<br>• DIAG_H (Handshake)<br>• ERR_C (PHY Ready negation) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.29 Port Serial ATA Active Register (P0SACT)

The port serial ATA active register (P0SACT) is used to indicate what command slots have commands in them.

The P0SACT register is shown in Figure 21-30 and described in Table 21-31.

**Figure 21-30. Port Serial ATA Active Register (P0SACT)**

| 31 | 0 |
|---|---|
| DS[*n*] | |
| R/W1S-0 | |

LEGEND: R/W = Read/Write; W1S = Write 1 to set; -*n* = value after reset

**Table 21-31. Port Serial ATA Active Register (P0SACT) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | DS[*n*] | 0-1 | This field is bit significant. Each bit *n* corresponds to the TAG and command slot of a native queued command, where bit 0 corresponds to TAG 0 and command slot 0. This field is set by software prior to issuing a native queued command for a particular command slot. Prior to writing P0CI[TAG] to 1, software will set DS[TAG] to 1 to indicate that a command with that TAG is outstanding. The device clears bits in this field by sending a Set Device Bits FIS to the Port. The Port clears bits in this field that are set to 1 in the SActive field of the Set Device Bits FIS. The Port only clears bits that correspond to native queued commands that have completed successfully.<br>Software should only write to this bit field when P0CMD.ST bit is set to 1. This bit field is cleared when P0CMD.ST is written from a 1 to a 0 by software. This bit field is not cleared by a Port reset (COMRESET) or a software reset. |

### 21.4.30 Port Command Issue Register (P0CI)

The port command issue register (P0CI) is used to indicate what that a command has been constructed and may be carried out.

The P0CI register is shown in Figure 21-31 and described in Table 21-32.

**Figure 21-31. Port Command Issue Register (P0CI)**

| 31 | 0 |
|---|---|
| CI[*n*] | |
| R/W1S-0 | |

LEGEND: R/W = Read/Write; W1S = Write 1 to set; -*n* = value after reset

**Table 21-32. Port Command Issue Register (P0CI) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | CI[*n*] | 0-1 | Command Issued. This field is bit significant. Each bit *n* corresponds to a command slot, where bit 0 corresponds to command slot 0. This bit field is set by software to indicate to the Port that a command has been built in system memory for a command slot and may be sent to the device. When the Port receives a FIS that clears the BSY, DRQ, and ERR bits for the command, it clears the corresponding bit *n* in this register for that command slot. Bits in this field can only be set to 1 by software when P0CMD.ST is set to 1. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.31 Port Serial ATA Notification Register (P0SNTF)

The port serial ATA notification register (P0SNTF) is used to determine if asynchronous notification events have occurred for directly connected devices and devices connected to a Port Multiplier.

The P0SNTF register is shown in Figure 21-32 and described in Table 21-33.

**Figure 21-32. Port Serial ATA Notification Register (P0SNTF)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | PMN[*n*] | |
| R-0 | | R/W1C-0 | |

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear; -*n* = value after reset

**Table 21-33. Port Serial ATA Notification Register (P0SNTF) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Reserved. |
| 1-0 | PMN[*n*] | 0-1 | PM Notify. Indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the SATASS Port with the Notification bit set. Individual bits are cleared by software writing 1s to the corresponding bit *n* positions. This bit field is reset on Global reset, but it is not reset by Port reset (COMRESET) or software reset.<br>• PM Port 0 sets bit 0<br>• PM Port 1 sets bit 1<br>Note that two different types of ports exist. HBA Port (the device has the support for 2 HBA Ports) and Port Multiplier Port (PMP). A Port of a Port Multiplier (PMP) connects a single device to a PM. As many as 15 devices can be connected to a single HBA Port. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.32 Port DMA Control Register (P0DMACR)

The port DMA control register (P0DMACR) contains bits for controlling the Port DMA engine. The software can change the fields of this register only when P0CMD.ST=0. Power-up (system reset), Global reset, or Port reset (COMRESET) reset this register to the default value.

The P0DMACR register is shown in Figure 21-33 and described in Table 21-34.

**Figure 21-33. Port DMA Control Register (P0DMACR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 12 | 11 8 | 7 4 | 3 0 |
|---|---|---|---|
| RXABL | TXABL | RXTS | TXTS |
| R-0 | R/W-0 | R/W-4h | R/W-4h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-34. Port DMA Control Register (P0DMACR) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved. |
| 15-12 | RXABL | 0-Fh | Receive Burst Limit. Allows software to limit the OCP master write burst size. |
| | | | Note: SATASS master breaks the burst at 1Kbyte address boundaries regardless of the RXABL value. |
| | | 0 | Limit OCP burst size to 256 DWORDS |
| | | 1h | Limit OCP Burst size to 1 DWORD |
| | | 2h | Limit OCP Burst size to 2 DWORDs |
| | | 3h | Limit OCP Burst size to 4 DWORDs |
| | | 4h | Limit OCP Burst size to 8 DWORDs |
| | | 5h | Limit OCP Burst size to 16 DWORDs |
| | | 6h | Limit OCP Burst size to 32 DWORDs |
| | | 7h | Limit OCP Burst size to 64 DWORDs |
| | | 9h-Fh | Limit OCP Burst size to 256 DWORDS |
| | | | Note: The maximum sized burst that will be issued on the OCP is 64 bytes; for this reason higher programmed values than 0x7 will still result with 64 bytes burst. |
| | | | Note: This field is read-write when P0CMD.ST=0 and read-only when P0CMD.ST=1. |
| 11-8 | TXABL | 0-Fh | Transmit Burst Limit. This field allows software to limit the OCP master read burst size. This bit field is read/write when P0CMD.ST = 0 and read-only when P0CMD.ST = 1. |
| | | | Note: SATASS master breaks the burst at 1Kbyte address boundaries regardless of the TXABL value. |
| | | 0 | Limit OCP burst size to 256 DWORDS |
| | | 1h | Limit OCP Burst size to 1 DWORD |
| | | 2h | Limit OCP Burst size to 2 DWORDs |
| | | 3h | Limit OCP Burst size to 4 DWORDs |
| | | 4h | Limit OCP Burst size to 8 DWORDs |
| | | 5h | Limit OCP Burst size to 16 DWORDs |
| | | 6h | Limit OCP Burst size to 32 DWORDs |
| | | 7h | Limit OCP Burst size to 64 DWORDs |
| | | 9h-Fh | Limit OCP Burst size to 256 DWORDs |
| | | | Note: The maximum sized burst that will be issued on the OCP is 64 bytes; for this reason higher programmed values than 0x7 will still result with 64 bytes burst. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-34. Port DMA Control Register (P0DMACR) Field Description (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7-4 | RXTS | 0-Fh | Receive Transaction Size (RX_TRANSACTION_SIZE). This field defines the Port DMA transaction size in DWORDs for receive (system bus write, device read) operation. This bit field is read/write when P0CMD.ST = 0 and read-only when P0CMD.ST = 1. The maximum value of this bit field is determined by the Rx FIFO Depth (P0_RXFIFO_DEPTH). If software attempts to write a value exceeding this maximum value, the maximum value would be set instead. |
| | | 0 | 1 DWORD |
| | | 1h | 2 DWORDs |
| | | 2h | 4 DWORDs |
| | | 3h | 8 DWORDs |
| | | 4h | 16 DWORDS |
| | | 5h | 32 DWORDs |
| | | 6h | 64 DWORDs |
| | | 7h-Fh | Reserved |
| 3-0 | TXTS | 0-Fh | Transmit Transaction Size (TX_TRANSACTION_SIZE). This field defines the DMA transaction size in DWORDs for transmit (system bus read, device write) operation. This bit field is read/write when P0CMD.ST = 0 and read-only when P0CMD.ST = 1. The maximum value of this bit field is determined by the Tx FIFO Depth (P0_TXFIFO_DEPTH). If software attempts to write a value exceeding this maximum value, the maximum value would be set instead. |
| | | 0 | 1 DWORD |
| | | 1h | 2 DWORDs |
| | | 2h | 4 DWORDs |
| | | 3h | 8 DWORDs |
| | | 4h | 16 DWORDS |
| | | 5h | 32 DWORDs |
| | | 6h-Fh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.33 Idle Register (IDLE)

The Idle register (IDLE) is used to control the idle and standby modes.

The IDLE register is shown in Figure 21-34 and described in Table 21-35.

**Figure 21-34. Idle Register (IDLE)**

| 31 | 18 | 17 | 16 |
|---|---|---|---|
| Reserved | | OVERRIDE1 | OVERRIDE0 |
| R-0 | | R/W-0 | R/W-0 |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | STANDBYMODE | | IDLEMODE | |
| R-0 | | R/W-0 | | R/W-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 21-35. Idle Register (IDLE) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-18 | Reserved | 0 | Reserved |
| 17 | OVERRIDE1 | | Override for Clock Stopping. Normally the functional clock can only be stopped if the link is put into partial or slumber power mode. However, if there is no device attached (such as in a removable media situation) or the device has not been started the user may wish to stop the functional clocks but not be able to enter a low power state. In this case, software can set the OVERRIDE bit to 1, removing the requirement for a low power state.<br>**WARNING:** If there is a device attached, the OVERRIDE bit is used, and the functional clock is stopped when the link is not in a low power state it will ruin the link and cause undetermined behavior. A port reset or full SATASS reset may be required to recover. |
| | | 0 | Normal |
| | | 1 | Override |
| 16 | OVERRIDE0 | | Override for Clock Stopping. Normally the functional clock can only be stopped if the link is put into partial or slumber power mode. However, if there is no device attached (such as in a removable media situation) or the device has not been started the user may wish to stop the functional clocks but not be able to enter a low power state. In this case, software can set the OVERRIDE bit to 1, removing the requirement for a low power state.<br>**WARNING:** If there is a device attached, the OVERRIDE bit is used, and the functional clock is stopped when the link is not in a low power state it will ruin the link and cause undetermined behavior. A port reset or full SATASS reset may be required to recover. |
| | | 0 | Normal |
| | | 1 | Override |
| 15-4 | Reserved | 0 | Reserved |
| 3-2 | STANDBYMODE | | Gives direct control to the standby_mode control of the standby generic. |
| 1-0 | IDLEMODE | | Gives direct control to the idle_mode control of the idle generic.<br>**Note:** Only force idle mode is supported. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.34 PHY Configuration Receive 0 Register (CFGRX0)

The PHY configuration receive 0 register (CFGRX0) is used to program the phy CFGRX0 bus. It is described in the figure and table below.

**Figure 21-35. PHY Configuration Receive 0 Register (CFGRX0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| LOOPBACK | | RX_TRIM_ BYPASS | CDR_ELV_ IDLE_FIX | CDRAUX | | CALFILTERDEPTH | |
| R/W-0x0 | | R/W-0x0 | R/W-0x0 | R/W-0x0 | | R/W-0x0 | |

| 23 | 22 | 19 | 18 | 16 |
|---|---|---|---|---|
| ENOC | EQ | | CDR | |
| R/W-0x0 | R/W-0x0 | | R/W-0x0 | |

| 15 | 13 | 12 | 11 | 10 | 8 |
|---|---|---|---|---|---|
| LOS | | ALIGN | | TERM | |
| R/W-0x0 | | R/W-0x0 | | R/W-0x0 | |

| 7 | 6 | 5 | 4 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| INVPAIR | RATE | | BUSWIDTH | | ENRXLDO | ENRX |
| R/W-0x0 | R/W-0x0 | | R/W-0x0 | | R/W-0x0 | R/W-0x0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-36. PHY Configuration Receive 0 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | LOOPBACK | 10/11 | Internal Loopback with following options |
| | | | CFGRX_1[14] Loop back from TX (pre-driver digital output) to RX (input of RXDIG20). Digital loop back. |
| | | | CFGRX_1[14] Loop back from TX (pre-driver digital output) to RX (input of equalizer). Analog loop back. |
| | | 0 | Loopback disabled |
| | | 1 | Not supported |
| 29 | RX_TRIM_BYPASS | | For calibration --> Bypasses the trim bits generated from the calibration algorithm |
| | | 0 | Use the trim code generated by trx_dig |
| | | 1 | Bypass the trim code generated by trx_dig |
| 28 | CDR_ELV_IDLE_FIX | 0 | CDR ELV Idle fix is enabled |
| | | 1 | CDR ELV Idle fix is disabled |
| 27-26 | CDRAUX | | CDRAUX[1:0]: Clock/data recovery auxilliary. Along with the CDR field, configures the clock/data recovery algorithm. |
| 25-24 | CAL_FILTER_DEPTH | | For calibration --> Average engine can be programmed to have depth of 7,15, 31 |
| | | 00 | Depth of 7 samples |
| | | 01 | Depth of 15 samples |
| | | 10 | Depth of 31 samples |
| | | 11 | Depth of 7 samples |
| 23 | ENOC | | ENOC: Enable offset compensation. Enables samplers offset compensation. Can be written DYNAMIC |
| | | 0 | |
| | | 1 | |
| 22-19 | EQ | | Equalizer. Enables and configures the adaptive equalizer to compensate for loss in the transmission media |
| | | 0 | Equalizer enabled |
| | | 1 | Equalizer disabled |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-36. PHY Configuration Receive 0 Register Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 18-16 | CDR | | Clock/data recovery. Along with the CDRAUX field, configures the clock/data recovery algorithm. |
| | | 0 | |
| | | 1 | |
| 15-13 | LOS | | Enables loss of signal detection |
| | | 0000 | LOS disabled |
| | | 100 | LOS enabled, without CDR override control [CDR enable] |
| | | 110 | LOS enabled, with CDR override control [CDR disable] |
| | | 001 | LOS disabled |
| | | 01x | LOS disabled |
| | | 101 | LOS disabled |
| | | 11x | -LOS disabled |
| 12-11 | ALIGN | | Symbol alignment. Enables internal or external symbol alignment |
| | | 00 | Alignment disabled |
| | | 01 | Comma alignment enabled |
| | | 10 | Alignment Jog |
| | | 11 | Reserved |
| 10-8 | TERM | | Selects input termination options. External AC and DC coupled modes supported. |
| | | 000 | Reserved |
| | | 001 | Common point set to 0.8 VDDA. This configuration is for AC coupled systems |
| | | 010 | Reserved |
| | | 011 | Common point floating. This configuration is for DC coupled systems which require the common mode voltage to be determined by the transmitter only. |
| | | 100 | Common point set to VSSA. This configuration is for PCI- Express and USB3.0 |
| | | 101 | Common point set to 0.2 VDDA. (For SATA system). |
| | | 110 | Reserved |
| | | 111 | Common point floating, wide common mode range. |
| 7 | INVPAIR | | Inverts polarity of RXPi and RXNi. |
| | | 0 | Normal Polarity. RXP considered to be positive |
| | | 1 | Inverted Polarity. RXN considered to be positive |
| 6-5 | RATE | | Read-only and reflects the value being driven to the phy's RATE field. Controlled by hardware and determined by the speed at which the link is operating. |
| 4-2 | BUSWIDTH | | Selects the parallel interface width.This field must be configured for 10-bit width. |
| | | 000 | 10-bit operation |
| | | 001 | Reserved |
| | | 01x | Reserved |
| | | 1xx | Reserved |
| 1 | ENRXLDO | | Enables RXLDO when high. When ENRXLDO='0', RX Analog circuits will be powered down. RXLDO has to be enabled and allowed to settle prior to enabling the receiver with ENRX. |
| 0 | ENRX | | Read-only and reflects the ENRX oh the phy. Controlled by hardware and will be 1 when the sub-system is set to receive data. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.35 PHY Configuration Receive 1 Register (CFGRX1)

The PHY configuration receive 1 register (CFGRX1) is used to program the phy CFGRX1 bus. It is shown and described in the figure and table below.

**Figure 21-36. PHY Configuration Receive 1 Register (CFGRX1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| EQ_ICM_S2 | | EQ_ICM_S1 | | RESERVED | | | |
| R/W-0x0 | | R/W-0x0 | | R/W-0x0 | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EQ_I_STAGE2 | | | EQ_I_STAGEFB | | | EQ_I_STAGE1 | |
| R/W-0x0 | | | R/W-0x0 | | | R/W-0x0 | |
| 15 | 14 | 13 | 12 | 11 | 9 | 8 | |
| EQ_I_STAGE1 | ANALOG_ LOOPBACK | RXTRIM_ CALIB | RXTRIM_ BYPASS_CTRL | | | RXTRIM_BYPASS_BITS | |
| R/W-0x0 | R/W-0 | R/W-0 | R/W-0 | R/W-0x0 | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXTRIM_ BYPASS_BITS | BYPASS_ CALOUT_AVG | CTG_DIG_ RSVD1 | CTG_DIG_ RSVD0 | ENTEST | TESTPATT | | |
| R/W-0x0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0x0 | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-37. PHY Configuration Receive 1 Register (CFGRX1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | EQ_ICM_S2 | | Trims common mode pullup current in second equalizer stage |
| 29-28 | EQ_ICM_S1 | | Trims common mode pullup current in first equalizer stage |
| 27-24 | RESERVED | | Reserved for future use |
| 23-21 | EQ_I_STAGE2 | | Trims the current in the second stage of the equalizer |
| 20-18 | EQ_I_STAGEFB | | Trims the current in the feedback stage of the equalizer |
| 17-15 | EQ_I_STAGE1 | | Trims the current in the first stage of the equalizer |
| 14 | ANALOG_LOOP BACK | | Enables analog loop back |
| 13 | RXTRIM_CALIB | | RX trimming control to the calibration block |
| | | 0 | In functioanl mode |
| | | 1 | eFuse training mode |
| 12 | RXTRIM_ BYPASS_CTRL | 0 | res_nwell settings are controlled by fuse/calib settingsRXTRIM[4:0] |
| | | 1 | res_nwell settings are controlled by cfg_ctrl[11:7](CFGRX_1[11:7]) |
| 11-7 | RXTRIM_ BYPASS_BITS | | Bypass bits for RXTRIM[4:0] when CFGRX_1[12]=1 |
| | | 0 | |
| | | 1 | |
| 6 | BYPASS_CALOUT_AVG | | For calibration --> Bypasses the averaging filter for the calout signal inside digital |
| | | 0 | Use the output of the digital filter for CALOUT |
| | | 1 | Ignore the output of the digital filter and use the output from the bubble removal circuit for CALOUT |
| 5 | CTG_DIG_RSVD 1 | | Clock inversion for rpclk. In X2256, this was mapped to ctl_dig_rsvd[1]. This is a temporary mapping. |
| 4 | CTG_DIG_RSVD 0 | | Clock inversion for fclk. In X2256, this was mapped to ctl_dig_rsvd[0]. This is a temporary mapping. |
| 3 | ENTEST | | Read-only and reflects the value on the serdes_testpatt_p0_entestrx input. The value of the input is driven to the phy's ENTEST field |
| 2-0 | TESTPATT | | Read-only and reflects the value on the serdes_testpatt_p0_testpatt input. The value of the input is driven to the phy's TESTPATT field |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.36 PHY Configuration Receive 2 Register (CFGRX2)

The PHY configuration receive 2 register (CFGRX2) is used to program the phy CFGRX2 bus. It is shown and described in the figure and table below.

**Figure 21-37. PHY Configuration Receive 2 Register (CFGRX2)**

| 31–27 | 26–22 | 21–17 | 16 |
|---|---|---|---|
| SAMP_BYPASS_SAD1 | SAMP_BYPASS_SAT0 | SAMP_BYPASS_SAD0 | SAMP_BYPASS_SAT1 |
| R/W-0x00 | R/W-0x00 | R/W-0x00 | R/W-0x00 |

| 15–12 | 11–8 |
|---|---|
| SAMP_BYPASS_SAT1 | RESERVED |
| R/W-0x00 | R/W-0 |

| 7–0 |
|---|
| RESERVED |
| R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-38. PHY Configuration Receive 2 Register (CFGRX2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-27 | SAMP_BYPASS_SAD1 | | Bypass bits for SAD1[4:0] when CFG_CTRL[94](CFGRX_3[30]) = 1 |
| 26-22 | SAMP_BYPASS_SAT0 | | Bypass bits for SAT0[4:0] when CFG_CTRL[94](CFGRX_3[30]) = 1 |
| 21-17 | SAMP_BYPASS_SAD0 | | Bypass bits for SAD0[4:0] when CFG_CTRL[94](CFGRX_3[30]) = 1 |
| 16-12 | SAMP_BYPASS_SAT1 | | Bypass bits for SAT1[4:0] when CFG_CTRL[94](CFGRX_3[30]) = 1 |
| 11-0 | RESERVED | | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.37 PHY Configuration Receive 3 Register (CFGRX3)

The PHY configuration receive 3 register (CFGRX3) is used to program the phy CFGRX3 bus. It is shown and described in the figure and table below.

**Figure 21-38. PHY Configuration Receive 3 Register (CFGRX3)**

| 31 | 30 | 29 24 |
|---|---|---|
| AMUX_EYESCAN_REF | SAMP_OC_SEL | SAMP_ES_VREF_BYPASS_BITS |
| R/W-0 | R/W-0 | R/W-0x00 |

| 23 20 | 19 17 | 16 |
|---|---|---|
| SAMP_ESCM_RES | SAMP_ESCM_I | SAMP_3_VREF_2_ES |
| R/W-0x0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 12 8 |
|---|---|---|---|
| SAMP_2_VREF_2_ES | SAMP_1_VREF_2_ES | SAMP_0_VREF_2_ES | SAMP_IBIAS_Z |
| R/W-0 | R/W-0 | R/W-0 | R/W-0x00 |

| 7 | 6 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| SAMP_ES_VREF_BYPASS_CTRL | RESERVED | SAMP_EN_3_ODD | SAMP_EN_2_EOTR | SAMP_EN_1_EVEN | SAMP_EN_0_OETR |
| R/W-0 | R/W-0x0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-39. PHY Configuration Receive 3 Register (CFGRX3) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | AMUX_EYESCAN_REF | 1 | Connect the eyescan reference voltages to the test mux inputs (used with amuxsel_rx[3:0]) |
| 30 | SAMP_OC_SEL | 0 | Sampler offset correction is controlled via the TRXDIG SAT1, SAD0, SAT0, SAD1 |
| | | 1 | Sampler offset correction is controlled via the configuration register (CFG_CTRL[63:44])(CFGRX_2[31:12]) |
| 29-24 | SAMP_ES_VREF_BYPASS_BITS | | Adjusts the eyescan reference voltage level, which introduces linearly decreasing offset with code |
| 23-20 | SAMP_ESCM_RES | | Trims the pullup resistors in the eyescan common mode loop; roughly should track CFGRX_1[10:7] |
| 19-17 | SAMP_ESCM_I | | Trims the current in the eyescan common mode loop; roughly should track CFG_CTRL[23:21](CFGRX_1[23:21]) |
| 16 | SAMP_3_VREF_2_ES | | Sampler 3 VREF |
| | | 0 | Reference voltages are EQCM |
| | | 1 | Connect reference voltage of sampler 3 to eyescan |
| 15 | SAMP_2_VREF_2_ES | | Sampler 2 VREF |
| | | 0 | Reference voltages are EQCM |
| | | 1 | Connect reference voltage of sampler 2 to eyescan |
| 14 | SAMP_1_VREF_2_ES | | Sampler 1 VREF |
| | | 0 | Reference voltages are EQCM |
| | | 1 | Connect reference voltage of sampler 1 to eyescan |
| 13 | SAMP_0_VREF_2_ES | | Sampler 0 VREF |
| | | 0 | Reference voltages are EQCM |
| | | 1 | Connect reference voltage of sampler 0 to eyescan |
| 12-8 | SAMP_IBIAS_Z | | Sampler bias |
| | | | Adjusts the current mirror ratio in the sampler bias current block. Lower numbers correspond to higher bias current. |
| 7 | SAMP_ES_VREF_BYPASS_CTRL | 0 | Eyescan reference offset voltage controled through IEEE1500 |
| | | 1 | Eyescan reference offset voltage controled through CFGRX_3[29:24] |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-39. PHY Configuration Receive 3 Register (CFGRX3) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 6-4 | RESERVED | | Reserved |
| 3 | SAMP_EN_3_ODD | | Samapler 3 |
| | | 0 | |
| | | 1 | Enable sampler 3 (odd data) |
| 2 | SAMP_EN_2_EOTR | | Sampler 2 |
| | | 0 | |
| | | 1 | Enable sampler 2 (even to odd transition) |
| 1 | SAMP_EN_1_EVEN | | Sampler 1 |
| | | 0 | |
| | | 1 | Enable sampler 1 (even data) |
| 0 | SAMP_EN_0_OETR | | Sampler 0 |
| | | 0 | |
| | | 1 | Enable sampler 0 (odd to even transition) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.38 PHY Configuration Receive 4 Register (CFGRX4)

The PHY configuration receive 4 register (CFGRX4) is used to program the phy CFGRX4 bus. It is shown and described in the figure and table below.

**Figure 21-39. PHY Configuration Receive 4 Register (CFGRX4)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | RCLK_SAMP | | RCLK_DIG | | RESERVED | DCD_EN_BUF |
| R/W-0 | | R/W-0x0 | | R/W-0x0 | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DCD_EN_CLK | DCD_EN_M | DCD_EN_P | RESERVED | PI_VCM_Z_0 | PI_VCM_1 | PI_VCM_Z_2 | PI_VCM_Z_3 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PI_BIAS_DISABLE | PI_ED_CAL_LEVEL_DEC | PI_ED_CAL_LEVEL_INC | PI_ED_CAL | PI_ED_RESET | PI_ED_EN | PI_ED_VOUTP | PI_ED_VOUTM |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PI_I50U | PI_I100U_Z | PI_IBIAS_Z | | | | | |
| R/W-0 | R/W-0 | R/W-0x00 | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-40. PHY Configuration Receive 4 Register (CFGRX4) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | RESERVED | | UNUSED |
| 29-28 | RCLK_SAMP | | RCLK_SAMP[1:0] |
| 27-26 | RCLK_DIG | | RCLK_DIG[1:0] |
| 25 | RESERVED | | UNUSED |
| 24 | DCD_EN_BUF | | DCD_EN_BUF |
| 23 | DCD_EN_CLK | | DCD_EN_CLK |
| 22 | DCD_EN_M | | DCD_EN_M |
| 21 | DCD_EN_P | | DCD_EN_P |
| 20 | RESERVED | | UNUSED |
| 19 | PI_VCM_Z_0 | | PI_VCM_Z[0] |
| 18 | PI_VCM_1 | | PI_VCM[1] |
| 17 | PI_VCM_Z_2 | | PI_VCM_Z[2] |
| 16 | PI_VCM_Z_3 | | PI_VCM_Z[3] |
| 15 | PI_BIAS_DISABLE | | PI_BIAS_DISABLE |
| 14 | PI_ED_CAL_LEVEL_DEC | | PI_ED_CAL_LEVEL_DEC |
| 13 | PI_ED_CAL_LEVEL_INC | | PI_ED_CAL_LEVEL_INC |
| 12 | PI_ED_CAL | | PI_ED_CAL |
| 11 | PI_ED_RESET | | PI_ED_RESET |
| 10 | PI_ED_EN | | PI_ED_EN |
| 9 | PI_ED_VOUTP | | PI_ED_VOUTP |
| 8 | PI_ED_VOUTM | | PI_ED_VOUTM |
| 7 | PI_I50U | | PI_I50U |
| 6 | PI_I100U_Z | | PI_I100U_Z |
| 5-0 | PI_IBIAS_Z | | PI_IBIAS_Z[5:0] |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.39 Receive Bus PHY-to-Controller Status Register (Phy STSRX)

The receive bus PHY-to-controller status register (Phy STSRX) is used to observe the phy STSRX bus. It is shown and described in the figure and table below.

**Figure 21-40. Receive Bus PHY-to-Controller Status Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RX_RTRIM | | | | | RX_CALOUT_D | RX_CALOUT_A | RESERVED |
| R/W-0X0 | | | | | | R/W-0 | R-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | OCIP | BSRXN | BSRXP | LOSDTCT | ODDCG | SYNC | TESTFAIL |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-41. Receive Bus PHY-to-Controller Status Register Field Descriptions Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | | Reserved; driven low |
| 15-11 | RX_RTRIM | | Trim bits for RX resistor |
| 10 | RX_CALOUT_D | | CalOut after the digital filter |
| 9 | RX_CALOUT_A | | CalOut coming from analog |
| 8-7 | Reserved | | Reserved, driven low. |
| 6 | OCIP | | Offset Compensation in Progres. Driven high asynchronously during offset compensation |
| 5 | BSRXN | | Boundary scan data. Reflects the state of the comparator attached to RXNi when IEEE1149.6 boundary scan is enabled. Driven low otherwise. |
| 4 | BSRXP | | Boundary scan data. Reflects the state of the comparator attached to RXPi when IEEE1149.6 boundary scan is enabled. Driven low otherwise. |
| 3 | LOSDTCT | | Loss of Signal detect. Driven high asynchronously when a loss of signal (electrical idle) condition is detected for channel i. |
| 2 | ODDCG | | Odd code group. Low when SYNC is high. Thereafter, it toggles every cycle, so that it is high during odd numbered code groups (the code group output when SYNC is high is defined as code group 0). Synchronous to RXBCLKIN[i]. |
| 1 | SYNC | | Symbol alignment. When comma detection is enabled, this output is high when an aligned comma is received, in the same cycle that the comma pattern is output on RDi. Alternatively, when an alignment jog is requested, it is high to indicate that the request has been completed. Synchronous to RXBCLKIN[i]. |
| 0 | TESTFAIL | | Test failure. Driven high when an error is encountered during a test sequence executed on channel i. Synchronous to RXBCLKIN[i]. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.40 PHY Configuration Transmit 0 Register (CFGTX0)

The PHY configuration transmit 0 register (CFGTX0) is used to program the phy CFGTX0 bus. It is shown and described in the figure and table below.

**Figure 21-41. PHY Configuration Transmit 0 Register (CFGTX0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| LOOPBACK | | CALFILTERDEPTH | | RESERVED | DET_CTL | | ENIDL |
| R/W-0x0 | | R/W-0x0 | | R/W-0 | R/W-0x0 | | R/W-0 |

| 23 | 22 | 21 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|
| RESERVED | | TM_EXTRA_LOAD | | RESERVED | DEEMP | |
| R/W-0 | | R/W-0x0 | | R/W-0 | R/W-0x00 | |

| 15 | 13 | 12 | 9 | 8 |
|---|---|---|---|---|
| DEEMP | | SWING | | RESERVED |
| R/W-0x00 | | R/W-0x0 | | R/W-0 |

| 7 | 6 | 5 | 4 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| INVPAIR | RATE | | BUSWIDTH | | ENTXLDO | ENTX |
| R/W-0 | R/W-0x0 | | R/W-0x0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-42. PHY Configuration Transmit 0 Register (CFGTX0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | LOOPBACK | | Not Used. The RX has similar settings, hence this field is not used. |
| 29-28 | CAL_FILTER_DEPTH | | For calibration --> Average engine can be programmed to have depth of 7,15, 31. |
| | | 00 | Depth of 7 samples |
| | | 01 | Depth of 15 samples |
| | | 10 | Depth of 31 samples |
| | | 11 | Depth of 7 samples |
| 27 | RESERVED | | Reserved |
| 26-25 | DET_CTL | | Receiver detect control. Controls PCI-E receiver detect. |
| 24 | ENIDL | | Read-only and reflects the value being driven to the phy's ENIDL field. This bit is 0 during normal operation and 1 when idle is enabled by either partial mode, no valid data, or for OOB signaling. |
| 23-22 | RESERVED | | Reserved ; driven low |
| 21-19 | TM_EXTRA_LOAD | | TestMode. Default is 000 : auto program dummy load based on mode. |
| | | 0 | [19] -> '1': Enable different extraload resistors to make driver constant current at speed. Current level defined in CFGTX_2[26:25] |
| | | 1 | [20] -> '1': Enable static dummy load during electrical idle. Load current level defined in CFGTX_2[28:27] |
| | | | [21] -> '1': Enable switching dummy load during electrical idle. Load current level defined in CFGTX_2[30:29] |
| 18 | RESERVED | | Reserved- drive low |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-42. PHY Configuration Transmit 0 Register (CFGTX0) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 17-13 | DEEMP | | De-emphasis. Selects one of output de-emphasis settings from 0 to 50.2%<br>For PCIE<br>Value --> Amplitude Reduction (%) --> Amplitude Reduction (dB)<br>00000 --> 0 --> 0<br>00001 to 01111 --> 33.7 --> -3.57<br>10000 to 11111 --> 50.2 --> -6.05<br>For eSATA - FULL RATE (CFGTX_0[6:5]=00)<br>Value --> Amplitude Reduction (%) --> Amplitude Reduction (dB)<br>00000 --> 0 --> 0<br>00001 to 01111 --> 10 --> -1<br>10000 to 11111 --> 25 --> -2.5<br>For eSATA - HALF RATE (CFGTX_0[6:5]=01)<br>Value --> Amplitude Reduction (%) --> Amplitude Reduction (dB)<br>00000 --> 0 --> 0<br>00001 to 01111 --> 10 --> -1<br>10000 to 11111 --> RESERVED --> RESERVED |
| 12-9 | SWING | | TX Output swing selection. Selects one of output amplitude settings.<br>For PCIE<br>Value --> Amplitude (mVdfpp) --> ANA {swingsel,swing[1:0]}<br>0000 --> 148 (half) --> 111<br>0001 --> 148 (half) --> 111<br>0010 --> 258 (half) --> 110<br>0011 --> 258 (half) --> 110<br>0100 --> 295 (full/half) --> 101<br>0101 --> 295 (full/half) --> 101<br>0110 --> 516 (full/half) --> 100<br>0111 --> 516 (full/half) --> 100<br>1000 --> 774 (full/half) --> 001<br>1001 --> 774 (full/half) --> 001<br>1010 --> 1069 (full/half) --> 000<br>1011 --> 1069 (full/half) --> 000<br>1100 --> 1069 (full/half) --> 000<br>1101 --> 1069 (full/half) --> 000<br>1110 --> 1069 (full/half) --> 000<br>1111 --> 1069 (full/half) --> 000<br>For eSATA<br>Only 516mV (for Gen1) and 590mV (for Gen2) swing value is supported in SATA mode and all the codes above will map to this in TXA |
| 8 | RESERVED | | Reserved |
| 7 | INVPAIR | | Inverts polarity of TXPi and TXNi. |
| | | 0 | Normal Polarity. TXP is positive data |
| | | 1 | Inverted Polarity. TXN is positive data |
| 6-5 | RATE | | Read-only and reflects the value being driven to the phy's RATE field. Controlled by hardware and determined by the speed at which the link is operating. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-42. PHY Configuration Transmit 0 Register (CFGTX0) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 4-2 | BUSWIDTH | | Selects the parallel interface width (8 or 10 bit). |
| | | | Only 10-bit mode will be used. |
| | | 000 | 10-bit operation |
| | | | Any value other than 000 is reserved. |
| 1 | ENTXLDO | | Enables TX Analog LDO when high. When ENTXLDO='0', TX Analog circuits will be powered down. TXLDO has to be enabled and allowed to settle prior to enabling the transmitter with ENTX. |
| | | 0 | Disabled |
| | | 1 | Enabled |
| 0 | ENTX | | Read-only and reflects the value being driven to the ENTX of the phy. This bit is controlled by hardware. |
| | | 0 | Transmit domain is in reset or when in slumber mode |
| | | 1 | Transmitter enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.41 PHY Configuration Transmit 1 Register (CFGTX1)

The PHY configuration transmit 1 register (CFGTX1) is used to program the phy CFGTX1 bus. It is shown and described in the figure and table below.

**Figure 21-42. SERDES_TXCFG1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TRIM_HIGH_THRESHOLD | | | | TX_DISABLE_ON_THE_FLY | TX_FORCE_UPDATE | TX_TRIM_BYPASS | TRIM_2B_MODE |
| R/W-0X0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | TRIM_STEP_CHANGE | | TRIM_LOW_THRESHOLD | | CALIB_WAIT_CYCLES | | ENABLE_LOOP_DELAY |
| R/W-0x0 | | | R/W-0x0 | | R/W-0x0 | | R/W-0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BYPASS_CALOUT_AVG | NOLCKRST | ENBSPLS | ENBSRX | ENBSTX | BSINITCLK | BSINRXN | BSINRXP |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSYNC_ENABLE | ENTEST | BSTX | TESTPATT | | | USE_STAIRCASE | ENLFPS |
| R/W-0 | R/W-0 | R/W-0 | R/W-0x0 | | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-43. PHY Configuration Transmit 1 Register (CFGTX1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | TRIM_HIGH_THRESHOLD | | For calibration --> 4-b programmable threshold<br>Unless the difference between the previous and new trim code required exceeds 4-b threshold, trim bit will be updated in one shot. Otherwise the trim bits will be updated in steps determined by TRIM_STEP_CHANGE. |
| 27 | TX_DISABLE_ON_THE_FLY | | For calibration --> Disables the pattern match. Updates can only be made through electrical idle or TX_FORCE_UPDATE bit. |
| | | 0 | Enables pattern match |
| | | 1 | Disables pattern match |
| 26 | TX_FORCE_UPDATE | | For calibration --> Forces the update to be made to the Analog trim bits without looking for any specific data pattern or electrical idle. |
| | | 0 | Disables forced update. |
| | | 1 | Enables forced update<br>Can be written DYNAMIC. |
| 25 | TX_TRIM_BYPASS | | For calibration --> Bypasses the trim bits generated from the calibration algorithm |
| | | 0 | Use the trim code generated by trx_dig |
| | | 1 | Bypass the trim code generated by trx_dig |
| 24 | TRIM_2B_MODE | | For calibration --> Enables 2-b data pattern detection mode for trim updates. |
| | | 0 | 2-b detection mode, e.g. "00" or "11" |
| | | 1 | 3-b detection mode, e.g. "00" or "111" |
| 23 | RESERVED | | Reserved |
| 22-21 | TRIM_STEP_CHANGE | | For calibration --> Step change value for staircase logic |
| | | 00 | make updates in steps of 1 |
| | | 01 | make updates in steps of 2 |
| | | 10 | make updates in steps of 4 |
| | | 11 | make updates in steps of 8 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-43. PHY Configuration Transmit 1 Register (CFGTX1) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 20-19 | TRIM_LOW_THRESHOLD | | For calibration --> 2-b programmable threshold. |
| | | 00 | make updates only when diff is >= 1 |
| | | 01 | make updates only when diff is >= 2 |
| | | 10 | make updates only when diff is >= 3 |
| | | 11 | make updates only when diff is >= 4 |
| | | | Unless the difference between the previous and current trim code exceeds or equals above mentioned threshold, no updates trim bit updates are made in Analog. |
| 18-17 | CALIB_WAIT_CYCLES | | For calibration --> 2-b programmable number of wait cycles before polling the calout signal |
| | | 0 | Additional clock cycle delay of 1 MHz clock |
| | | 1 | Additional clock cycle delay of 1 MHz clock |
| | | 2 | Additional clock cycle delay of 1 MHz clock |
| | | 3 | Additional clock cycle delay of 1 MHz clock |
| 16 | ENABLE_LOOP_DELAY | | For calibration --> Enables the additional loop delay of 40 clock cycles before sweeping the trim code during the calibration |
| | | 0 | Disables the loop delay |
| | | 1 | Enables the loop delay, ensures at least 32-clock cycles between two full cycles of trim code sweep |
| 15 | BYPASS_CALOUT_AVG | | For calibration --> Bypasses the averaging filter for the callout signal inside digital |
| | | 0 | Use the output of the digital filter for CALOUT |
| | | 1 | Ignore the output of the digital filter and use the output from the bubble removal circuit for CALOUT. |
| 14 | Reserved | | Reserved; driven low. |
| 13 | ENBSPLS | | This field is read only and reflects the value on the serdes_bscan_ctrl_p0_enbspt input. The value of the input is what is driven to the phy's ENBSPT field |
| 12 | ENBSRX | | This field is read only and reflects the value on the serdes_bscan_ctrl_p0_enbsrx input. The value of the input is what is driven to the phy's ENBSRX field. |
| 11 | ENBSTX | | This field is read only and reflects the value on the serdes_bscan_ctrl_p0_enbstx input. The value of the input is what is driven to the phy's ENBSTX field. |
| 10 | BSINITCLK | | This field is read only and reflects the value on the serdes_bscan_clk_p0_bsinitclk input. The value of the input is what is driven to the phy's BSINITCLK field. |
| 9 | BSINRXN | | This field is read only and reflects the value on the serdes_bscan_p0_bsinrxn input. The value of the input is what is driven to the phy's BSINRXN field. |
| 8 | BSINRXP | | This field is read only and reflects the value on the serdes_bscan_p0_bsinrxp input. The value of the input is what is driven to the phy's BSINRXP field. |
| 7 | TSYNC_ENABLE | | In x2256: was mapped to CFGTX_0 [21] |
| | | | In x2282: DO NOT USE |
| 6 | ENTEST | | This field is read only and reflects the value on the serdes_testpatt_p0_entesttx input. The value of the input is what is driven to the phy's ENTEST field. |
| 5 | BSTX | | This field is read only and reflects the value on the serdes_bscan_p0_bstx input. The value of the input is what is driven to the phy's BSTX field. |
| 4-2 | TESTPATT | | Not Used. The RX has similar settings, hence this field is not used. |
| 1 | USE_STAIRCASE | | For calibration |
| | | 0 | Disables staircase logic for trim code change |
| | | 1 | Enables staircase logic for trim code change |
| 0 | ENLFPS | | DO NOT USE: Drive Low |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.42 PHY Configuration Transmit 2 Register (CFGTX2)

The PHY configuration transmit 2 register (CFGTX2) is used to program the phy CFGTX2 bus. It is shown and described in the figure and table below.

**Figure 21-43. PHY Configuration Transmit 2 Register (CFGTX2)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | TM_DUMMY2IDLE | | TM_DUMMY1IDLE | | TM_TEXTRALOAD | | DISABLE_FB2_UNUSED |
| R/W-0 | R/W-0X0 | | R/W-0x0 | | R/W-0x0 | | R/W-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DISABLE_FB1_TM_SRCONTROL | SC_SATA | | SC_PCIE | | DISABLE_PCIE_SC | DISABLE_SATA_SC | Reserved |
| R/W-0 | R/W-0x0 | | R/W-0x0 | | R/W-0 | R/W-0 | R/W-0 |
| | | | | | | | TMTRIM |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | TXDCC_PWRDN | Reserved | | | | | TMTRIM |
| R/W-0 | R/W-0 | R/W-0x00 | | | | | R/W-0x00 |
| 7 | | | 4 | 3 | 2 | 1 | 0 |
| TMTRIM | | | | TRIMBYPASS | RDTCT_VTMODE | | RESERVED |
| R/W-0x00 | | | | R/W-0x0 | R/W-0x0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-44. PHY Configuration Transmit 2 Register (CFGTX2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | Reserved | | Reserved |
| 30-29 | TM_DUMMY2IDLE | | TestMode.Default is 00. If tm_dummy2_enz=1, then this controls ldo dummy digital load. |
| | | 00 | no resistive dummy load |
| | | 01 | 1.5 mA dummy load |
| | | 10 | 1.5 mA dummy load |
| | | 11 | 3mA dummy load |
| | | | Other settings: Extra load turned on |
| 28-27 | TM_DUMMY1IDLE | | TestMode.Default is 00. If tm_dummy1_enz=1, then this controls ldo dummy resistor load. |
| | | 00 | no resistive dummy load |
| | | 01 | 6 mA resistive dummy load |
| | | 10 | 9.6 mA resistive dummy load |
| | | 11 | 15mA resistive dummy load |
| | | | Other settings: Extra load turned on |
| 26-25 | TM_TEXTRALOAD | | TestMode: default is 00. If txa_tmel_enz=1, the driver dummy current controlled by tx_extraload. |
| | | 00 | Extra load current turned off |
| | | 01 | 1mA dummy load |
| | | 10 | 2 mA dummy load |
| | | 11 | 3 mA dummy load |
| 24 | DISABLE_FB2_UNUSED | | Reserved |
| 23 | DISABLE_FB1_TM_SRCONTROL | | TM_SRCONTROL |
| 22-21 | SC_SATA | | TestMode: Control for SATA slew-control. Default value is 00. |
| 20-19 | SC_PCIE | | TestMode: Control for PCIe slew-control. Default value is 00 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-44. PHY Configuration Transmit 2 Register (CFGTX2) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 18 | DISABLE_PCIE_SC | | TestMode: Default value for PCIE/SATA mode=0. |
| | | 0 | Enable output divider slew-rate control for PCIe mode (default) |
| | | 1 | Disable slew-rate control from digital and instead force from SC_PCIE |
| 17 | DISABLE_SATA_SC | | TestMode. Automatically disabled in pcie mode. Default value for PCIE/SATA mode=0. |
| | | 0 | Enable output driver slew-rate control for SATA mode (default). Slew rate automatically adjusted based on RATE in SATA mode. |
| | | 1 | Disable slew-rate control from digital and instead force from SC_SATA |
| 16 | RESERVED | | Boundary scan initialization for RXN. The IEEE1149.6 boundary scan comparator attached to RXN is initialized to the value of this bit on the rising edge of BSINITCLK |
| 15 | RESERVED | | Boundary scan initialization for RXP. The IEEE1149.6 boundary scan comparator attached to RXP is initialized to the value of this bit on the rising edge of BSINITCLK |
| 14 | TXDCC_PWRDN | | TestMode. Default is 0. |
| | | 0 | TX clock will be duty-cycle corrected (default) |
| | | 1 | TX clock will bypass duty-cycle correction logic |
| 13-9 | RESERVED | | Reserved. Default is 00000 |
| 8-4 | TMTRIM | | TestMode: Termination resistor calibration code for grounded resistor. Default is 00000 |
| 3 | TRIMBYPASS | | TestMode: Default is 0. |
| | | 0 | TX Termination resistor calibrated through continuous digital loop (default) |
| | | 1 | Over-ride digital control of termination resistor with TMTRIM |
| 2-1 | RDTCT_VTMODE | | Receive detect test mode to program threshold. Default is 00. |
| | | 00 | .285V threshold |
| | | 01 | 0.3V threshold |
| | | 10 | .27V threshold |
| | | 11 | 0.35V (VBG_REF) |
| 0 | RESERVED | | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.43 PHY Configuration Transmit 3 Register (CFGTX3)

The PHY configuration transmit 3 register (CFGTX3) is shown and described in the figure and table below.

**Figure 21-44. PHY Configuration Transmit 3 Register (CFGTX3)**

| 31 | 28 | 27 | 16 |
|---|---|---|---|
| RESERVED | | RXLDO_CTRL | |
| R/W-0x0 | | R/W-0x0000 | |

| 15 | 12 | 11 | 10 | 1 | 0 |
|---|---|---|---|---|---|
| RXLDO_CTRL | | RXLDO_HITRAN_UNUSED | TXLDO_CTRL | | TXLDO_DISABLE_OVERSHOOT |
| R/W-0x0000 | | R/W-0 | R/W-0x000 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-45. PHY Configuration Transmit 2 Register (CFGTX3) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-28 | RESERVED | | Reserved |
| 27-12 | RXLDO_CTRL | | TestMode: RX LDO control bits<br>[27:24] = 0111 (LDO loop compensation control)<br>[23] = 0 (When '1', overrides LDO UP signal to be always high)<br>[22] = 0 (Low dropout mode)<br>[21] = 1 ( Vref magnitude selection)<br>[20] = 1 ( Iref magnitude selection)<br>[19:18]= 11 (Quiescent current programmability)<br>[17] = 0 (disable overshoot control block)<br>[16:12]=01110 (default trim setting to get 1.2V) |
| 11 | RXLDO_HITRAN_UNUSED | | TestMode: RX LDO transient improvement bit |
| 10-1 | TXLDO_CTRL | | TestMode:TX LDO control bits<br>[10:7] = 0111 (LDO loop compensation control)<br>[6] = 0 (When '1', overrides LDO UP signal to be always high)<br>[5] = 0 (Low dropout mode)<br>[4] = 1 ( Vref magnitude selection)<br>[3] = 1 ( Iref magnitude selection)<br>[2:1]= 11 (Quiescent current programmability) |
| 0 | TXLDO_DISABLE_OVERSHOOT | | TestMode: TX LDO disable overshoot control if 1. Default is 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.4.44 PHY Configuration Transmit 4 Register (CFGTX4)

The PHY configuration transmit 4 register (CFGTX4) is shown and described in the figure and table below.

**Figure 21-45. PHY Configuration Transmit 4 Register (CFGTX4)**

| 31 | 16 |
|---|---|
| Reserved | |
| R/W-0x000 0000 | |

| 15 | 5 | 4 | 3 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | TM_BIASCTR | TRIM_MODE | | HYST_DISABLE |
| R/W-0x000 0000 | | R/W-0 | R/W-0X0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-46. PHY Configuration Transmit 2 Register (CFGTX4) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-5 | Reserved | | Reserved |
| 4 | TM_BIASCTR | | TM_BIASCTR used in DCC |
| 3-1 | TRIM_MODE | | TRIM_MODE[2:0] |
| 0 | HYST_DISABLE | | HYST_DISABLE_Z and CALOUT_TX |

### 21.4.45 Transmit Bus Controller-to-PHY Status Register

The transmit bus controller-to-PHY status register is shown and described in the figure and table below.

**Figure 21-46. Transmit Bus Controller-to-PHY Status Register**

| 31 | 10 |
|---|---|
| Reserved | |
| R/W-0 | |

| 9 | 8 | 7 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| TX_CALOUT_D | TX_CALOUT_A | TX_RTRIM | | Reserved | RDTCTIP | TESTFAIL |
| R/W-0 | R/W-0 | R/W-0x00 | | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-47. Transmit Bus Controller-to-PHY Status Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | | Reserved; driven low. |
| 9 | TX_CALOUT_D | | CalOut after the digital filter |
| 8 | TX_CALOUT_A | | CalOut coming from analog |
| 7-3 | TX_RTRIM | | Trim bits for TX resistor |
| 2 | Reserved | | Reserved; driven low. |
| 1 | RDTCTIP | | Receiver detect in progress. Driven high synchronous to TXBCLKIN[i] when PCI-E receiver detection starts for channel i, and deasserted synchronous to TXBCLKIN[i] when the TXPi and TXNi common mode returns to normal (or when receiver detection disabled). |
| 0 | TESTFAIL | | Test failure. Driven high when an error is encountered during a test sequence executed on channel i, or if an over or underflow occurs in the Tx serializer when ENFTP = 0. Synchronous to TXBCLKIN[i]. |


Copyright © 2011–2016, Texas Instruments Incorporated

## 21.5 SATA PLL Registers

The PLL that clocks the SERDES is controlled through a register space that is different from the SATA Controller and the PHY. Dedicated registers that pertain to SATA exist within the control module space.

The Base address for the SATA PLL configuration registers is 4814_0000h.

**Table 21-48. SATA PLL Controller Registers**

| Address Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 720h | SATA_PLLCFG0 | SATA SerDes PLL Configuration Register 0 | Section 21.5.1 |
| 724h | SATA_PLLCFG1 | SATA SerDes PLL Configuration Register 1 | Section 21.5.2 |
| 728h | SATA_PLLCFG2 | SATA SerDes PLL Configuration Register 2 | Section 21.5.3 |
| 72Ch | SATA_PLLCFG3 | SATA SerDes PLL Configuration Register 3 | Section 21.5.4 |
| 730h | SATA_PLLCFG4 | SATA SerDes PLL Configuration Register 4 | Section 21.5.5 |
| 734h | SATA_PLLSTATUS | SATA SerDes PLL Status Register | Section 21.5.6 |
| 738h | SATA_RXSTATUS | SATA SerDes Receive Status Register | Section 21.5.7 |
| 73Ch | SATA_TXSTATUS | SATA SerDes Transmit Status Register | Section 21.5.8 |
| 740h | SATA_TESTCFG | SATA SerDes Test Config Register | Section 21.5.9 |

### 21.5.1 SATA PLL Configuration 0 Register (SATA_PLLCFG0)

The register access is determined by the state of the lock bit. When the lock bit is 0 and the module is in NonSec Priv mode, bit access is Read/Write as shown in Figure 21-47. Otherwise, bit access is read only. When the lock bit is set to 1, the register is not accessible.

The SATA_PLLCFG0 register is shown in Figure 21-47 and described in Table 21-49.

**Figure 21-47. SATA PLL Configuration 0 Register (SATA_PLLCFG0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SEL_IN_FREQ | DIGCLRZ | Reserved | | AMUXSEL | TESTCLKMUXSEL | APLL_MISC_CTRL | |
| R/W-1 | R/W-1 | R-0 | | R/W-0 | R/W-0 | R/W-0x00 | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| APLL_MISC_CTRL | | | | DIS_REFCLK | EN_3P | PFD_CLR | CLK_FLIP |
| R/W-0x00 | | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EN_RTRIM | EN_MEAS | EN_LATCH | | CP_CTRL | | | |
| R/W-0 | R/W-0 | R/W-0x0 | | R/W-0x0 | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESVALUE | | C1_2X | ENDIGLDO | SELSC | ENBGSC_REF | ENPLLDO | ENPLL |
| R/W-0x0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-49. SATA PLL Configuration 0 Register (SATA_PLLCFG0) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | SEL_IN_FREQ | | Select input frequency. |
| | | 0 | 100M |
| | | 1 | 20M |
| 30 | DIGCLRZ | | CLRZ for APLL DIG and DLL DIG. Used to reset flops and state machines. |
| 29-28 | Reserved | | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-49. SATA PLL Configuration 0 Register (SATA_PLLCFG0) Field Description (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 27 | AMUXSEL | | AMUX select |
| | | 0 | KVCO |
| | | 1 | VRTRIM |
| 26 | TESTCLKMUXSEL | | Selects test clock mux |
| | | 0 | REF Clock |
| | | 1 | Divided clock |
| 25-20 | APLL_MISC_CTRL | | Used by APLL DIG. |
| 19 | DIS_REFCLK | 0 | Disable |
| | | 1 | Enable |
| 18 | EN_3P | 0 | No effect |
| | | 1 | Generates single phase output and puts other phases at ‘0’ |
| 17 | PFD_CLR | | Reset for phase-frequency detector |
| | | 0 | Closed loop |
| | | 1 | |
| 16 | CLK_FLIP | 0 | No effect |
| | | 1 | Flips the output clock phase |
| 15 | EN_RTRIM | | Enable resistor calibration |
| | | 0 | Disable |
| | | 1 | Enable |
| 14 | EN_MEAS | | Enable measurement circuit inside APLL ANA |
| | | 0 | ? |
| | | 1 | ? |
| 13-12 | EN_LATCH | | Enable output latch in differential ring |
| | | 0 | ? |
| | | 1 | ? |
| 11-8 | CP_CTRL | | Charge pump control bit +- 50 % |
| 7-6 | RESVALUE | | Sets loop filter resistor value |
| 5 | C1_2X | | Increase filter capacitance by 2x in low reference (20 MHz) clock |
| 4 | ENDIGLDO | | Enable DIG LDO. Required to wake pll up for plll dig |
| | | 0 | Disable |
| | | 1 | Enable |
| 3 | SELSC | | SELSC |
| 2 | ENBGSC_REF | | ENBGSC_REF |
| 1 | ENPLLDO | | Enable PLL LDO |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | ENPLL | | Enable PLL |
| | | 0 | Disable |
| | | 1 | Enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.5.2 SATA PLL Configuration 1 Register (SATA_PLLCFG1)

The register access is determined by the state of the lock bit. When the lock bit is 0 and the module is in NonSec Priv mode, bit access is Read/Write as shown in Figure 21-48. Otherwise, bit access is read only. When the lock bit is set to 1, the register is not accessible.

The SATA_PLLCFG1 register is shown in Figure 21-48 and described in Table 21-50.

**Figure 21-48. SATA PLL Configuration 1 Register (SATA_PLLCFG1)**

| 31 | 30 | 29 | 26 | 25 | 24 |
|---|---|---|---|---|---|
| ENSATAMODE | PLLREFSEL | NP1_DIV_INT | | MDIVINT | |
| R/W-1 | R/W-0 | R/W-0000 | | R/W-01001011 | |

| 23 | 18 | 17 | 16 |
|---|---|---|---|
| MDIVINT | | MDIVFRAC | |
| R/W-01001011 | | R/W-00h | |

| 15 | 8 |
|---|---|
| MDIVFRAC | |
| R/W-00h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MDIVFRAC | | EN_CLKAUX | EN_CLK125M | EN_CLK100M | EN_CLK50M | ENSSC | MDIVPULSE |
| R/W-00h | | R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-50. SATA PLL Configuration 1 Register (SATA_PLLCFG1) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | ENSATAMODE | | Mode select |
| | | 0 | PCIe (2.5 GHz) |
| | | 1 | SATA (1.5 GHz). Used by APLLDIG and TRX_DIG for different purpose. Needs to configure this bit to get spectrum clocks.<br>**Note:** Always program to 0. |
| 30 | PLLREFSEL | | Chose reference clock value. (Note: only needs to be set if using DLL outputs) |
| | | 0 | 100 MHz |
| | | 1 | 20 MHz |
| 29-26 | NP1_DIV_INT | | Integer value of N+1 divider |
| 25-18 | MDIVINT | | 8-bit value, which is used as integer portion of feedback divider |
| 17-6 | MDIVFRAC | | 12-bit value, which is used as fractional portion of feedback divider |
| 5 | EN_CLKAUX | | Enable for spectrum clock (testclk) |
| | | 0 | Enable |
| | | 1 | Disable |
| 4 | EN_CLK125M | | Enable for 125MHz clock |
| | | 0 | Disable |
| | | 1 | Enable |
| 3 | EN_CLK100M | | Enable for 100MHz clock |
| | | 0 | Disable |
| | | 1 | Enable |
| 2 | EN_CLK50M | | Enable for 50MHz clock |
| | | 0 | Disable |
| | | 1 | Enable |
| 1 | ENSSC | | Enable spread spectrum support |
| | | 0 | Disable |
| | | 1 | Enable |
| 0 | MDIVPULSE | | Update M div when pulse is high |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.5.3 SATA PLL Configuration 2 Register (SATA_PLLCFG2)

The register access is determined by the state of the lock bit. When the lock bit is 0 and the module is in NonSec Priv mode, bit access is Read/Write as shown in Figure 21-49. Otherwise, bit access is read only. When the lock bit is set to 1, the register is not accessible.

The SATA_PLLCFG2 register is shown in Figure 21-49 and described in Table 21-51.

**Figure 21-49. SATA PLL Configuration 2 Register (SATA_PLLCFG2)**

| 31 | 30 24 | 23 21 | 20 0 |
|---|---|---|---|
| SSCDNSPREAD | SSCMANT | SSCEXPO | SSCFRSPREAD |
| R/W-00h | R/W-00h | R/W-00h | R/W-00h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-51. SATA PLL Configuration 2 Register (SATA_PLLCFG2) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31 | SSCDNSPREAD | | SSC downspread |
| 30-24 | SSCMANT | | Mantesa portion of the modified frequency in SSC operation |
| 23-21 | SSCEXPO | | Exponent portion of the modified frequency in SSC operation |
| 20-0 | STS_TX | | SSC frequency spread |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.5.4 SATA PLL Configuration 3 Register (SATA_PLLCFG3)

The register access is determined by the state of the lock bit. When the lock bit is 0 and the module is in NonSec Priv mode, bit access is Read/Write as shown in Figure 21-50. Otherwise, bit access is read only. When the lock bit is set to 1, the register is not accessible.

The SATA_PLLCFG3 register is shown in Figure 21-50 and described in Table 21-52.

**Figure 21-50. SATA PLL Configuration 3 Register (SATA_PLLCFG3)**

| 31 27 | 26 | 25 | 24 |
|---|---|---|---|
| Reserved | DIGLDO_PULLDOWNZ | DIGLDO_ENFUNC5 | DIGLDO_ENFUNC4 |
| R-0 | R/W-1 | R/W-0 | R/W-0 |

| 23 | 22 | 21 | 20 |
|---|---|---|---|
| DIGLDO_ENFUNC3 | DIGLDO_ENFUNC2 | DIGLDO_ENFUNC1 | DIGLDO_VSET |
| R/W-0 | R/W-0 | R/W-0 | R/W-00000 |

| 19 16 |
|---|
| DIGLDO_VSET |
| R/W-00000 |

| 15 13 | 12 0 |
|---|---|
| Reserved | PLLLDO_CTRL |
| R-000 | R/W-0000000000000 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-52. SATA PLL Configuration 3 Register (SATA_PLLCFG3) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-27 | Reserved | | Reserved |
| 26 | DIGLDO_PULLDOWNZ | | TestMode: DIG LDO test mode: Details under definition |
| 25 | DIGLDO_ENFUNC5 | | TestMode: DIG LDO test mode: Details under definition |
| 24 | DIGLDO_ENFUNC4 | | TestMode: DIG LDO test mode: Details under definition |
| 23 | DIGLDO_ENFUNC3 | | TestMode: DIG LDO test mode: Details under definition |
| 22 | DIGLDO_ENFUNC2 | | TestMode: DIG LDO test mode: Details under definition |
| 21 | DIGLDO_ENFUNC1 | | TestMode: DIG LDO test mode: Details under definition |
| 20-16 | DIGLDO_VSET | | TestMode: DIG LDO test mode: Details under definition |
| 15-13 | Reserved | | Reserved |
| 12-0 | PLLLDO_CTRL | | TestMode: PLL LDO test mode: Details under definition |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.5.5 SATA PLL Configuration 4 Register (SATA_PLLCFG4)

The register access is determined by the state of the lock bit. When the lock bit is 0 and the module is in NonSec Priv mode, bit access is Read/Write as shown in Figure 21-51. Otherwise, bit access is read only. When the lock bit is set to 1, the register is not accessible.

The SATA_PLLCFG4 register is shown in Figure 21-51 and described in Table 21-53.

**Figure 21-51. SATA PLL Configuration 4 Register (SATA_PLLCFG4)**

| 31 | 26 | 25 | 24 |
|---|---|---|---|
| Reserved | | AUX_CLK_SEL | AUX_DIV |
| R/W-000000 | | R/w-0 | R/W-0x00 |

| 23 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|
| AUX_DIV | | RTRIM_RANGE | | RTRIM_EXT_EN | RTRIM_SPEED |
| R/W-0x00 | | R/W-00 | | R/W-0 | R/W-0 |

| 15 | 14 | 13 | 10 | 9 | 8 |
|---|---|---|---|---|---|
| RTRIM_MODE | | RTRIM_EXT_VAL | | VTUNE_RANGE | |
| R/W-0 | | R/W-0x00 | | R/W-00 | |

| 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| VTUNE_EXT_EN | VTUNE_SPEED | VTUNE_MODE | | VTUNE_EXT_VAL | |
| R/W-0 | R/W-0 | R/W-00 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-53. SATA PLL Configuration 4 Register (SATA_PLLCFG4) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-26 | Reserved | 1 | Reserved |
| 25 | AUX_CLK_SEL | | Auxiliary clock select |
| | | 0 | Divided |
| | | 1 | Ref clk |
| 24-20 | AUX_DIV | | Auxiliary divider control |
| 19-18 | RTRIM_RANGE | | Range for rtrim |
| | | 00 | 0.9-0.8 |
| | | 01 | 1.0-0.9 |
| | | 10 | 1.1-1.0 |
| | | 11 | 1.1-0.9 |
| 17 | RTRIM_EXT_EN | | Select loop starting point. |
| | | 0 | Mid-code “0000” |
| | | 1 | External input RTRIM_EXT_VAL |
| 16 | RTRIM_SPEED | | Selection on wait for # REFCLKs after previous update |
| | | 0 | 128 |
| | | 1 | 256 |
| 15-14 | RTRIM_MODE | | Loop mode |
| | | 00 | Off |
| | | 10 | On - Freeze (after lock freeze loop) |
| | | 11 | On - Continuous (after lock keep lock running) |
| 13-10 | RTRIM_EXT_VAL | | 2’s compliment vtune control loop value. Valid range : “0100” (max) to “0000” (mid) to “1100 (min) |
| 9-8 | VTUNE_RANGE | | Range for Vtune |
| | | 00 | 0.9-0.8 |
| | | 01 | 1.0-0.9 |
| | | 10 | 1.1-1.0 |
| | | 11 | 1.1-0.9 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 21-53. SATA PLL Configuration 4 Register (SATA_PLLCFG4) Field Description (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7 | VTUNE_EXT_EN | | Select loop starting point |
| | | 0 | Mid-code "0000" |
| | | 1 | External input VTUNE_EXT_VAL |
| 6 | VTUNE_SPEED | | Selection on wait for # REFCLKs after previous update |
| | | 0 | 128 |
| | | 1 | 256 |
| 5-4 | VTUNE_MODE | | Loop mode |
| | | 00 | Off |
| | | 10 | On - Freeze (after lock freeze loop) |
| | | 11 | On - Continuous (after lock keep lock running) |
| 3-0 | VTUNE_EXT_VAL | | 2’s compliment vtune control loop value. Valid range : “0100” (max) to “0000” (mid) to “1100 (min) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.5.6 SATA PLL Status (SATA_PLLSTATUS)

The SATA PLL status register (SATA_PLLSTATUS) is used to reflect the SATA PLL status.

The register access is affected by the state of the lock bit. When the lock bit is set to 0, access is read only. When the lock bit is set to 1, the register is not accessible.

The SATA_PLLSTATUS register is shown in Figure 21-52 and described in Table 21-54.

**Figure 21-52. SATA PLL Status (SATA_PLLSTATUS)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0000h | |

| 15 | 12 | 11 | 8 |
|---|---|---|---|
| Reserved | | RTRIMSTS | |
| R-0000h | | R-0000 | |

| 7 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| VTUNESTS | | APLLDIGSTS5 | APLLDIGSTS4 | APLLDIGSTS1 | APLLDIGSTS0 |
| R-0000 | | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-54. SATA PLL Status (SATA_PLLSTATUS) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | Reserved | | Reserved |
| 11-8 | RTRIMSTS | | Value of the rtrim loop (may not be static in Continuous mode of operation) |
| 7-4 | VTUNESTS | | Value of the vtune control loop (may not be static in Continuous mode of operation) |
| 3 | APLLDIGSTS5 | | Not used |
| 2 | APLLDIGSTS4 | | Not used |
| 1 | APLLDIGSTS1 | | SSC is engaged |
| 0 | APLLDIGSTS0 | | APLL is locked |

### 21.5.7 SATA Receive Status (SATA_RXSTATUS)

The SATA receive status register (SATA_RXSTATUS) is used to reflect the SATA receive status.

The register access is affected by the state of the lock bit. When the lock bit is set to 0, access is read only. When the lock bit is set to 1, the register is not accessible.

The SATA_RXSTATUS register is shown in Figure 21-53 and described in Table 21-55.

**Figure 21-53. SATA Receive Status (SATA_RXSTATUS)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TESTFAIL |
| R-00 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-55. SATA Receive Status (SATA_RXSTATUS) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | | Reserved |
| 0 | TESTFAIL | | Test failure. Driven high when an error is encountered during a test sequence executed on channel i. Synchronous to RXBCLKIN[i]. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.5.8 SATA Transmit Status (SATA_TXSTATUS)

The SATA transmit status register (SATA_TXSTATUS) is used to reflect the SATA transmit status.

The register access is affected by the state of the lock bit. When the lock bit is set to 0, access is read only. When the lock bit is set to 1, the register is not accessible.

The SATA_TXSTATUS register is shown in Figure 21-54 and described in Table 21-56.

**Figure 21-54. SATA Transmit Status (SATA_TXSTATUS)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | TESTFAIL |
| R-00 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-56. SATA Transmit Status (SATA_TXSTATUS) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | | Reserved |
| 0 | TESTFAIL | | Test failure. Driven high when an error is encountered during a test sequence executed on channel i, or if an over or underflow occurs in the Tx serialiser when ENFTP = 0 (see section 7.3.3). Synchronous to TXBCLKIN[i]. |

### 21.5.9 SATA TEST Configuration (SATA_TESTCFG)

The SATA test configuration register (SATA_TESTCFG) is used to control ??.

The register access is determined by the state of the lock bit. When the lock bit is 0 and the module is in NonSec Priv mode, bit access is Read/Write as shown in Figure 21-55. Otherwise, bit access is read only. When the lock bit is set to 1, the register is not accessible.

The SATA_TESTCFG register is shown in Figure 21-55 and described in Table 21-57.

**Figure 21-55. SATA TEST Configuration (SATA_TESTCFG)**

| 31 | 10 | 9 | 7 | 6 | 3 | 2 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | TX_TESTPATT | | Reserved | | RX_TESTPATT | |
| R-00 | | R/W-000 | | R-000 | | R/W-000 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-57. SATA TEST Configuration (SATA_TESTCFG) Field Description**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | | Reserved |
| 9-7 | TX_TESTPATT | | Not Used. The RX has similar settings, so this field is not used. |
| 6-3 | Reserved | | Reserved |
| 0 | RX_TESTPATT | | Enables and selects test patterns. |
| | | 000 | Reserved |
| | | 001 | An alternating 0/1 pattern with a period of 2 UI. |
| | | 010 | Uses a 7-bit LFSR with feedback polynomial x7+x6+1. |
| | | 011 | Uses a 23-bit LFSR with feedback polynomial x23+x18+1. |
| | | 100 | Uses a 31-bit LFSR with feedback polynomial x31+x28+1. |
| | | 101 | Reserved |
| | | 110 | Reserved |
| | | 111 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

## 21.6 SATA Subsystem Registers

The following PRCM Registers are used to release the SATA Controller from Reset. These registers are found within the PRCM module.

The Base address for the PRCM module is 4818_0000h.

**Table 21-58. SATA Subsystem Controller Registers**

| Address Offset | Acronym | Register Description | Section |
|---|---|---|---|
| 504h | CM_DEFAULT_L3_MED_CLKSTCTRL | Clock Domain Power Control Register | Section 21.6.1 |
| 560h | CM_ALWON2_SATA_CLKCTRL | SATA Clock Management/Control Register | Section 21.6.2 |

### *21.6.1 Clock Domain Power Control Register (CM_DEFAULT_L3_MED_CLKSTCTRL)*

The clock domain power control register (CM_DEFAULT_L3_MED_CLKSTCTRL) enables the domain power state transition. It controls the software-supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also holds one status bit per clock input of the domain.

The CM_DEFAULT_L3_MED_CLKSTCTRL register is shown in Figure 21-56 and described in Table 21-59.

**Figure 21-56. Clock Domain Power Control Register (CM_DEFAULT_L3_MED_CLKSTCTRL)**

| 31 | 26 | 25 | 24 |
|---|---|---|---|
| Reserved | | Reserved | |
| R-00h | | R-00000h | |

| 23 | 16 |
|---|---|
| Reserved | |
| R-00000h | |

| 15 | 9 | 8 |
|---|---|---|
| Reserved | | CLKACTIVITY_L3_MED_GCLK |
| R-00000h | | R-0 |

| 7 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | CLKTRCTRL | |
| R-00h | | R/W-1h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 21-59. Clock Domain Power Control Register (CM_DEFAULT_L3_MED_CLKSTCTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | | Reserved |
| 8 | CLKACTIVITY_L3_MED_GCLK | | This field indicates the state of the L3_SLOW_GCLK clock in the domain. |
| | | 0 | Corresponding clock is gated. |
| | | 1 | Corresponding clock is active. |
| 7-2 | Reserved | | Reserved |
| 1-0 | CLKTRCTRL | | Controls the clock state transition of the L3_MED clock domain in DEFAULT power domain. |
| | | 0h | NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. |
| | | 1h | SW_SLEEP: Start a software forced sleep transition on the domain. |
| | | 2h | SW_WKUP: Start a software forced wake-up transition on the domain. |
| | | 3h | HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions |


Copyright © 2011–2016, Texas Instruments Incorporated

### 21.6.2 Clock Domain SATA Control Register (CM_DEFAULT_SATA_CLKCTRL)

The clock domain SATA control register (CM_DEFAULT_SATA_CLKCTRL) manages the SATA clocks.

The CM_DEFAULT_SATA_CLKCTRL register is shown in Figure 21-57 and described in Table 21-60.

**Figure 21-57. Clock Domain SATA Control Register (CM_DEFAULT_SATA_CLKCTRL)**

| 31 | 24 |
|---|---|
| Reserved | |
| R-0000h | |

| 23 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|
| Reserved | | STBYST | IDLEST | |
| R-0000h | | R-1 | R-3h | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0000 | |

| 7 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | MODULEMODE | |
| R-0000h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 21-60. Clock Domain SATA Control Register (CM_DEFAULT_SATA_CLKCTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-19 | Reserved | | Reserved |
| 18 | STBYST | | Module standby status |
| | | 0 | Module is functional (not in standby). |
| | | 1 | Module is in standby. |
| 17-16 | IDLEST | | Module idle status |
| | | 0h | Module is fully functional, including OCP. |
| | | 1h | Module is performing transition: wakeup, or sleep, or sleep abortion. |
| | | 2h | Module is in Idle mode (only OCP part). It is functional if using separate functional clock. |
| | | 3h | Module is disabled and cannot be accessed. |
| 15-2 | Reserved | | Reserved |
| 1-0 | MODULEMODE | | Controls the way mandatory clocks are managed. |
| | | 0h | Module is disable by software. Any OCP access to the module results in an error, except if resulting from a module wakeup (asynchronous wakeup). |
| | | 1h | Reserved |
| | | 2h | Module is explicitly enabled. Interface clock (if not used for functions) can be gated according to the clock domain state. Functional clocks will stay present. As long as in this configuration, power domain sleep transition cannot happen. |
| | | 3h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

# Serial Port Interface (SPI)

This chapter describes the master/slave multichannel serial port interface (SPI) module.

**Topic** **Page**

**22.1 Introduction** ..... **3487**
**22.2 Architecture** ..... **3488**
**22.3 SPI Registers** ..... **3526**

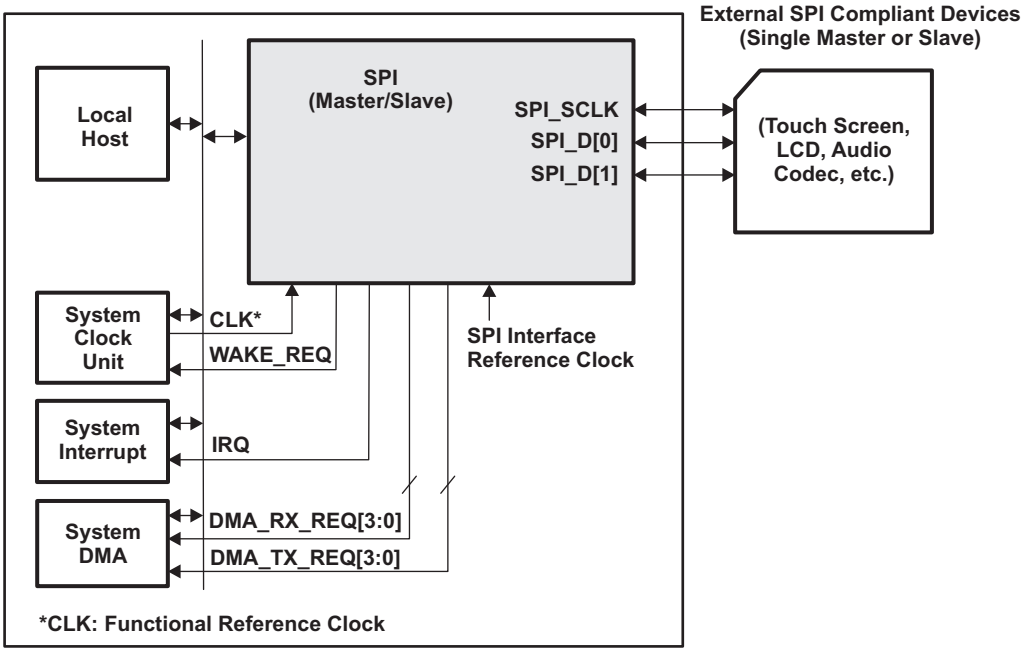

Copyright © 2011–2016, Texas Instruments Incorporated

## 22.1 Introduction

### 22.1.1 Overview

SPI is a general-purpose receive/transmit master/slave controller that can interface with up to four slave external devices or one single external master. It allows a duplex, synchronous, serial communication between a CPU and SPI compliant external devices (slaves and masters). Figure 22-1 shows a high-level overview of a SPI system.

**Figure 22-1. SPI System Overview**




### 22.1.2 Features

The SPI includes these distinctive features:

- Serial clock with programmable frequency, polarity and phase
- It supports maximum frequency of 48 MHz.
- Clock frequency granularity can be changed: power of two or one single clock cycle.
- SPI enable generation programmable
- SPI enable with programmable polarity
- Wide selection of SPI word lengths ranging from 4 bits to 32 bits
- Master / Slave modes
- Master multichannel mode:
  - SPI interface provides supports for up to four SPI channels
  - SPI word Transmit / Receive slot assignment based on round robin arbitration
  - SPI configuration per channel (clock definition, enable polarity and word width)
  - Full duplex/half duplex
  - Transmit-only/receive-only/transmit-and-receive modes
  - Flexible I/O port controls per channel
  - Programmable clock granularity
- Independent DMA requests (read/write) per channel


Copyright © 2011–2016, Texas Instruments Incorporated

- Single interrupt line for multiple interrupt source events
- 32-bit-wide bus, for system configuration and data transfer
- Programmable delay before the first SPI word transmitted
- No dead cycle between two successive words in slave mode
- Multiple SPI word access with a channel using a FIFO enabled
- Programmable timing control between chip select and external clock generation
- Support DMA providing 256 bit aligned addresses when FIFO is used
- Built-in FIFO available for a single channel. FIFO size of up to 64 bytes is supported.

## 22.2 Architecture

### *22.2.1 SPI Interface*

This section lists the name and description of the SPI interface used for connection to external SPI compliant devices. SPI has a total of seven external SPI signals shown in Table 22-1.

**Table 22-1. SPI Interface Pins**

| Pin | Type | Description |
|---|---|---|
| SPI_SCLK | I/O | SPI serial clock (output when master, input when slave) |
| SPI_D[0] | I/O | Can be configured as either input or output (MOSI or MISO) |
| SPI_D[1] | I/O | Can be configured as either input or output (MOSI or MISO) |
| $\overline{\text{SPI_SCS[0]}}$ | I/O | SPI chip select 0 output when master, input when slave (active low) |
| $\overline{\text{SPI_SCS[1]}}$ | I/O | SPI chip select 1 output when master, input when slave (active low) |
| $\overline{\text{SPI_SCS[2]}}$ | I/O | SPI chip select 2 output when master, input when slave (active low) |
| $\overline{\text{SPI_SCS[3]}}$ | I/O | SPI chip select 3 output when master, input when slave (active low) |

### *22.2.2 SPI Transmission*

This section describes the transmissions supported by SPI. The SPI protocol is a synchronous protocol that allows a master device to initiate serial communication with a slave device. Data is exchanged between these devices. A slave select line ($\overline{\text{SPI_SCS[n]}}$) can be used to allow selection of an individual slave SPI device. Slave devices that are not selected do not interfere with SPI bus activities. Connected to multiple external devices, SPI exchanges data with a single SPI device at a time through two main modes:

- Two data pins interface mode. (See Section 22.2.2.1)
- Single data pin interface mode (recommended for half-duplex transmission). (See Section 22.2.2.2)

The flexibility of SPI allows exchanging data with several formats through programmable parameters described in Section 22.2.2.3.


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.2.2.1 Two Data Pins Interface Mode

The two data pins interface mode, allows a full duplex SPI transmission where data is transmitted (shifted out serially) and received (shifted in serially) simultaneously on separate data lines SPI_D[0] and SPI_D[1]. Data leaving the master exits on transmit serial data line also known as MOSI: MasterOutSlaveIn. Data leaving the slave exits on the receive data line also known as MISO: MasterInSlaveOut.

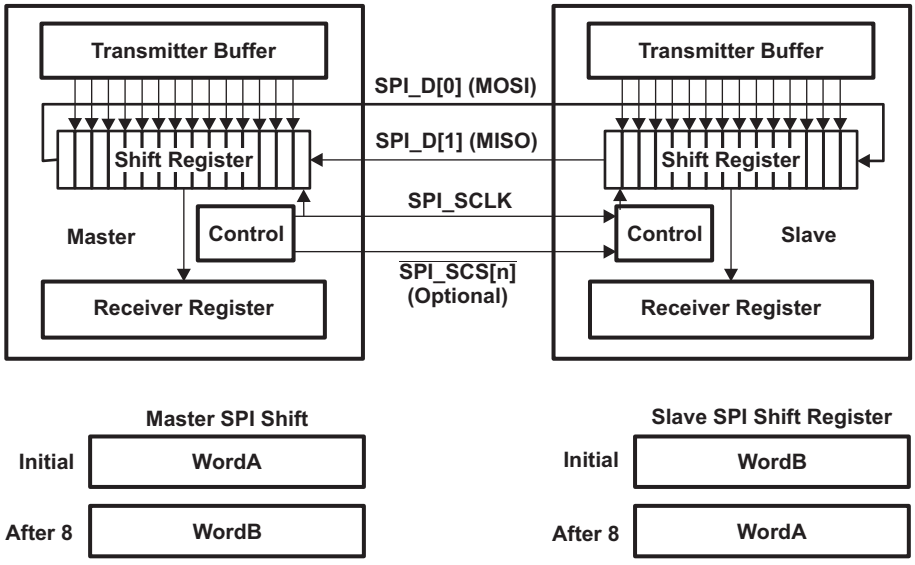
SPI has a unified SPI port control: SPI_D[1:0] can be independently configured as receive or transmit lines. The user has the responsibility to program which data line to use and in which direction (receive or transmit), according to the external slave/master connection.

The serial clock (SPI_SCLK) synchronizes shifting and sampling of the information on the two serial data lines (SPI_D[1:0]). Each time a bit is transferred out from the Master, one bit is transferred in from Slave.

Figure 22-2 shows an example of a full duplex system with a Master device on the left and a Slave device on the right. After 8 cycles of the serial clock SPI_SCLK, the WordA has been transferred from the master to the slave. At the same time, the WordB has been transferred from the slave to the master.

When referring to the master device, the control block transmits the clock SPI_SCLK and the enable signal ($\overline{\text{SPI_SCS[n]}}$) (optional depends on MCSPI_MODULECTRL[PIN34] bit field).

**Figure 22-2. SPI Full-Duplex Transmission**




### 22.2.2.2 Single Data Pin Interface Mode

In single data pin interface mode, under software control, a single data line is used to alternatively transmit and receive data (Half duplex transmission).

SPI has a unified SPI port control: SPI_D[1:0] can be independently configured as receive or transmit lines. The user has the responsibility to program which data line to use and in which direction (receive or transmit), according to the external slave/master connection.

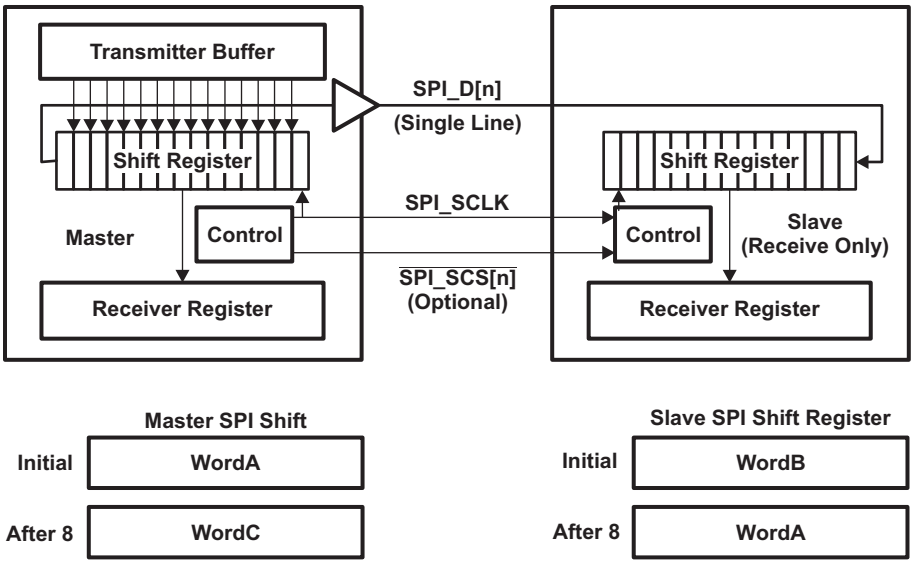
As for a full duplex transmission, the serial clock (SPI_SCLK) synchronizes shifting and sampling of the information on the single serial data line.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.2.2.1 Example With a Receive-Only Slave

Figure 22-3 shows a half duplex system with a Master device on the left and a receive-only Slave device on the right. Each time a bit is transferred out from the Master, one bit is transferred in the Slave. After 8 cycles of the serial clock SPI_SCLK, the WordA has been transferred from the master to the slave.

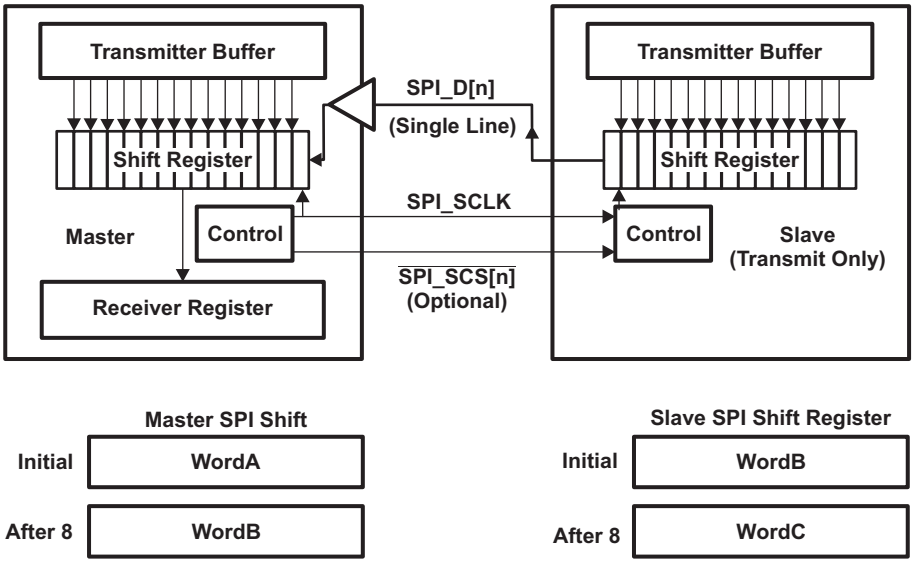
**Figure 22-3. SPI Half-Duplex Transmission (Receive-only Slave)**




#### 22.2.2.2.2 Example With a Transmit-Only Slave

Figure 22-4 shows a half duplex system with a Master device on the left and a transmit-only Slave device on the right. Each time a bit is transferred out from the Slave, one bit is transferred in the Master. After 8 cycles of the serial clock SPI_SCLK, the WordA has been transferred from the slave to the master.

**Figure 22-4. SPI Half-Duplex Transmission (Transmit-Only Slave)**





Copyright © 2011–2016, Texas Instruments Incorporated

### 22.2.2.3 Transfer Formats

This section describes the transfer formats supported by SPI.

The flexibility of SPI allows setting the parameters of the SPI transfer:

- SPI word length
- SPI enable generation programmable
- SPI enable assertion
- SPI enable polarity
- SPI clock frequency
- SPI clock phase
- SPI clock polarity

The consistency between SPI word length, clock phase and clock polarity of the master SPI device and the communicating slave device remains under software responsibility.

#### *22.2.2.3.1 Programmable Word Length*

SPI supports any SPI word from 4 to 32 bits long.

The SPI word length can be changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

#### *22.2.2.3.2 Programmable SPI Enable Generation*

SPI is able to generate or not the SPI enable, if management of chip select is de-asserted a point to point connection is mandatory. Only a single master of slave device can be connected to the SPI bus.

#### *22.2.2.3.3 Programmable SPI Enable ($\overline{\text{SPI_SCS[n]}}$)*

The polarity of the $\overline{\text{SPI_SCS[n]}}$ signals is programmable. $\overline{\text{SPI_SCS[n]}}$ signals can be active high or low.

The assertion of the $\overline{\text{SPI_SCS[n]}}$ signals is programmable: $\overline{\text{SPI_SCS[n]}}$ signals can be manually asserted or can be automatically asserted.

Two consecutive words for two different slave devices may go along with active $\overline{\text{SPI_SCS[n]}}$ signals with different polarity.

#### *22.2.2.3.4 Programmable SPI Clock (SPI_SCLK)*

The phase and the polarity of the SPI serial clock are programmable when SPI is a SPI master device or a SPI slave device. The baud rate of the SPI serial clock is programmable when it is a SPI master.

When SPI is operating as a slave, the serial clock SPI_SCLK is an input from the master.

#### *22.2.2.3.5 Bit Rate*

In Master Mode, an internal reference clock CLKSPIREF is used as an input of a programmable divider to generate bit rate of the serial clock SPI_SCLK. Granularity of this clock divider can be changed.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.2.3.6 Polarity and Phase

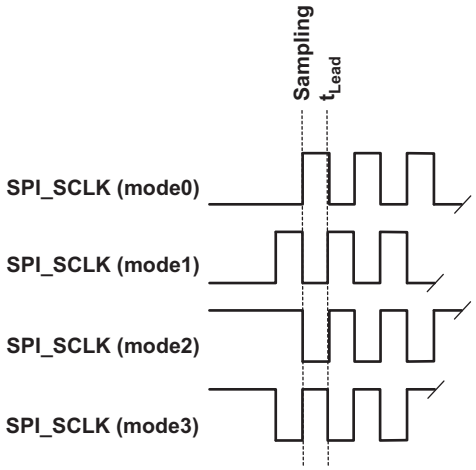
SPI supports four sub-modes of the SPI format transfer that depend on the polarity (POL) and the phase (PHA) of the SPI serial clock (SPI_SCLK). Table 22-2 and Figure 22-5 show a summary of the four sub-modes. Software selects one of four combinations of serial clock phase and polarity.

Two consecutive SPI words for two different slave devices may go along with active SPI_SCLK signal with different phase and polarity.

**Table 22-2. Phase and Polarity Combinations**

| Polarity (POL) | Phase (PHA) | SPI Mode | Comments |
|---|---|---|---|
| 0 | 0 | mode0 | SPI_SCLK active high and sampling occurs on the rising edge. |
| 0 | 1 | mode1 | SPI_SCLK active high and sampling occurs on the falling edge. |
| 1 | 0 | mode2 | SPI_SCLK active low and sampling occurs on the falling edge. |
| 1 | 1 | mode3 | SPI_SCLK active low and sampling occurs on the rising edge. |

**Figure 22-5. Phase and Polarity Combinations**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.2.3.7 Transfer Format With PHA = 0

This section describes the concept of a SPI transmission with the SPI mode0 and the SPI mode2.

In the transfer format with PHA = 0, $\overline{\text{SPI_SCS[n]}}$ is activated a half cycle of SPI_SCLK ahead of the first SPI_SCLK edge.

In both master and slave modes, SPI drives the data lines at the time of $\overline{\text{SPI_SCS[n]}}$ is asserted.

Each data frame is transmitted starting with the MSB. At the extremity of both SPI data lines , the first bit of SPI word is valid a half cycle of SPI_SCLK after the $\overline{\text{SPI_SCS[n]}}$ assertion.

Therefore, the first edge of the SPI_SCLK line is used by the master to sample the first data bit sent by the slave. On the same edge, the first data bit sent by the master is sampled by the slave.

On the next SPI_SCLK edge, the received data bit is shifted into the shift register, and a new data bit is transmitted on the serial data line.

This process continues for a total of pulses on the SPI_SCLK line defined by the SPI word length programmed in the master device, with data being latched on odd numbered edges and shifted on even numbered edges.

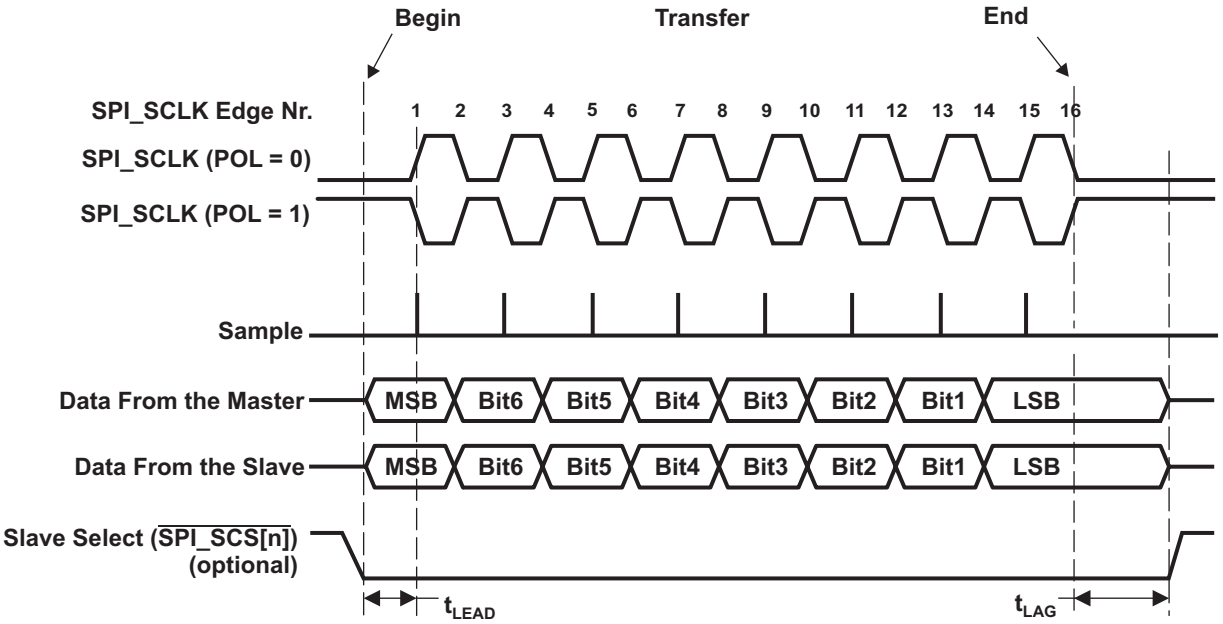
Figure 22-6 is a timing diagram of a SPI transfer for the SPI mode0 and the SPI mode2, when SPI is master or slave, with the frequency of SPI_SCLK equals to the frequency of CLKSPIREF. It should not be used as a replacement for SPI timing information and requirements.

When SPI is in slave mode, if the $\overline{\text{SPI_SCS[n]}}$ line is not de-asserted between successive transmissions then the content of the Transmitter register is not transmitted, instead the last received SPI word is transmitted.

In master mode, the $\overline{\text{SPI_SCS[n]}}$ line must be negated and reasserted between each successive SPI word. This is because the slave select pin freezes the data in its shift register and does not allow it to be altered if PHA bit equals 0.

In 3-pin mode without using the $\overline{\text{SPI_SCS[n]}}$ signal, the controller provides the same waveform but with $\overline{\text{SPI_SCS[n]}}$ forced to low state. In this mode, $\overline{\text{SPI_SCS[n]}}$ is useless.

**Figure 22-6. Full Duplex Single Transfer Format with PHA = 0**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.2.3.8 Transfer Format With PHA = 1

This section describes SPI full duplex transmission with the SPI mode1 and the SPI mode 3.

In the transfer format with PHA = 1, $\overline{\text{SPI_SCS[n]}}$ is activated a delay ($t_{Lead}$) ahead of the first SPI_SCLK edge.

In both master and slave modes, SPI drives the data lines on the first SPI_SCLK edge.

Each data frame is transmitted starting with the MSB. At the extremity of both SPI data lines , the first bit of SPI word is valid on the next SPI_SCLK edge, a half cycle of SPI_SCLK later. It is the sampling edge for both the master and slave.

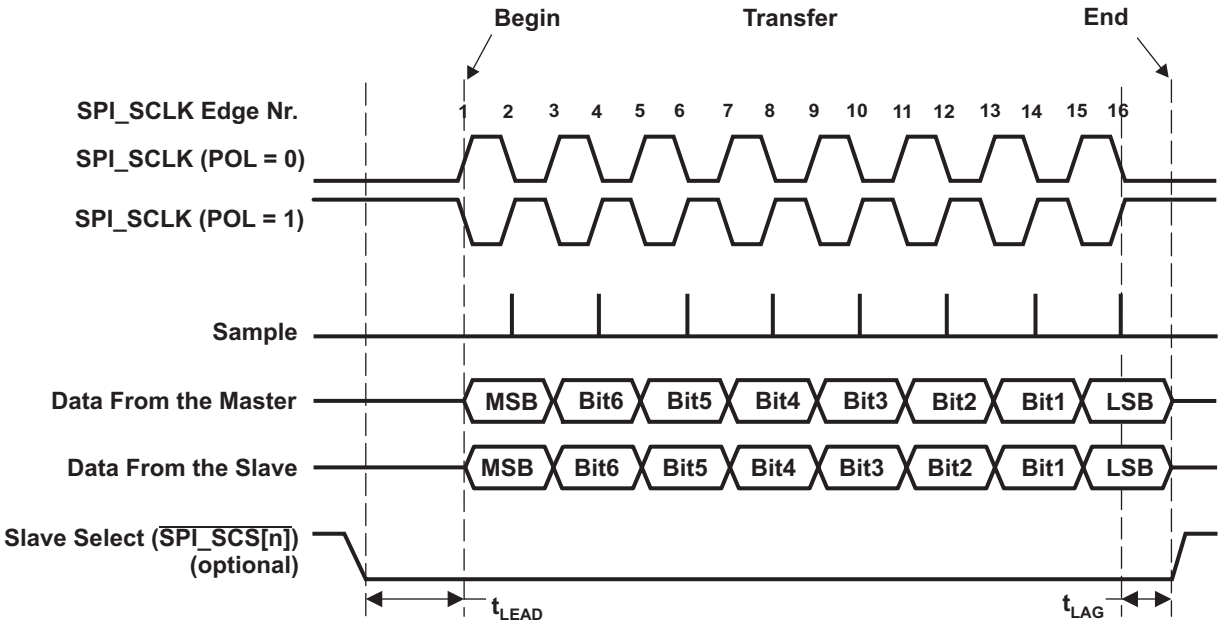
When the third edge occurs, the received data bit is shifted into the shift register. The next data bit of the master is provided to the serial input pin of the slave.

This process continues for a total of pulses on the SPI_SCLK line defined by the word length programmed in the master device, with data being latched on even numbered edges and shifted on odd numbered edges.

Figure 22-7 is a timing diagram of a SPI transfer for the SPI mode1 and the SPI mode3, when SPI is master or slave, with the frequency of SPI_SCLK equals to the frequency of CLKSPIREF. It should not be used as a replacement for SPI timing information and requirements.

The $\overline{\text{SPI_SCS[n]}}$ line may remain active between successive transfers. In 3-pin mode without using the $\overline{\text{SPI_SCS[n]}}$ signal, the controller provides the same waveform but with $\overline{\text{SPI_SCS[n]}}$ forced to low state. In this mode, $\overline{\text{SPI_SCS[n]}}$ is useless.

**Figure 22-7. Full Duplex Single Transfer Format With PHA = 1**





Copyright © 2011–2016, Texas Instruments Incorporated

### 22.2.3 Master Mode

SPI is in master mode when the MS bit of the register MCSPI_MODULCTRL is cleared.

In master mode SPI supports multi-channel communication with up to four independent SPI communication channel contexts. SPI initiates a data transfer on the data lines (SPI_D[1:0]) and generates clock (SPI_SCLK) and control signals ($\overline{\text{SPI_SCS[n]}}$) to a single SPI slave device at a time.

#### 22.2.3.1 Dedicated Resources Per Channel

In the following sections, the letter “I” indicates the channel number that can be 0, 1, 2, or 3. Each channel has the following dedicated resources:

- Its own channel enable, programmable with the bit EN of the register MCSPI_CH(I)CTRL. Disabling the channel, outside data word transmission, remains under user responsibility.
- Its own transmitter register MCSPI_TX on top of the common shift register. If the transmitter register is empty, the status bit TXS of the register MCSPI_CH(I)STAT is set.
- Its own receiver register MCSPI_RX on top of the common shift register. If the receiver register is full, the status bit RXS of the register MCSPI_CH(I)STAT is set.
- A fixed SPI ENABLE line allocation ($\overline{\text{SPI_SCS[ I]}}$ port for channel I), SPI enable management is optional.
- Its own communication configuration with the following parameters via the register (I)CONF
  - Transmit/Receive modes, programmable with the bit TRM.
  - Interface mode (Two data pins or Single data pin) and data pins assignment, both programmable with the bits IS and DPE.
  - SPI word length, programmable with the bits WL.
  - SPI_SCS[n] polarity, programmable with the bit EPOL.
  - SPI_SCS[n] kept active between words, programmable with the bit FORCE.
  - Turbo mode, programmable with the bit TURBO.
  - SPI_SCLK frequency, programmable with the bit CLKD, the granularity of clock division can be changed using CLKG bit, the clock ratio is then concatenated with MCSPI_CHCTRL[EXTCLK] value.
  - SPI_SCLK polarity, programmable with the bit POL
  - SPI_SCLK phase, programmable with the bit PHA.
  - Start bit extension enable, programmable with the bit SBE to support LoSSI transfer specification.
  - Start bit polarity, programmable with the bit SBPOL
  - Use a FIFO Buffer or not( ), programmable with FFER and FFEW, depending on transfer mode TRM.
- Two DMA requests events, read and write, to synchronize read/write accesses of the DMA controller with the activity of SPI. The DMA requests are enabled with the bits DMAR and DMAW.
- Three interrupts events

The transfers will use the latest loaded parameters of the register (I)CONF.

The configuration parameters SPI_SCS[n] polarity, Turbo mode, SPI_SCLK phase and SPI_SCLK polarity can be loaded in the (I)CONF register only when the channel is disabled. The user has the responsibility to change the other parameters of the (I)CONF register when no transfer occurs on the SPI interface.


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.2.3.2 Interrupt Events in Master Mode

In master mode, the interrupt events related to the transmitter register state are TX_empty and TX_underflow. The interrupt event related to the receiver register state is RX_full.

#### *22.2.3.2.1 TX_empty*

The event TX_empty is activated when a channel is enabled and its transmitter register becomes empty (transient event). Enabling channel automatically raises this event, except for the Master receive only mode. (See Section 22.2.3.5). When FIFO buffer is enabled (MCSPI_CH(I)CONF[FFEW] set to 1), the TX_empty is asserted as soon as there is enough space in buffer to write a number of byte defined by MCSPI_XFERLEVEL[AEL].

Transmitter register must be loaded to remove the source of the interrupt and the TX_empty interrupt status bit must be cleared for interrupt line de-assertion (if event enabled as interrupt source) . (See Section 22.2.5).

When FIFO is enabled, no new TX_empty event will be asserted as soon as CPU has not performed the number of write into transmitter register defined by MCSPI_XFERLEVEL[AEL]. It is the responsibility of CPU to perform the right number of writes.

#### *22.2.3.2.2 TX_underflow*

The event TX_underflow is activated when the channel is enabled and if the transmitter register or FIFO is empty (not updated with new data) at the time of shift register assignment.

The TX_underflow is a harmless warning in master mode.

To avoid having TX_underflow event at the beginning of a transmission, the event TX_underflow is not activated when no data has been loaded into the transmitter register since channel has been enabled.

To avoid having TX_underflow event the Transmitter register must be loaded seldom.

TX_underflow interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

**Note:** Be careful, when more than one channel have a FIFO enable bit field FFER or FFEW set, the module force no use of FIFO, software must take care that only one enabled channel is configured to use the FIFO buffer.

#### *22.2.3.2.3 RX_ full*

The event RX_full is activated when channel is enabled and receiver register becomes filled (transient event). When FIFO buffer is enabled (MCSPI_CH(I)CONF[FFER] set to 1), the RX_full is asserted as soon as there is a number of bytes holds in buffer to read defined by MCSPI_XFERLEVEL[AFL].

Receiver register must be read to remove source of interrupt and RX_full interrupt status bit must be cleared for interrupt line de-assertion (if event enabled as interrupt source).

When FIFO is enabled, no new RX_full event will be asserted as soon as CPU has not performed the number of read into receive register defined by MCSPI_XFERLEVEL[AFL]. It is the responsibility of CPU to perform the right number of reads.

#### *22.2.3.2.4 End of Word Count*

The event end of word count (EOW) is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfer defined in MCSPI_XFERLEVEL[WCNT] register. If the value was programmed to 0000h, the counter is not enabled and this interrupt is not generated.

The End of Word count interrupt also indicates that the SPI transfer is halt on channel using the FIFO buffer as soon as MCSPI_XFERLEVEL[WCNT] is not reloaded and channel re-enabled.

End of Word interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).


Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.3.3 Master Transmit and Receive Mode

This mode is programmable per channel (bits TRM of the register (I)CONF).

The channel access to the shift registers, for transmission/reception, is based on its transmitter and receiver register state and round robin arbitration.

The channel that meets the rules below is included in the round robin list of already active channels scheduled for transmission and/or reception. The arbiter skips the channel that does not meet the rules and search for the next following enabled channel, in rotation.

**Rule 1:** Only enabled channels (bit EN of the register MCSPI_CH(I)CTRL), can be scheduled for transmission and/or reception.

**Rule 2:** An enabled channel can be scheduled if its transmitter register is not empty (bit TXS of the register MCSPI_CH(I)STAT) or its FIFO is not empty in case of buffer use for the corresponding channel (bit FFE of the register MCSPI_CH(I)STAT), that is updated with new data, at the time of shift register assignment. If the transmitter register or FIFO is empty, at the time of shift register assignment, the event TX_underflow is activated and the next enabled channel with new data to transmit is scheduled. (See also transmit only mode).

**Rule 3:** An enabled channel can be scheduled if its receive register is not full (bit RXS of the register MCSPI_CH(I)STAT) ) or its FIFO is not full in case of buffer use for the corresponding channel (bit FFF of the register MCSPI_CH(I)STAT) at the time of shift register assignment. (See also receive only mode). Therefore the receiver register of FIFO cannot be overwritten. The RX_overflow bit, in the MCSPI_IRQSTATUS register is never set in this mode.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) the updated transmitter register of the next scheduled channel is loaded into the shift register. This bit is meaningless when using the Buffer for this channel. The serialization (transmit and receive) starts according to the channel communication configuration. On serialization completion the received data is transferred to the channel receive register.

The built-in FIFO is available in this mode and can be configured in one data direction Transmit or Receive, then the FIFO is seen as a unique FFNBYTE bytes buffer, or it can also be configured in both data direction Transmit and Receive, then the FIFO is split into two separate FFNBYTE/2 bytes buffer with their own address space management, in this last case the definition of AEL and AFL levels is based on FFNBYTE/2 bytes and is under CPU responsibility.

#### 22.2.3.4 Master Transmit-Only Mode

This mode avoids the CPU to read the receiver register (minimizing data movement) when only transmission is meaningful.

The master transmit only mode is programmable per channel (bits TRM of the register (I)CONF).

In master transmit only mode, transmission starts after data is loaded into the transmitter register.

**Rule 1** and **Rule 2**, defined in Section 22.2.3.3, are applicable in this mode.

**Rule 3**, defined in Section 22.2.3.3, is not applicable: In master transmit only mode, the receiver register or FIFO state “full” does not prevent transmission, and the receiver register is always overwritten with the new SPI word. This event in the receiver register is not significant when only transmission is meaningful. So, the RX_overflow bit, in the MCSPI_IRQSTATUS register is never set in this mode.

The SPI module automatically disables the RX_full interrupt status. The corresponding interrupt request and DMA Read request are not generated in master transmit only mode.

The status of the serialization completion is given by the bit EOT of the register MCSPI_CH(I)STAT. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFEW bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique FFNBYTE bytes buffer.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.3.5 Master Receive-Only Mode

This mode avoids the CPU to refill the transmitter register (minimizing data movement) when only reception is meaningful.

The master receive mode is programmable per channel (bits TRM of the register (I)CONF).

The master receive only mode enables channel scheduling only on empty state of the receiver register.

**Rule 1** and **Rule 3**, defined in Section 22.2.3.3, are applicable in this mode.

**Rule 2**, defined in Section 22.2.3.3, is not applicable: In master receive only mode, after the first loading of the transmitter register of the enabled channel, the transmitter register state is maintained as full. The content of the transmitter register is always loaded into the shift register, at the time of shift register assignment. So, after the first loading of the transmitter register, the bits TX_empty and TX_underflow, in the MCSPI_IRQSTATUS register are never set in this mode.

The status of the serialization completion is given by the bit EOT of the register MCSPI_CH(I)STAT. The bit RX_full in the MCSPI_IRQSTATUS register is set when a received data is loaded from the shift register to the receiver register. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique FFNBYTE bytes buffer.

#### 22.2.3.6 Single-Channel Master Mode

When the SPI is configured as a master device with a single enabled channel, the assertion of the SPIM_CSX signal can be controlled in two different ways:

- In 3 pin mode : MCSPI_MODULCTRL[1] PIN34 and MCSPI_MODULCTRL[0] SINGLE bits are set to 1, the controller transmit SPI word as soon as transmit register or FIFO is not empty.
- In 4 pin mode : MCSPI_MODULCTRL[1] PIN34 bit is cleared to 0 and MCSPI_MODULCTRL[0] SINGLE bit is set to 1, SPI_SCS[n] assertion/deassertion controlled by Software. (See Section 22.2.3.6.1) using the MCSPI_CHxCONF[20] FORCE bit.

##### *22.2.3.6.1 Programming Tips When Switching to Another Channel*

When a single channel is enabled and data transfer is ongoing:

- Wait for completion of the SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) before disabling the current channel and enabling a different channel.
- Disable the current channel first, and then enable the other channel.

##### *22.2.3.6.2 Keep $\overline{\text{SPI_SCS[n]}}$ Active Mode (Force $\overline{\text{SPI_SCS[n]}}$)*

Continuous transfers are manually allowed by keeping the $\overline{\text{SPI_SCS[n]}}$ signal active for successive SPI words transfer. Several sequences (configuration/enable/disable of the channel) can be run without deactivating the $\overline{\text{SPI_SCS[n]}}$ line. This mode is supported by all channels and any master sequence can be used (transmit-receive, transmit-only, receive-only).

Keeping the $\overline{\text{SPI_SCS[n]}}$ active mode is supported when:

- A single channel is used (MCSPI_MODULCTRL[Single] bit is set to 1).
- Transfer parameters of the transfer are loaded in the configuration register (MCSPI_CH(I)CONF) in the appropriate channel.

  The state of the $\overline{\text{SPI_SCS[n]}}$ signal is programmable.
  - Writing 1 into the bit FORCE of the register MCSPI_CH(I)CONF drives high the $\overline{\text{SPI_SCS[n]}}$ line when MCSPI_CHCONF(I)[EPOL] is cleared to 0, and drives it low when MCSPI_CHCONF(I)[EPOL] is set.
  - Writing 0 into the bit FORCE of the register MCSPI_CH(I)CONF drives low the $\overline{\text{SPI_SCS[n]}}$ line when MCSPI_CHCONF(I)[EPOL] is cleared to 0, and drives it high when MCSPI_CHCONF(I)[EPOL] is set.
- A single channel is enabled (MCSPI_CH(I)CTRL[En] set to 1) . The first enabled channel activates the $\overline{\text{SPI_SCS[n]}}$ line.


Copyright © 2011–2016, Texas Instruments Incorporated

Once the channel is enabled, the SPI_SCS[n] signal is activated with the programmed polarity.

As in multi-channel master mode, the start of the transfer depends on the status of the transmitter register, the status of the receiver register and the mode defined by the bits TRM in the configuration register (transmit only, receive only or transmit and receive) of the enabled channel.

The status of the serialization completion of each SPI word is given by the bit EOT of the register MCSPI_CH(I)STAT. The bit RX_full in the MCSPI_IRQSTATUS register is set when a received data is loaded from the shift register to the receiver register.

A change in the configuration parameters is propagated directly on the SPI interface. If the SPI_SCS[n] signal is activated the user must insure that the configuration is changed only between SPI words, in order to avoid corrupting the current transfer.

**NOTE:** The SPI_SCS[n] polarity, the SPI_SCLK phase and SPI_SCLK polarity must not be modified when the SPI_SCS[n] signal is activated. The Transmit/Receive mode, programmable with the bit TRM can be modified only when the channel is disabled. The channel can be disabled and enabled while the SPI_SCS[n] signal is activated.

The delay between SPI words that requires the connected SPI slave device to switch from one configuration (Transmit only for instance) to another (receive only for instance) must be handled under software responsibility.

At the end of the last SPI word, the channel must be deactivated (MCSPI_CH(I)CTRL[En] is cleared to 0) and the SPI_SCS[n] can be forced to its inactive state (MCSPI_CH(I)CONF[Force]).

Figure 22-8 and Figure 22-9 show successive transfers with SPI_SCS[n] kept active low with a different configuration for each SPI word in respectively single data pin interface mode and two data pins interface mode. The arrows indicate when the channel is disabled before a change in the configuration parameters and enabled again.

**Figure 22-8. Continuous Transfers With SPI_SCS[n] Maintained Active (Single-Data-Pin Interface Mode)**

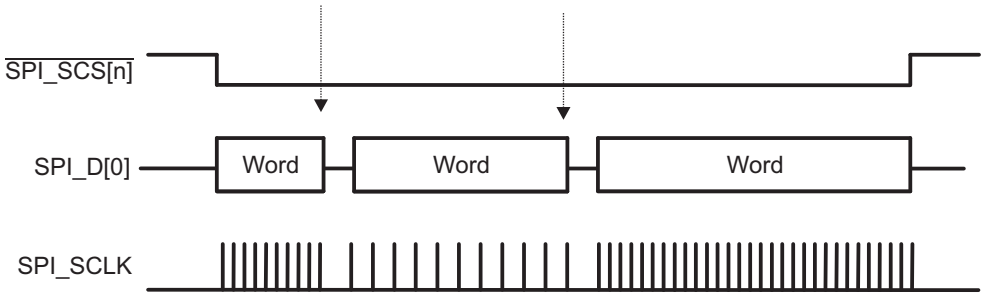
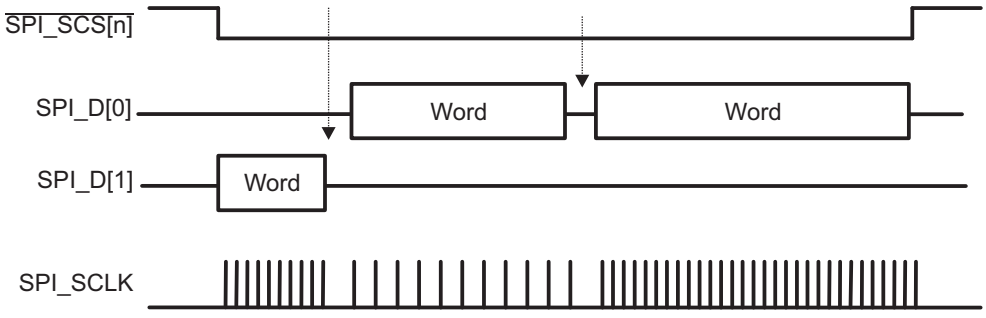



**Figure 22-9. Continuous Transfers With SPI_SCS[n] Maintained Active (Dual-Data-Pin Interface Mode)**





Copyright © 2011–2016, Texas Instruments Incorporated

**NOTE:** The turbo mode is also supported for the Keep $\overline{\text{SPI_SCS[n]}}$ active mode when the following conditions are met:

- A single channel will be explicitly used (MCSPI_MODULCTRL[Single] bit is set to 1).
- The turbo mode is enabled in the configuration of the channel (Turbo bit of the register (i)CONF).

#### *22.2.3.6.3 Turbo Mode*

The purpose of the Turbo mode is to improve the throughput of the SPI interface when a single channel is enabled, by allowing transfers until the shift register and the receiver register are full.

This mode is programmable per channel (bit Turbo of the register (I)CONF). When several channels are enabled, the bit Turbo of the registers MCSPI_CH(I)CONF has no effect, and the channel access to the shift registers remains as described in Section 22.2.3.3.

In Turbo mode, **Rule 1** and **Rule 2** defined in Section 22.2.3.3 are applicable, but Rule 3 is not applicable. An enabled channel can be scheduled if its receive register is full (bit RXS of the register MCSPI_CH(I)STAT) at the time of shift register assignment until the shift register is full.

The receiver register cannot be overwritten in turbo mode. In consequence the RX_overflow bit, in MCSPI_IRQSTATUS register is never set in this mode.


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.2.3.7 Start Bit Mode

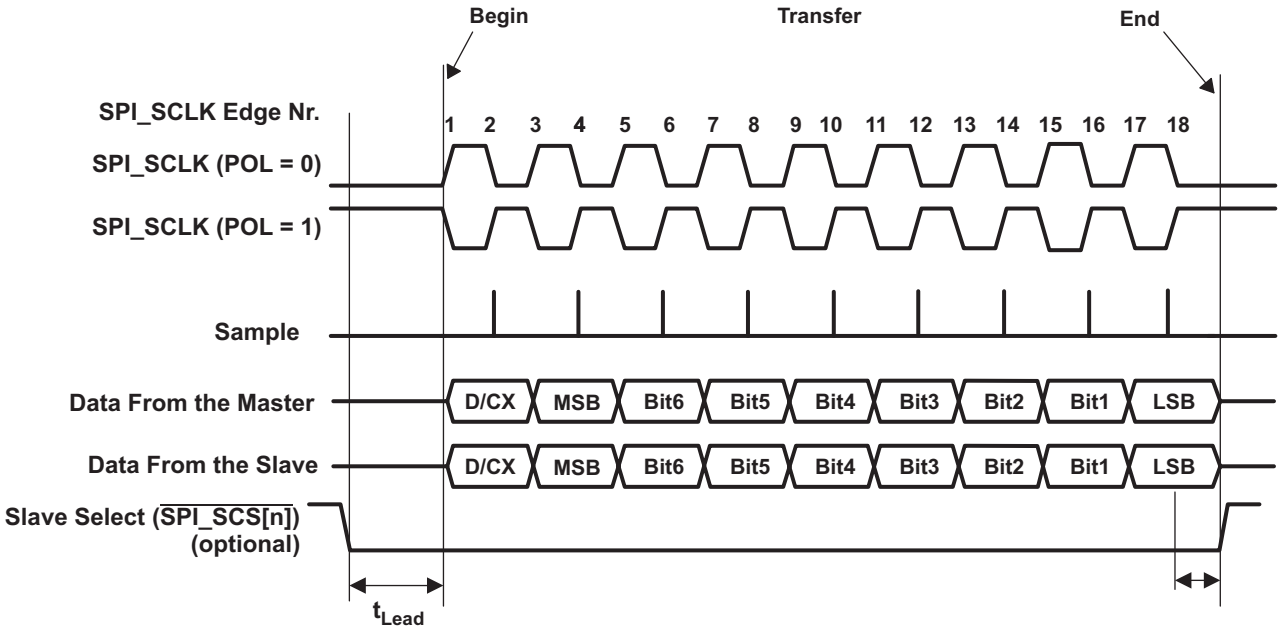
The purpose of the start bit mode is to add an extended bit before the SPI word transmission specified by word length WL (see Figure 22-10). This feature is only available in master mode. This mode is compliant with write command/data format as specified in LoSSI protocol.

This mode is programmable per channel (Start bit enable (SBE) bit of the register MCSPI_CH(I)CONF).

The polarity of the extended bit is programmable per channel and it indicates whether the next SPI word must be handled as a command when SBPOL is cleared to 0 or as a data or a parameter when SBPOL is set to 1. Moreover, start bit polarity SBPOL can be changed dynamically during start bit mode transfer without disabling the channel for reconfiguration, in this case you have the responsibility to configure the SBPOL bit before writing the SPI word to be transmitted in TX register.

The start bit mode could be used at the same time as turbo mode and/or manual chip select mode. In this case, only one channel could be used, no round-robin arbitration is possible.

**Figure 22-10. Extended SPI Transfer With Start Bit PHA = 1**





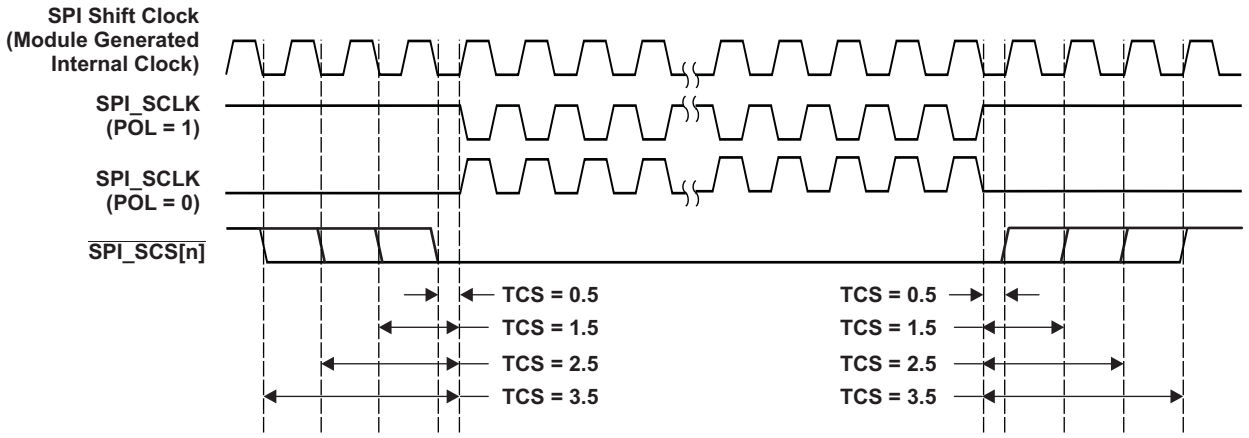
Copyright © 2011–2016, Texas Instruments Incorporated

### 22.2.3.8 Chip-Select Timing Control

The chip select timing control is only available in master mode with automatic chip select generation (FORCE bit field is cleared to 0), to add a programmable delay between chip select assertion and first clock edge or chip select removal and last clock edge. The option is available only in 4 pin mode MCSPI_MODULCTRL[1] PIN34 is cleared to 0.

This mode is programmable per channel (TCS bit of the register MCSPI_CH(I)CONF). Figure 22-11 shows the chip-select $\overline{\text{SPI_SCS[n]}}$ timing control.

**Figure 22-11. Chip-Select $\overline{\text{SPI_SCS[n]}}$ Timing Controls**




**NOTE:** Because of the design implementation for transfers using a clock divider ratio set to 1 (clock bypassed), a half cycle must be added to the value between chip-select assertion and the first clock edge with PHA = 1 or between chip-select removal and the last clock edge with PHA = 0.

With an odd clock divider ratio which occurs when granularity is one clock cycle, that means that MCSPI_CH(I)CONF[CLKG] is set to 1 and MCSPI_CH(I)CONF[CLKD] has an even value, the clock duty cycle is not 50%, then one of the high level or low level duration is selected to be added to TCS delay.

Table 22-3 summarizes all delays between chip select and first (setup) or last (hold) clock edge.

In 3-pin mode this option is useless, the chip select $\overline{\text{SPI_SCS[n]}}$ is forced to low state.

T_ref = CLKSPIREF period in ns

$F_{ratio}$ = SPI clock division ratio

**Table 22-3. Chip Select ↔ Clock Edge Delay Depending on Configuration**

| Clock Ratio $F_{ratio}$ | Clock Phase PHA | Chip Select ↔ Clock Edge Delay | |
|---|---|---|---|
| | | Setup | Hold |
| 1 | 0 | T_ref × (TCS + ½) | T_ref × (TCS + 1) |
| | 1 | T_ref × (TCS + 1) | T_ref × (TCS + ½) |
| Even ≥ 2 | x | T_ref × $F_{ratio}$ × (TCS + ½) | T_ref × $F_{ratio}$ × (TCS + ½) |
| Odd ≥ 3 (only with MCSPI_CH(I)CONF[CLKG] set to 1 | 0 | T_ref × [{$F_{ratio}$ × TCS) + ($F_{ratio}$ + ½)] | T_ref × [{$F_{ratio}$ × TCS) + ($F_{ratio}$ + ½)] |
| | 1 | T_ref × [{$F_{ratio}$ × TCS) + ($F_{ratio}$ - ½)] | T_ref × [{$F_{ratio}$ × TCS) + ($F_{ratio}$ - ½)] |


Copyright © 2011–2016, Texas Instruments Incorporated

The clock divider ratio depends on divider granularity MCSPI_CH(I)CONF[CLKG]:

- MCSPI_CH(I)CONF[CLKG] = 0 : granularity is power of 2.
  $F_{ratio} = 2^{MCSPI_CH(I)CONF[CLKD]}$
- MCSPI_CH(I)CONF[CLKG] = 1 : EXTCLK concatenated with CLKD + 1.
  $F_{ratio}$ = MCSPI_CH(I)CNTRL[EXTCLK].MCSPI_CH(I)CONF[CLKD] + 1

### 22.2.3.9 Clock Ratio Granularity

By default the clock division ratio is defined by the register MCSPI_CH(I)CONF[CLKD] with power of 2 granularity leading to a clock division in range 1 to 32768, in this case the duty cycle is always 50%. With bit MCSPI_CH(I)CONF[CLKG] the clock division granularity can be changed to one clock cycle, in that case the register MCSPI_CH(I)CTRL[EXTCLK] is concatenated with MCSPI_CH(I)CONF[CLKD] to give a 12-bit width division ratio in range 1 to 4096.

When granularity is one clock cycle (MCSPI_CH(I)CONF[CLKG] set to 1), for odd value of clock ratio the clock duty cycle is kept to 50-50 using falling edge of clock reference CLKSPIREF.

**Table 22-4. CLKSPIO High/Low Time Computation**

| Clock Ratio $F_{ratio}$ | CLKSPIO High Time | CLKSPIO Low Time |
|---|---|---|
| 1 | $T_{high}$_ref | $T_{low}$_ref |
| Even ≥ 2 | t_ref × ($F_{ratio}$/2) | t_ref × ($F_{ratio}$/2) |
| Odd ≥ 3 | t_ref × ($F_{ratio}$/2) | t_ref × ($F_{ratio}$/2) |

T_ref = CLKSPIREF period in ns. $T_{high}$_ref = CLKSPIREF high Time period in ns. $T_{low}$_ref = CLKSPIREF low Time period in ns. $F_{ratio}$ = SPI clock division ratio

$F_{ratio}$ = MCSPI_CH(I)CTRL[EXTCLK].MCSPI_CH(I)CONF[CLKD] + 1

For odd ratio value the duty cycle is calculated as:

Duty_cycle = ½

Granularity examples: With a clock source frequency of 48 MHz:

**Table 22-5. Clock Granularity Examples**

| MCSPI_CH(I) CTRL | MCSPI_CH(I) CONF | MCSPI_CH(I) CONF | | MCSPI_CH(I) CONF | MCSPI_CH(I) CONF | Thigh (ns) | Tlow (ns) | Tperiod (ns) | Duty Cycle | Fout (MHz) |
|---|---|---|---|---|---|---|---|---|---|---|
| EXTCLK | CLKD | CLKG | $F_{ratio}$ | PHA | POL | | | | | |
| X | 0 | 0 | 1 | X | X | 10.4 | 10.4 | 20.8 | 50-50 | 48 |
| X | 1 | 0 | 2 | X | X | 20.8 | 20.8 | 41.6 | 50-50 | 24 |
| X | 2 | 0 | 4 | X | X | 41.6 | 41.6 | 83.2 | 50-50 | 12 |
| X | 3 | 0 | 8 | X | X | 83.2 | 83.2 | 166.4 | 50-50 | 6 |
| 0 | 0 | 1 | 1 | X | X | 10.4 | 10.4 | 20.8 | 50-50 | 48 |
| 0 | 1 | 1 | 2 | X | X | 20.8 | 20.8 | 41.6 | 50-50 | 24 |
| 0 | 2 | 1 | 3 | 1 | 0 | 31,2 | 31,2 | 62.4 | 50-50 | 16 |
| 0 | 2 | 1 | 3 | 1 | 1 | 31,2 | 31,2 | 62.4 | 50-50 | 16 |
| 0 | 3 | 1 | 4 | X | X | 41.6 | 41.6 | 83.2 | 50-50 | 12 |
| 5 | 0 | 1 | 81 | 1 | 0 | 842,4 | 842,4 | 1684.8 | 50-50 | 0.592 |
| 5 | 7 | 1 | 88 | X | X | 915.2 | 915.2 | 1830.4 | 50-50 | 0.545 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.3.10 FIFO Buffer Management (Optional USEFIFO = 1)

The SPI controller has a built-in FFNBYTE bytes buffer in order to unload DMA or interrupt handler and improve data throughput. The instantiation of this Buffer is optional it depends on a generic parameter USEFIFO, the FIFO is instantiated when it is set to 1. Allowed FIFO depth up to 64 bytes is supported and is defined by generic parameter FFNBYTE. When the FIFO is not instantiated writes in registers MCSPI_XFERLEVEL, MCSPI_CH(I)CONF[FFER] and MCSPI_CH(I)CONF[FFEW] have no functional effect, nevertheless read back is allowed to check written value.

This buffer can be used by only one channel at once and is selected by setting MCSPI_CH(I)CONF[FFER] or MCSPI_CH(I)CONF[FFEW] to 1.

If several channel are selected and several FIFO enable bit fields set to 1, the controller forces buffer not to be used, it is the responsibility of the driver to set only one FIFO enable bit field.

The buffer can be used in the modes defined:

- Master or Slave mode.
- Transmit only, Receive only or Transmit/Receive mode.
- Single channel or turbo mode, or in normal round robin mode. In round robin mode the buffer is used by only one channel.
- Every word length MCSPI_CH(I)CONF[WL] are supported.

Two levels AEL and AFL located in MCSPI_XFERLEVEL register rule the buffer management. The granularity of these levels is one byte, then it is not aligned with SPI word length. It is the responsibility of the driver to set these values as a multiple of SPI word length defined in MCSPI_CH(I)CONF[WL]. The number of byte written in the FIFO depends on word length (see Table 22-6).

**Table 22-6. FIFO Writes, Word Length Relationship**

| | SPI Word Length WL | | |
|---|---|---|---|
| | **3 ≤ WL ≤ 7** | **8 ≤ WL ≤ 15** | **16 ≤ WL ≤ 31** |
| Number of byte written in the FIFO | 1 byte | 2 bytes | 4 byte |

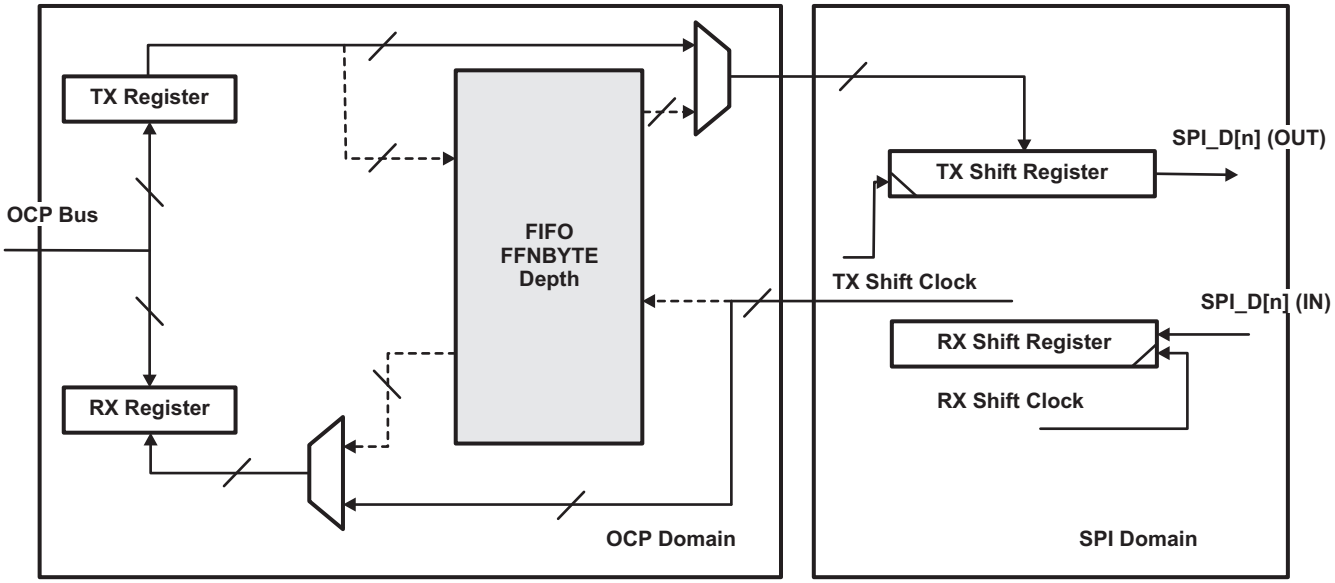
##### *22.2.3.10.1 Split FIFO*

The FIFO can be split into two parts when the module is configured in transmit/receive mode MCSPI_CH(I)CONF[TRM] is cleared to 0 and MCSPI_CH(I)CONF[FFER] and MCSPI_CH(I)CONF[FFEW] asserted. Then system can access a FFNBYTE/2 byte depth FIFO per direction.

The FIFO buffer pointers are reset when the corresponding channel is enabled or FIFO configuration changes.

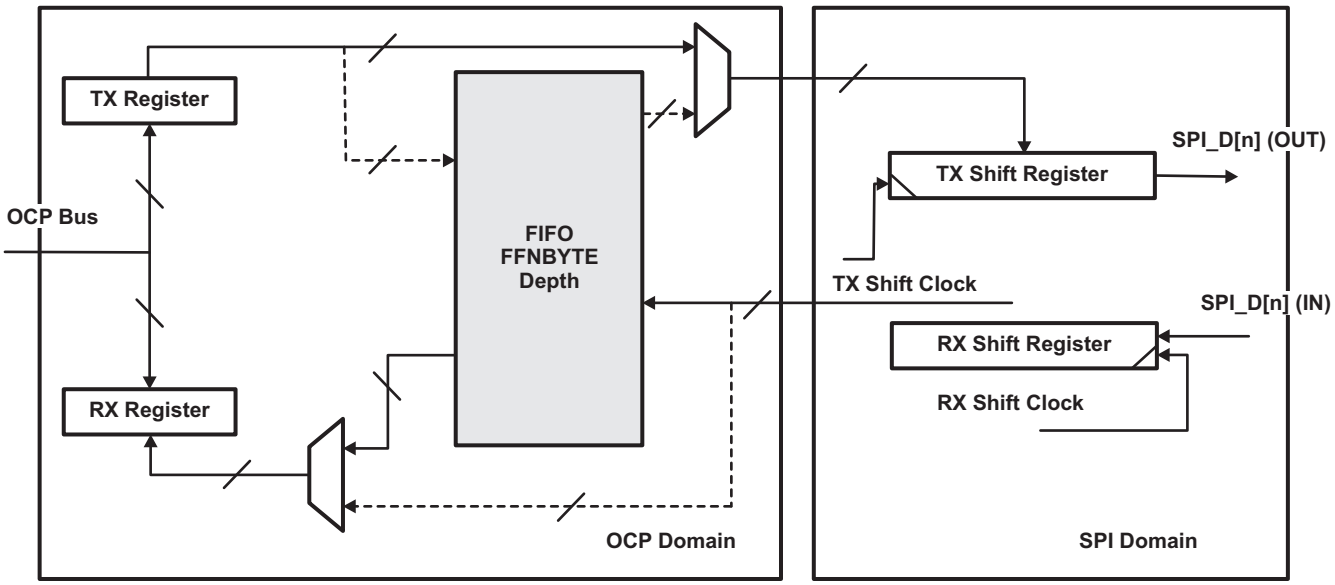

Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 22-12. Transmit/Receive Mode With No FIFO Used**




**Configuration:**
**MCSPI_CH(i)CONF[TRM ]= 0 (Transmit/receive mode)**
**MCSPI_CH(i)CONF[FFER] = 0 (FIFO disabled on receive path)**
**MCSPI_CH(i)CONF[FFEW] = 0 (FIFO disabled on transmit path)**

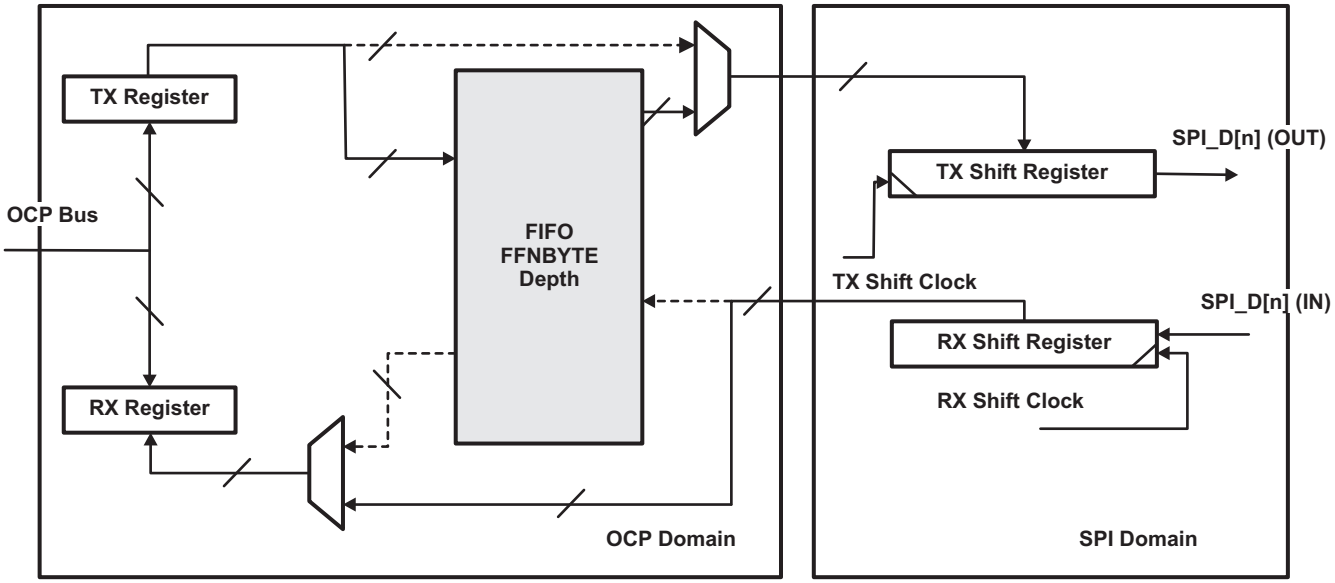
**Figure 22-13. Transmit/Receive Mode With Only Receive FIFO Enabled**




**Configuration:**
**MCSPI_CH(i)CONF[TRM] = 0 (Transmit/receive mode)**
**MCSPI_CH(i)CONF[FFER] = 1 (FIFO enabled on receive path)**
**MCSPI_CH(i)CONF[FFEW] = 0 (FIFO disabled on transmit path)**

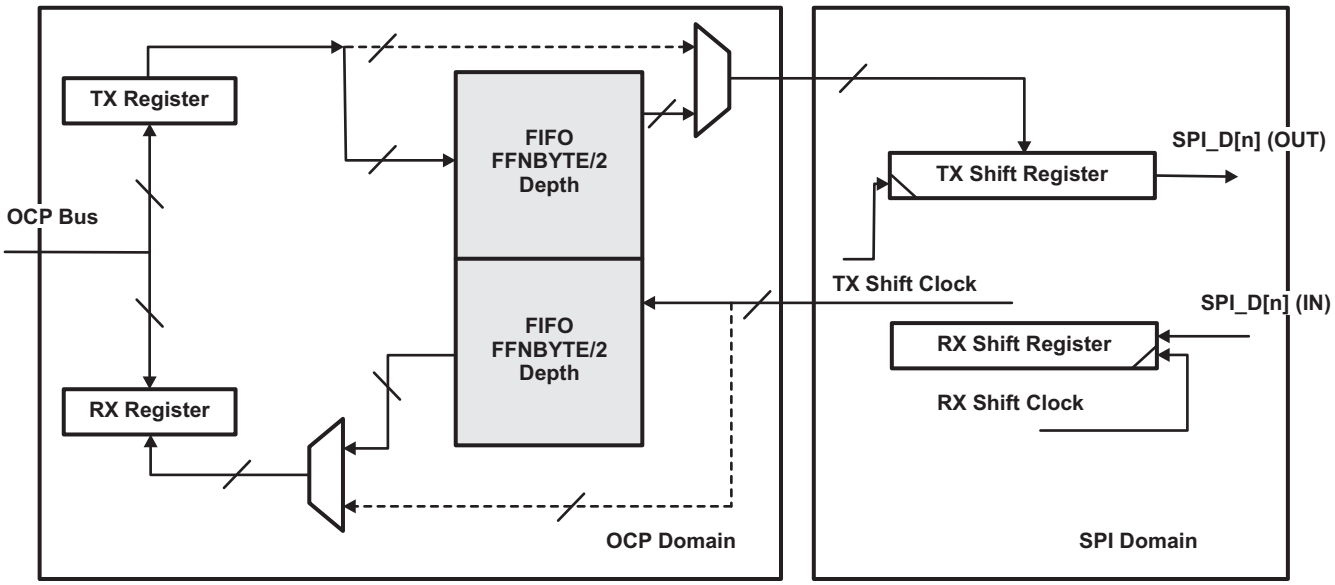

Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 22-14. Transmit/Receive Mode With Only Transmit FIFO Used**




**Configuration:**
**MCSPI_CH(i)CONF[TRM] = 0 (Transmit/receive mode)**
**MCSPI_CH(i)CONF[FFER] = 0 (FIFO disabled on receive path)**
**MCSPI_CH(i)CONF[FFEW] = 1 (FIFO enabled on transmit path)**

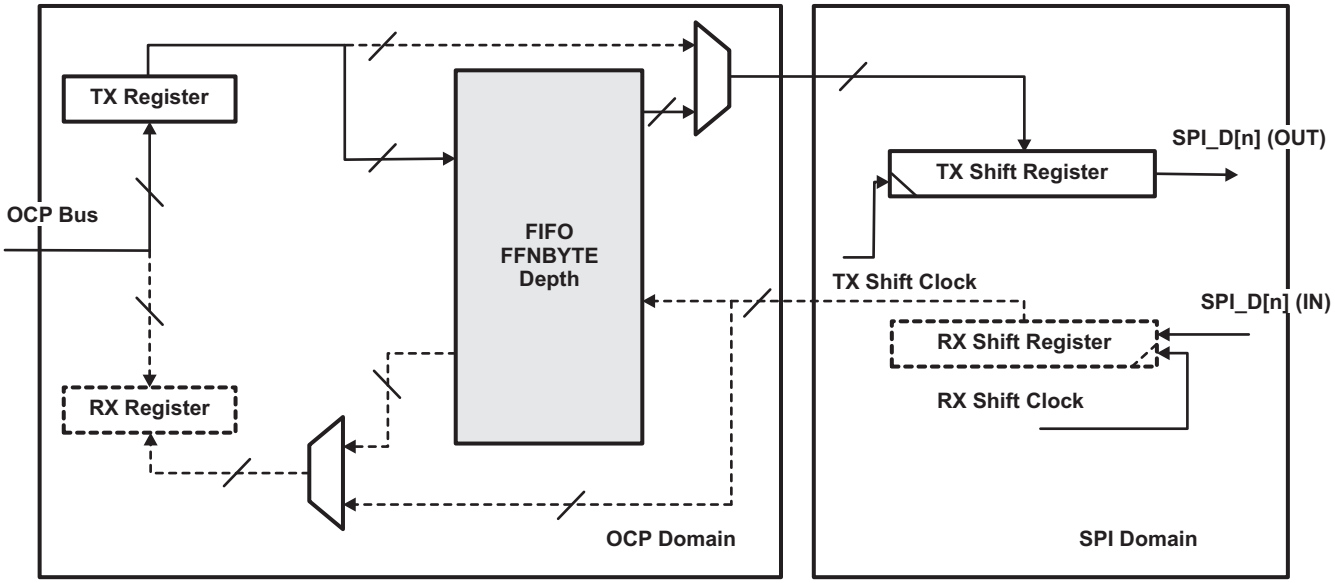
**Figure 22-15. Transmit/Receive Mode With Both FIFO Direction Used**




**Configuration:**
**MCSPI_CH(i)CONF[TRM] = 0 (Transmit/receive mode)**
**MCSPI_CH(i)CONF[FFER] = 1 (FIFO enabled on receive path)**
**MCSPI_CH(i)CONF[FFEW] = 0 (FIFO disabled on transmit path)**

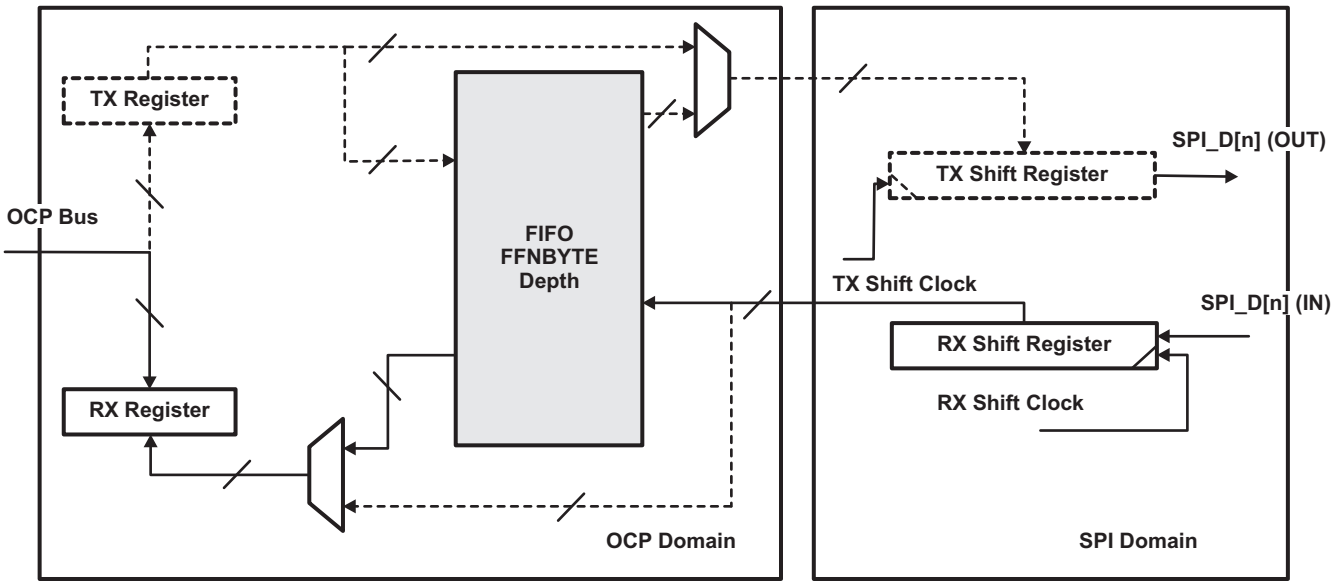

Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 22-16. Transmit-Only Mode With FIFO Used**




**Configuration:**
**MCSPI_CH(i)CONF[TRM] = 2h (Transmit-only mode)**
**MCSPI_CH(i)CONF[FFER] = 1 (FIFO enabled on transmit path)**
**MCSPI_CH(i)CONF[FFEW] (not applicable)**

**Figure 22-17. Receive-Only Mode With FIFO Used**




**Configuration:**
**MCSPI_CH(i)CONF[TRM] = 1 (Receive-only mode)**
**MCSPI_CH(i)CONF[FFER] = 1 (FIFO enabled on receive path)**
**MCSPI_CH(i)CONF[FFEW] (not applicable)**


Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.3.10.2 Buffer Almost Full

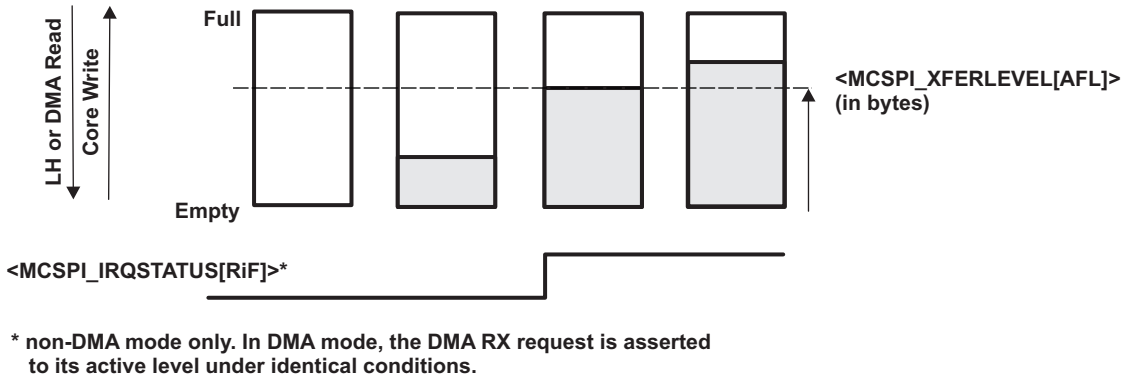
The bit field MCSPI_XFERLEVEL[AFL] is needed when the buffer is used to receive SPI word from a slave (MCSPI_CH(I)CONF[FFER] must be set to 1) and register width depends on the generic parameter FFNBYTE value. It defines the almost full buffer status.

When FIFO pointer reaches this level, an interrupt or a DMA request is sent to the CPU to enable the system to read AFL + 1 bytes from receive register. Be careful, AFL + 1 must correspond to a multiple value of MCSPI_CH(I)CONF[WL].

When DMA is used, the request is de-asserted after the first receive register read.

No new request will be asserted again as long as system has not performed the right number of read accesses.

**Figure 22-18. Buffer Almost Full Level (AFL)**




*** non-DMA mode only. In DMA mode, the DMA RX request is asserted to its active level under identical conditions.**

.

**NOTE:** SPI_IRQSTATUS register bits are not available in DMA mode. In DMA mode, the SPIm_DMA_RX*n* request is asserted on the same conditions than the SPI_IRQSTATUS RX*n*_FULL flag.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.3.10.3 Buffer Almost Empty

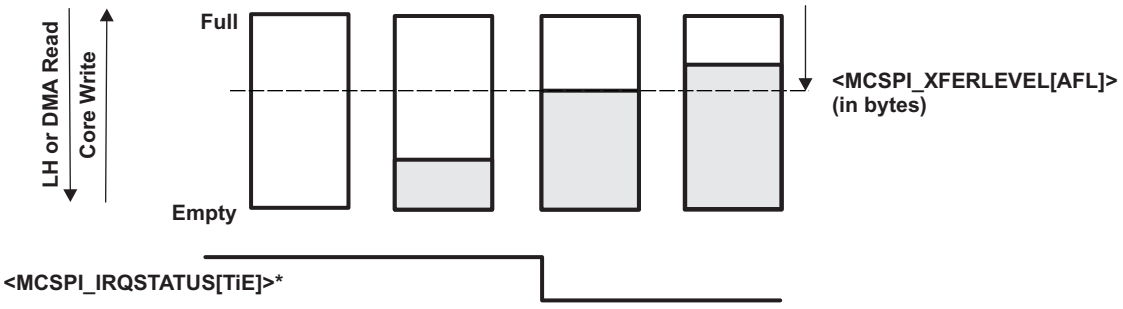
The bit field MCSPI_XFERLEVEL[AEL] is needed when the buffer is used to transmit SPI word to a slave (MCSPI_CH(I)CONF[FFEW] bit must be set to 1) and register width depends on the generic parameter FFNBYTE value. It defines the almost empty buffer status.

When FIFO pointer has not reached this level, an interrupt or a DMA request is sent to the CPU to enable the system to write AEL + 1 bytes to transmit register. Be careful, AEL + 1 must correspond to a multiple value of MCSPI_CH(I)CONF[WL].

When DMA is used, the request is de-asserted after the first transmit register write.

No new request will be asserted again as long as system has not performed the right number of write accesses.

**Figure 22-19. Buffer Almost Empty Level (AEL)**




* non-DMA mode only. In DMA mode, the DMA TX request is asserted to its active level under identical conditions.

.

#### 22.2.3.10.4 End of Transfer Management

When the FIFO buffer is enabled for a channel, the user shall previously configure the MCSPI_XFERLEVEL register the AEL and AFL levels and especially the WCNT bit field to define the number of Spi word to be transferred using the FIFO before enabling the channel.

This counter allows the controller to stop the transfer correctly after a defined number of Spi word transfer. If WNCT is cleared to 0, the counter is not used and the user must stop the transfer manually by disabling the channel, in this case the user doesn't know how many SPI transfers have been done. For receive transfer, software shall poll the corresponding FFE bit field and read the Receive register to empty the FIFO buffer.

When End Of Word count interrupt is generated, the user can disable the channel and poll on MCSPI_CH(I)STAT[FFE] register to know if SPI word is still there in FIFO buffer and read last words.

**NOTE:** If the FIFO buffer is enabled, the module is configured to use the WCNT feature in the master mode, and the software uses the TX_EMPTY interrupt to load data into the FIFO, then the WCNT value must be a multiple of the Almost Empty Level (AEL).

#### 22.2.3.10.5 Multiple SPI Word Access

The CPU has the ability to perform multiple SPI word access to the receive or transmit registers within a single 32-bit OCP access by setting the bit field MCSPI_MODULCTRL[MOA] to 1 under specific conditions:

- The channel selected has the FIFO enable.
- Only FIFO sense enabled support the kind of access.
- The bit field MCSPI_MODULCTRL[MOA] is set to 1.
- Only 32-bit OCP access and data width can be performed to receive or transmit registers, for other


Copyright © 2011–2016, Texas Instruments Incorporated

kind of access the CPU must de-assert MCSPI_MODULCTRL[MOA] bit fields.

- The Level MCSPI_XFERLEVEL[AEL] and MCSPI_XFERLEVEL[AFL] must be 32-bit aligned , it means that AEL[0] = AEL[1] = 1 or AFL[0] = AFL[1] = 1.
- If MCSPI_XFERLEVEL[WCNT] is used it must be configured according to SPI word length.
- The word length of SPI words allows to perform multiple SPI access, that means that MCSPI_CH(I)CONF[WL] < 16

Number of SPI word access depending on SPI word length:

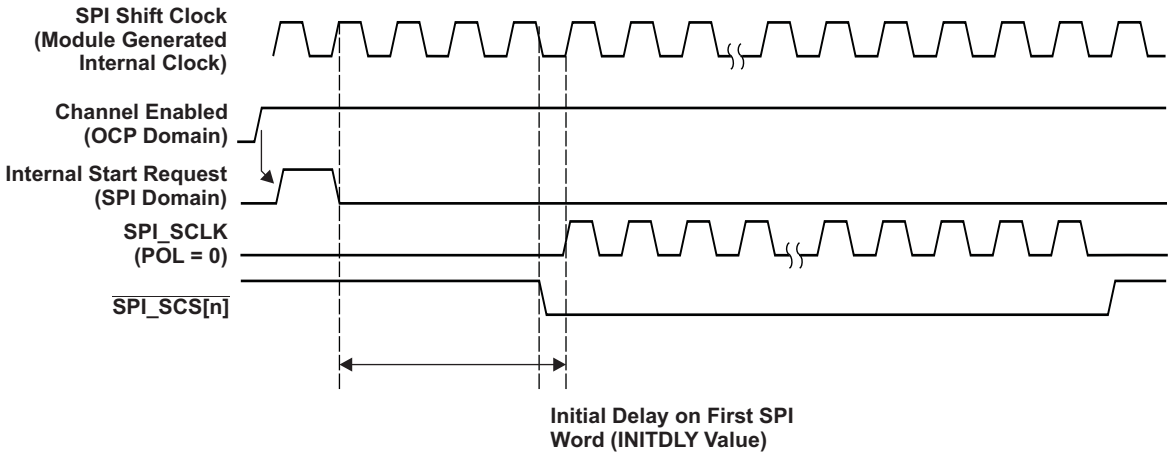
- 3 ≤ WL ≤ 7, SPI word length smaller or equal to byte length, four SPI words accessed per 32-bit OCP read/write. If word count is used (MCSPI_XFERLEVEL[WCNT]), set the bit field to WCNT[0] = WCNT[1] = 0
- 8 ≤ WL ≤ 15, SPI word length greater than byte or equal to 16-bit length, two SPI words accessed per 32-bit OCP read/write. If word count is used (MCSPI_XFERLEVEL[WCNT]), set the bit field to WCNT[0] = 0
- 16 ≤ WL multiple SPI word access not applicable.

#### 22.2.3.11 First SPI Word Delayed

The SPI controller has the ability to delay the first SPI word transfer to give time for system to complete some parallel processes or fill the FIFO in order to improve transfer bandwidth. This delay is applied only on first SPI word after SPI channel enabled and first write in Transmit register. It is based on output clock frequency.

This option is meaningful in master mode and single channel mode MCSPI_MODULECTRL[SINGLE] asserted.

**Figure 22-20. Master Single Channel Initial Delay**




Few delay values are available: No delay, 4/8/16/32 SPI cycles.

Its accuracy is half cycle in clock bypass mode and depends on clock polarity and phase.


Copyright © 2011–2016, Texas Instruments Incorporated

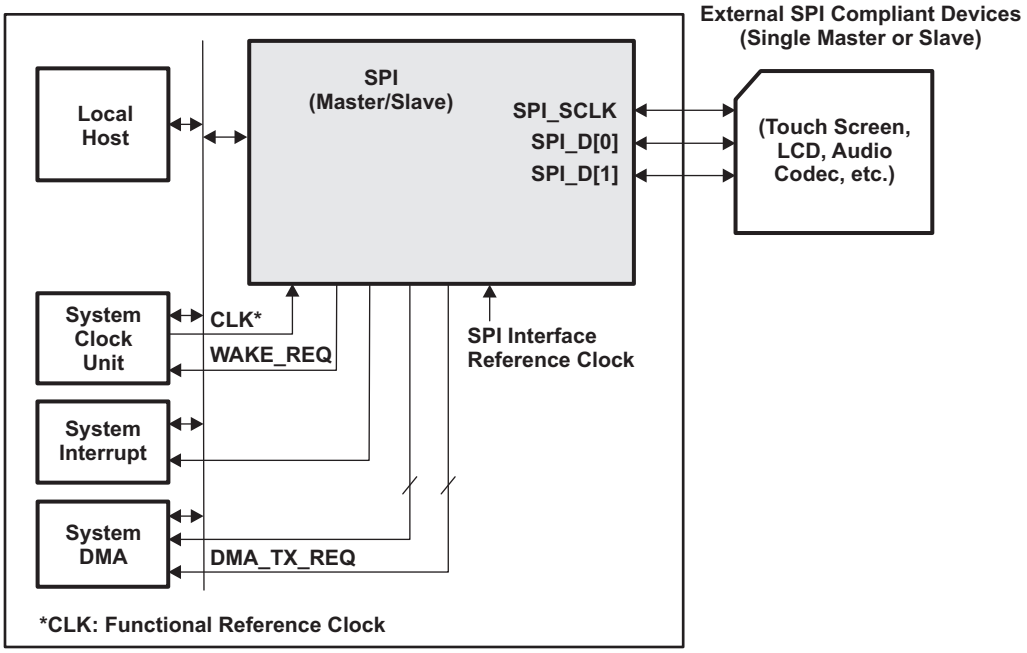
### 22.2.3.12 3-Pin or 4-Pin Mode

External SPI bus interface can be configured to use a restricted set of pin using the bit field MCSPI_MODULCTRL[1] PIN34 and depending on targeted application:

- If MCSPI_MODULECTRL[34] is cleared to 0 (default value), the controller is in 4-pin mode using the SPI pins: SPI_SCLK, SPI_D[0], SPI_D[1], and SPI_SCS[n].
- If MCSPI_MODULECTRL[34] is set to 1, the controller is in 3-pin mode using the SPI pins: SPI_SCLK, SPI_D[0], and SPI_D[1].

In this mode it is mandatory to put the controller in single channel master mode (MCSPI_MODULECTRL[SINGLE] asserted) and to connect only one SPI device on the bus.

**Figure 22-21. 3-Pin Mode System Overview**




In 3-pin mode all options related to chip select management are useless:

- MCSPI_CHxCONF[EPOL]
- MCSPI_CHxCONF[TCS0]
- MCSPI_CHxCONF[FORCE]

The chip select pin $\overline{\text{SPI_SCS[n]}}$ is forced to 0 in this mode.


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.2.4 Slave Mode

SPI is in slave mode when the bit MS of the register MCSPI_MODULCTRL is set.

In slave mode, SPI can be connected to up to 4 external SPI master devices. SPI handles transactions with a single SPI master device at a time.

In slave mode, SPI initiates data transfer on the data lines (SPI_D[1:0]) when it receives an SPI clock (SPI_SCLK) from the external SPI master device.

The controller is able to work with or without a chip select $\overline{\text{SPI_SCS[n]}}$ depending on MCSPI_MODULCTRL[1] PIN34 bit setting. It also support transfers without dead cycle between two successive words.

#### 22.2.4.1 Dedicated Resources

In slave mode, enabling a channel that is not channel 0 has no effect. Only channel 0 can be enabled. The channel 0, in slave mode has the following resources:

- Its own channel enable, programmable with the bit EN of the register MCSPI_CH0CTRL. This channel should be enabled before transmission and reception. Disabling the channel, outside data word transmission, remains under user responsibility.
- Any of the 4 ports $\overline{\text{SPI_SCS[3:0]}}$ can be used as a slave SPI device enable. This is programmable with the SPIENSLV bits of the register MCSPI_CH0CONF.
- Its own transmitter register MCSPI_TX on top of the common shift register. If the transmitter register is empty, the status bit TXS of the register MCSPI_CH0STAT is set. When SPI is selected by an external master (active signal on the $\overline{\text{SPI_SCS[n]}}$ port assigned to channel 0), the transmitter register content of channel0 is always loaded in shift register whether it has been updated or not. The transmitter register should be loaded before SPI is selected by a master.
- Its own receiver register MCSPI_RX on top of the common shift register. If the receiver register is full, the status bit RXS of the register MCSPI_CH0STAT is set.

**NOTE:** The transmitter register and receiver registers of the other channels are not used. Read from or Write in the registers of a channel that is not channel 0 has no effect.

- Its own communication configuration with the following parameters via the register MCSPI_CH0CONF:
  - Transmit/Receive modes, programmable with the bit TRM.
  - Interface mode (Two data pins or Single data pin) and data pins assignment, both programmable with the bits IS and DPE.
  - SPI word length, programmable with the bits WL.
  - $\overline{\text{SPI_SCS[n]}}$ polarity, programmable with the bit EPOL.
  - SPI_SCLK polarity, programmable with the bit POL.
  - SPI_SCLK phase, programmable with the bit PHA.
  - Use a FIFO buffer or not, programmable with FFER and FFEW, depending on transfer mode TRM.

The SPI_SCLK frequency of a transfer is controlled by the external SPI master connected to SPI. The bits CLKD0 of the register 0CONF are not used in slave mode.

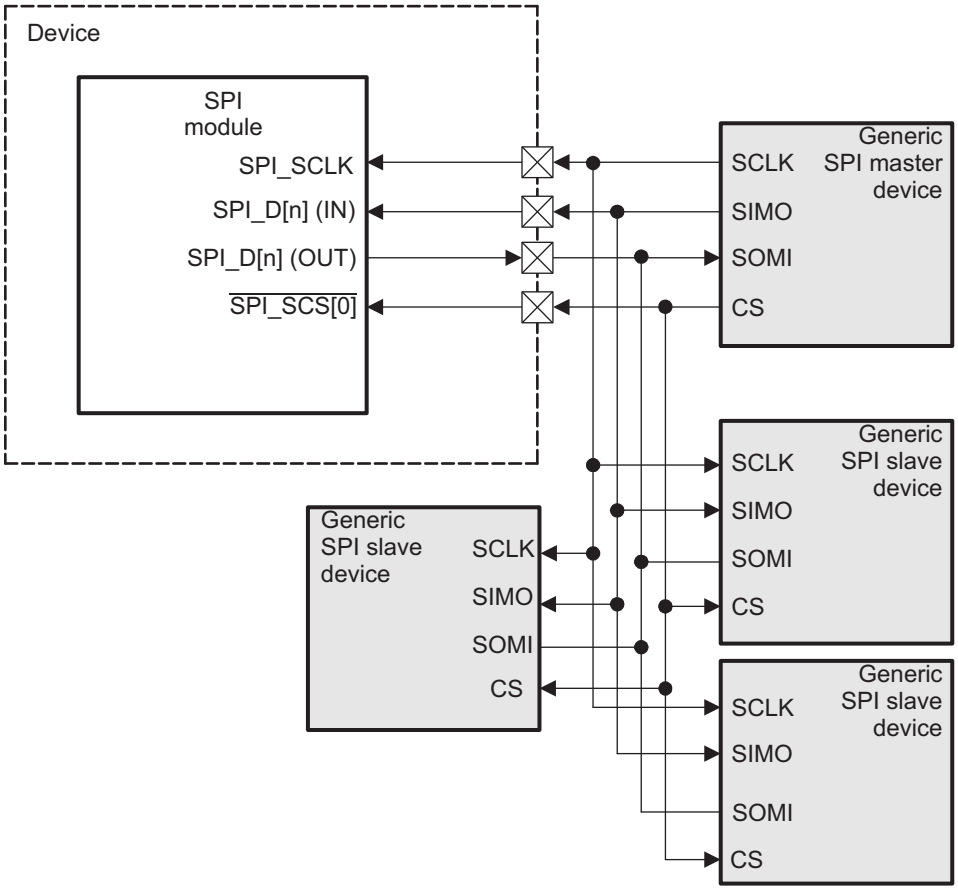
**NOTE:** The configuration of the channel can be loaded in the 0CONF register only when the channel is disabled.

- Two DMA requests events, read and write, to synchronize read/write accesses of the DMA controller with the activity of SPI. The DMA requests are enabled with the bits DMAR and DMAW of the register 0CONF.
- Four interrupts events.

Figure 22-22 shows an example of four slaves wired on a single master device.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 22-22. Example of SPI Slave with One Master and Multiple Slave Devices on Channel 0**




### 22.2.4.2 Interrupt Events in Slave Mode

The interrupt events related to the transmitter register state are TX_empty and TX_underflow. The interrupt events related to the receiver register state are RX_full and RX_overflow.

#### *22.2.4.2.1 TX_EMPTY*

The event TX_empty is activated when the channel is enabled and its transmitter register becomes empty. Enabling channel automatically raises this event. When FIFO buffer is enabled (MCSPI_CH(I)CONF[FFEW] set to 1), the TX_empty is asserted as soon as there is enough space in buffer to write a number of byte defined by MCSPI_XFERLEVEL[AEL].

Transmitter register must be load to remove source of interrupt and TX_empty interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

When FIFO is enabled, no new TX_empty event will be asserted as soon as Local Host has not performed the number of write into transmitter register defined by MCSPI_XFERLEVEL[AEL]. It is the responsibility of Local Host to perform the right number of writes.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.4.2.2 TX_UNDERFLOW

The event TX_underflow is activated when channel is enabled and if the transmitter register or FIFO (if use of buffer is enabled) is empty (not updated with new data) when an external master device starts a data transfer with SPI (transmit and receive).

When FIFO is enabled the data emitted while underflow event is raised is not the last data written in the FIFO.

The TX_underflow indicates an error (data loss) in slave mode.

To avoid having TX_underflow event at the beginning of a transmission, the event TX_underflow is not activated when no data has been loaded into the transmitter register since channel has been enabled.

TX_underflow interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

#### 22.2.4.2.3 RX_FULL

The event RX_full is activated when channel is enabled and receiver becomes filled (transient event). When FIFO buffer is enabled (MCSPI_CH(I)CONF[FFER] set to 1), the RX_full is asserted as soon as there is a number of bytes holds in buffer to read defined by MCSPI_XFERLEVEL[AFL].

Receiver register must be read to remove source of interrupt and RX_full interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

When FIFO is enabled, no new RX_full event will be asserted as soon as Local Host has not performed the number of read into receive register defined by MCSPI_XFERLEVEL[AFL]. It is the responsibility of Local Host to perform the right number of reads.

#### 22.2.4.2.4 RX_OVERFLOW

The RX0_OVERFLOW event is activated in slave mode in either transmit-and-receive or receive-only mode, when a channel is enabled and the SPI_RX*n* register or FIFO is full when a new SPI word is received. The SPI_RX*n* register is always overwritten with the new SPI word. If the FIFO is enabled data within the FIFO are overwritten, it must be considered as corrupted. The RX0_OVERFLOW event should not appear in slave mode using the FIFO.

The RX0_OVERFLOW indicates an error (data loss) in slave mode.

The SPI_IRQSTATUS[3] RX0_OVERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

#### 22.2.4.2.5 End of Word Count

The event EOW (End Of Word count) is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfer defined in MCSPI_XFERLEVEL[WCNT] register. If the value was programmed to 0000h, the counter is not enabled and this interrupt is not generated.

The End of Word count interrupt also indicates that the SPI transfer is halt on channel using the FIFO buffer as soon as MCSPI_XFERLEVEL[WCNT] is not reloaded and channel re-enabled.

End of Word interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

### 22.2.4.3 Slave Transmit-and-Receive Mode

The slave transmit and receive mode is programmable (bits TRM cleared to 0 in the register MCSPI_CH(I)CONF).

After the channel is enabled, transmission and reception proceed with interrupt and DMA request events.

In slave transmit and receive mode, transmitter register should be loaded before SPI is selected by an external SPI master device.


Copyright © 2011–2016, Texas Instruments Incorporated

Transmitter register or FIFO (if use of buffer enabled) content is always loaded in shift register whether it has been updated or not. The event TX_underflow is activated accordingly, and does not prevent transmission.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) the received data is transferred to the channel receive register. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured in one data direction Transmit or Receive, then the FIFO is seen as a unique FFNBYTE bytes buffer, or it can also be configured in both data direction Transmit and Receive, then the FIFO is split into two separate FFNBYTE/2 bytes buffer with their own address space management, in this last case the definition of AEL and AFL levels is based on FFNBYTE bytes and is under Local Host responsibility.

#### 22.2.4.4 Slave Receive-Only Mode

The slave receive mode is programmable (bits TRM set to 01 in the register (I)CONF).

In receive only mode, the Transmitter register should be loaded before SPI is selected by an external SPI master device. Transmitter register or FIFO (if use of buffer enabled) content is always loaded in shift register whether it has been updated or not. The event TX_underflow is activated accordingly, and does not prevent transmission.

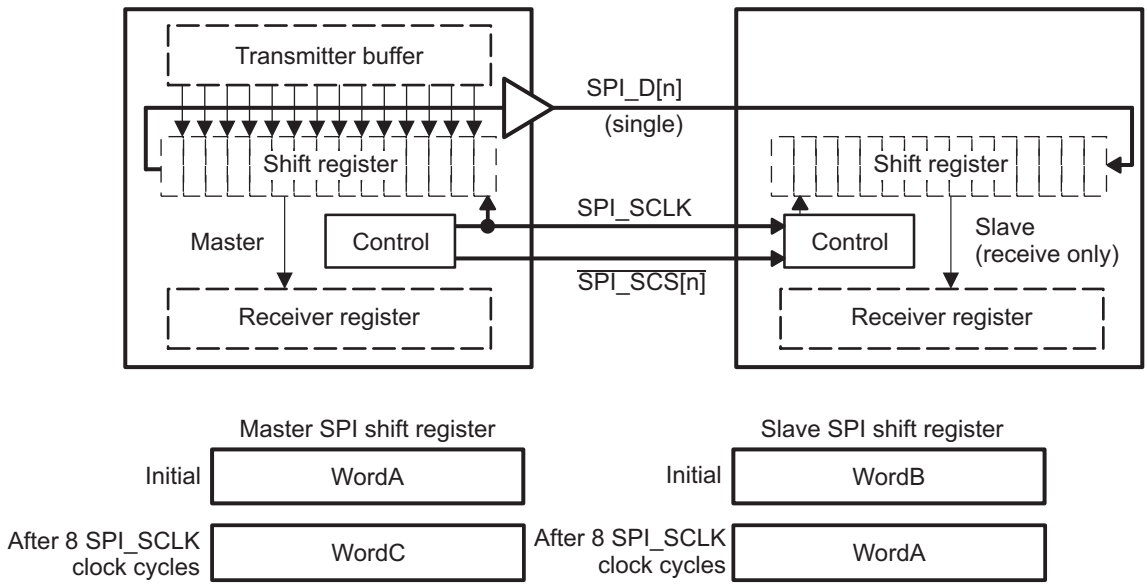
When an SPI word transfer completes (the SPIm.SPI_CH*n*STAT0[2] EOT bit (with *n* = 0) is set to 1), the received data is transferred to the channel receive register.

To use SPI as a slave receive only device with MCSPI_CH(I)CONF[TRM] = 00, the user has the responsibility of the inhibition of the TX_empty and TX_underflow interrupts and DMA write requests due to the transmitter register sate.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) the received data is transferred to the channel receive register. This bit is meaningless when using the Buffer for this channel. The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique FFNBYTE bytes buffer.

Figure 22-23 shows an example of a half-duplex system with a master device on the left and a receive-only slave device on the right. Each time a bit transfers out from the master, 1 bit transfers in from the slave. After eight cycles of the serial clock spim_clk, Word A transfers from the master to the slave.

**Figure 22-23. SPI Half-Duplex Transmission (Receive-Only Slave)**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.4.5 Slave Transmit-Only Mode

The slave transmit only mode is programmable (TRM bit set to 10 in the register MCSPI_CH(I)CONF). This mode avoids the CPU to read the receiver register (minimizing data movement) when only transmission is meaningful.

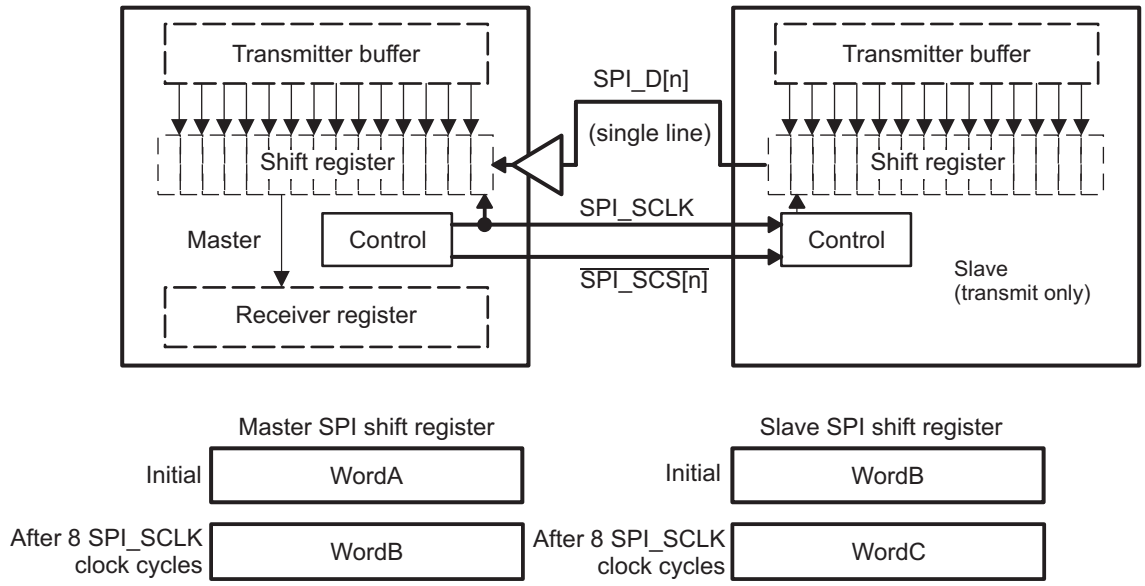
To use SPI as a slave transmit only device with (I)CONF[TRM] = 00, the user should inhibit the RX_full and RX_overflow interrupts and DMA read requests due to the receiver register sate.

On completion of SPI word transfer the bit EOT of the register MCSPI_CH(I)STAT is set. This bit is meaningless when using the buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique FFNBYTE bytes buffer.

Figure 22-24 shows a half-duplex system with a master device on the left and a transmit-only slave device on the right. Each time a bit transfers out from the slave device, 1 bit transfers in the master. After eight cycles of the serial clock spim_clk, WordB transfers from the slave to the master.

**Figure 22-24. SPI Half-Duplex Transmission (Transmit-Only Slave)**




### *22.2.5 Interrupts*

According to its transmitter register state and its receiver register state each channel can issue interrupt events if they are enabled.

The interrupt events are listed in the Section 22.2.3.2 and in Section 22.2.4.2.

Each interrupt event has a status bit, in the MCSPI_IRQSTATUS register, which indicates service is required, and an interrupt enable bit, in the MCSPI_IRQENABLE register, which enables the status to generate hardware interrupt requests.

When an interrupt occurs and later a mask is applied on it (IRQENABLE), the interrupt line is not asserted again even if the interrupt source has not been serviced.

SPI supports interrupt driven operation and polling.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 22.2.5.1 Interrupt-Driven Operation

Alternatively, an interrupt enable bit, in the MCSPI_IRQENABLE register, can be set to enable each of the events to generate interrupt requests when the corresponding event occurs. Status bits are automatically set by hardware logic conditions.

When an event occurs (the single interrupt line is asserted), the CPU must:

- Read the MCSPI_IRQSTATUS register to identify which event occurred,
- Read the receiver register that corresponds to the event, to remove the source of an RX_full event, or Write into the transmitter register that corresponds to the event, to remove the source of a TX_empty event. No action is needed to remove the source of the events TX_underflow and RX_overflow.
- Write a 1 into the corresponding bit of MCSPI_IRQSTATUS register to clear the interrupt status, and release the interrupt line.

The interrupt status bit should always be reset after channel enabling and before event are enabled as interrupt source.

#### 22.2.5.2 Polling

When the interrupt capability of an event is disabled in the MCSPI_IRQENABLE register, the interrupt line is not asserted and:

- The status bits in the MCSPI_IRQSTATUS register can be polled by software to detect when the corresponding event occurs.
- Once the expected event occurs, CPU must: Read the receiver register that corresponds to the event, to remove the source of an RX_full event, or write into the transmitter register that corresponds to the event, to remove the source of a TX_empty event. No action is needed to remove the source of the events TX_underflow and RX_overflow.
- Writing a 1 into the corresponding bit of MCSPI_IRQSTATUS register clears the interrupt status and does not affect the interrupt line state.

### *22.2.6 DMA Requests*

SPI can be interfaced with a DMA controller. At system level, the advantage is to discharge the local host of the data transfers.

According to its transmitter register state, its receiver register state or FIFO level (if use of buffer for the channel) each channel can issue DMA requests if they are enabled.

The DMA requests need to be disabled in order to get TX and RX interrupts, in order to define either the end of the transfer or the transfer of the last words for the modes listed below:

- Master Transmit On
- Master normal receive only mode
- Master turbo receive only mode
- Slave Transmit Only

There are 2 DMA request lines per channel. The management of DMA requests differ according to use of FIFO buffer or not:

#### 22.2.6.1 FIFO Buffer Disabled

The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. DMA Read request can be individually masked with the bit DMAR of the register MCSPI_CH(I)CONF. The DMA Read request line is de-asserted (OCP compliant) on read completion of the receive register of the channel.

The DMA Write request line is asserted when the channel is enabled and the transmitter register of the channel is empty. DMA Write request can be individually masked with the bit DMAW of the register MCSPI_CH(I)CONF. The DMA Write request line is deasserted (OCP compliant) on load completion of the transmitter register of the channel.


Copyright © 2011–2016, Texas Instruments Incorporated

Only one SPI word can be transmitted/received per OCP bus access to write/read the transmit or receive register.

#### 22.2.6.2 FIFO Buffer Enabled

The DMA Read request line is asserted when the channel is enabled and a number of bytes defined in MCSPI_XFERLEVEL[AFL] bit field is hold in FIFO buffer for the receive register of the channel. DMA Read request can be individually masked with the bit DMAR of the register MCSPI_CH(I)CONF. The DMA Read request line is de-asserted (OCP compliant) on the first SPI word read completion of the receive register of the channel. No new DMA request will be asserted again as soon as user has not performed the right number of read accesses defined by MCSPI_XFERLEVEL[AFL] it is under user responsibility.

The DMA Write request line is asserted when the channel is enabled and the number of bytes hold in FIFO buffer is below the level defined by the MCSPI_XFERLEVEL[AEL] bit field. DMA Write request can be individually masked with the bit DMAW of the register MCSPI_CH(I)CONF. The DMA Write request line is de-asserted (OCP compliant) on load completion of the first SPI word in the transmitter register of the channel. No new DMA request will be asserted again as soon as user has not performed the right number of write accesses defined by MCSPI_XFERLEVEL[AEL] it is under user responsibility.

Only one SPI word can be transmitted/received per OCP bus access to write/read the transmit or receive FIFO.

#### 22.2.6.3 DMA 256 Bit Aligned Addresses

The controller has two registers, MCSPI_DAFTX and MCSPI_DAFRX, used only with an enabled channel that manages the FIFO to be compliant the a DMA handler providing only 256-bit aligned addresses.

This features is activated when the bit field MCSPI_MODULCTRL[8] FDAA is set to 1 and only one enabled channel has its bit field MCSPI_CH(I)CONF[27] FFE(I)W or MCSPI_CH(I)CONF[28] FFE(I)R enabled.

In this case, the registers MCSPI_TX(I) and MCSPI_RX(I) are not used and data are managed through registers MCSPI_DAFTX and MCSPI_DAFRX.

### *22.2.7 Emulation Mode*

The MReqDebug input differentiates a regular access of a processor (application access), from an emulator access.

Application access: MReqDebug = 0

In functional mode, the consequences of a read of a receiver register MCSPI_RX(i) are the following:

- The source of an RXi_Full event in the MCSPI_IRQSTATUS register is removed, if it was enabled in the MCSPI_IRQENABLE register.
- The RXiS status bit in the MCSPI_IRQSTATUS register is cleared.
- In master mode, depending on the round robin arbitration, and the transmitter register state, the channel may access to the shift register for transmission/reception.

Emulator access: MReqDebug = 1

In emulation mode, SPI behavior is the same as in functional mode but a read of a receiver register MCSPI_RX(i) is not intrusive:

- MCSPI_RX(i) is still considered as not read. When the FIFO buffer is enabled, pointers are not updated.
- The source of an RXi_Full event in the MCSPI_IRQSTATUS register is not removed. The RXiS status bit in the MCSPI_CH(i)STAT register is held steady.

In emulation mode, as in functional mode, based on the ongoing data transfers, the status bits of the MCSPI_CH(i)STAT register may be optionally updated, the interrupt and DMA request lines may be optionally asserted.


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.2.8 Power Saving Management

Independently of the module operational modes (Transmit and/or Receive), two modes of operations are defined from a power management perspective: normal and idle modes.

The two modes are temporally fully exclusive.

SPI is compliant with the profile “IdleReq / SIdleAck / Swakup” in the Idle mode.

#### 22.2.8.1 Normal Mode

Both the OCP clock and SPI clock (CLKSPIREF) provided to SPI must be active for both master and slave modes. The auto-gating of the module OCP clock and SPI clock occurs when the following conditions are met:

- The bit AutoIdle of the register MCSPI_SYSCONFIG is set.
- In master mode, there is no data to transmit or receive in all channels.
- In slave mode, the SPI is not selected by the external SPI master device and no OCP accesses.

Autogating of the module OCP clock and SPI clock stops when the following conditions are met:

- In master mode, an OCP access occurs.
- In slave mode, an OCP access occurs or SPI is selected by an external SPI master device.

#### 22.2.8.2 Idle Mode

The OCP clock and SPI clock provided to SPI may be switched off on system power manager request and switched back on module request.

SPI is compliant with the power management handshaking protocol: idle request from the system power manager, idle acknowledgment from SPI.

The idle acknowledgment in response to an idle request from the system power manager varies according to a programmable mode in the MCSPI_SYSCONFIG register: No idle mode, force idle mode, and smart idle mode.

- When programmed for no idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to 01), the module ignores the system power manager request, and behaves normally, as if the request was not asserted.
- When programmed for smart idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to 10), the module acknowledges the system power manager request according to its internal state.
- When programmed for force idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is cleared to 00), the module acknowledges the system power manager request unconditionally.

The OCP clock will be optionally switched off, during the smart idle mode period, if the bit ClockActivity of the register MCSPI_SYSCONFIG is set.

The SPI clock will be optionally switched off, during the smart idle mode period, if the second bit ClockActivity of the register MCSPI_SYSCONFIG is set.

SPI assumes that both clocks may be switched off whatever the value set in the field ClockActivity of the register MCSPI_SYSCONFIG.

##### 22.2.8.2.1 Transitions from Normal Mode to Smart-Idle Mode

The module detects an idle request when the synchronous signal IdleReq is asserted.

When IdleReq is asserted, any access to the module will generate an error as long as the OCP clock is alive.

When configured as a slave device, SPI responds to the idle request by asserting the SIdleAck signal (idle acknowledgement) only after completion of the current transfer ($\overline{\text{SPI_SCS[n]}}$) slave selection signal deasserted by the external master) and if interrupt or DMA request lines are not asserted.


Copyright © 2011–2016, Texas Instruments Incorporated

As a master device, SPI responds to the idle request by asserting the SIdleAck signal (idle acknowledgement) only after completion of all the channel data transfers and if interrupt or DMA request lines are not asserted.

As long as SIdleAck is not asserted, if an event occurs, the module can still generate an interrupt or a DMA request after IdleReq assertion. In this case, the module ignores the idle request and SIdleAck will not get asserted: The system power manager will abort the power mode transition procedure. It is then the responsibility of the system to de-assert IdleReq before attempting to access the module.

When SIdleAck is asserted, the module does not assert any new interrupt or DMA request.

#### 22.2.8.2.2 Transition From Smart-Idle Mode to Normal mode

SPI detects the end of the idle period when the idle request signal (IdleReq) is deasserted.

Upon IdleReq de-assertion, the module switches back to normal mode and de-asserts SIdleAck signal. The module is fully operational.

#### 22.2.8.2.3 Force-Idle Mode

Force-idle mode is enabled as follows:

- The bit SIdleMode of the register MCSPI_SYSCONFIG is cleared to 00 (Force Idle).

  The force idle mode is an idle mode where SPI responds unconditionally to the idle request by asserting the SIdleAck signal and by deasserting unconditionally the interrupt and DMA request lines if asserted.

  The transition from normal mode to idle mode does not affect the interrupt event bits of the MCSPI_IRQSTATUS register.

  In force-idle mode, the module is supposed to be disabled at that time, so the interrupt and DMA request lines are likely deasserted. OCP clock and SPI clock provided to SPI can be switched off.

  An idle request during an SPI data transfer can lead to an unexpected and unpredictable result, and is under software responsibility.

  Any access to the module in force idle mode will generate an error as long as the OCP clock is alive and IdleReq is asserted.

The module exits the force idle mode when:

- The idle request signal (IdleReq) is de-asserted.

  Upon IdleReq de-assertion, the module switches back to normal mode and de-asserts SIdleAck signal. The module is fully operational. The interrupt and DMA request lines are optionally asserted a clock cycle later.

### 22.2.9 System Test Mode

SPI is in system test mode (SYSTEST) when the bit System_Test of the register MCSPI_MODULCTRL is set.

The SYSTEST mode is used to check in a very simple manner the correctness of the system interconnect either internally to interrupt handler, or power manager, or externally to SPI I/Os.

I/O verification can be performed in SYSTEST mode by toggling the outputs and capturing the logic state of the inputs. (See MCSPI_SYST register definition in Section 22.3.10)


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.2.10 Reset

#### 22.2.10.1 Internal Reset Monitoring

The module is reset by the hardware when an active-low reset signal, synchronous to the OCP interface clock is asserted on the input pin RESETN.

This hardware reset signal has a global reset action on the module. All configuration registers and all state machines are reset, in all clock domains.

Additionally, the module can be reset by software through the bit SoftReset of the register MCSPI_SYSCONFIG. This bit has exactly the same action on the module logic as the hardware RESETN signal. The register MCSPI_SYSCONFIG is not sensitive to software reset. The SoftReset control bit is active high. The bit is automatically reset to 0 by the hardware.

A global ResetDone status bit is provided in the status register MCSPI_SYSSTATUS. This bit is set to 1 when all the different clock domains resets (OCP domain and SPI domains) have been released (logical AND).

The global ResetDone status bit can be monitored by the software to check if the module is ready-to-use following a reset (either hardware or software).

The clock CLKSPIREF must be provided to the module, in order to allow the ResetDone status bit to be set.

When used in slave mode, the clock CLKSPIREF is needed only during the reset phase. The clock CLKSPIREF can be switched off after the ResetDone status is set.

#### 22.2.10.2 Reset values of registers

The reset values of registers and signals are described in Section 22.3.

### 22.2.11 Access to Data Registers

This section details the supported data accesses (read or write) from/to the data receiver registers MCSPI_RX(i) and data transmitter registers MCSPI_TX(i).

Supported access:

SPI supports only one SPI word per register (receiver or transmitter) and does not support successive 8-bit or 16-bit accesses for a single SPI word.

The SPI word received is always right justified on LSB of the 32-bit register MCSPI_RX(i), and the SPI word to transmit is always right justified on LSB of the 32-bit register MCSPI_TX(i).

The upper bits, above SPI word length, are ignored and the content of the data registers is not reset between the SPI data transfers.

The coherence between the number of bits of the SPI Word, the number of bits of the access and the enabled byte remains under the user's responsibility. Only aligned accesses are supported.

In Master mode, data should not be written in the transmit register when the channel is disabled.


Copyright © 2011–2016, Texas Instruments Incorporated

## 22.2.12 Programming Aid

### 22.2.12.1 Module Initialization

- Hard or soft reset.
- Read MCSPI_SYSSTATUS.
- Check if reset is done.
- Module configuration: (a) Write into MCSPI_MODULCTRL (b) Write into MCSPI_SYSCONFIG.
- Before the ResetDone bit is set, the clocks CLK and CLKSPIREF must be provided to the module.
- To avoid hazardous behavior, it is advised to reset the module before changing from MASTER mode to SLAVE mode or from SLAVE mode to MASTER mode.

### 22.2.12.2 SPI Operational Modes Configuration

The selection of the working mode is done through the MCSPI_CHxCONF register (where x = 0, 1, 2 and 3). Table 22-7 through Table 22-9 list the possible operating modes and their configurations.

**Table 22-7. SPI Receive Mode Initialization**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Set receive mode. | MCSPI_CHxCONF[13:12] TRM | 0x1 |
| Set the word length. | MCSPI_CHxCONF[11:7] WL | 0x8 |
| Clock initialization and channel enabling | MCSPI_MODULCTRL[2] MS<br>MCSPI_CHxCTRL[0] EN | 0x0<br>0x1 |
| Channels activated low during active state | MCSPI_CHxCONF[6] EPOL | 0x1 |
| Clock held high during active state | MCSPI_CHxCONF[1] POL | 0x0 |
| Data latched on odd-numbered edges of the SPI clock | MCSPI_CHxCONF[0] PHA | 0x0 |
| Reset the status bits. | MCSPI_IRQSTATUS | 0x0 |

**Table 22-8. SPI Transmit Mode Initialization**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Set transmit mode. | MCSPI_CHxCONF[13:12] TRM | 0x2 |
| Set the word length. | MCSPI_CHxCONF[11:7] WL | 0x8 |
| Clock initialization and channel enabling | MCSPI_MODULCTRL[2] MS<br>MCSPI_CHxCTRL[0] EN | 0x0<br>0x1 |
| Channels activated low during active state | MCSPI_CHxCONF[6] EPOL | 0x1 |
| Clock held high during active state | MCSPI_CHxCONF[1] POL | 0x0 |
| Data latched on odd-numbered edges of the SPI clock | MCSPI_CHxCONF[0] PHA | 0x0 |
| Reset the status bits. | MCSPI_IRQSTATUS | 0x0 |

**Table 22-9. SPI Transmit-and-Receive Mode Initialization**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Set transmit and receive mode. | MCSPI_CHxCONF[13:12] TRM | 0x0 |
| Set the word length. | MCSPI_CHxCONF[11:7] WL | 0x8 |
| Clock initialization and channel enabling | MCSPI_MODULCTRL[2] MS<br>MCSPI_CHxCTRL[0] EN | 0x0<br>0x1 |
| Channels activated low during active state | MCSPI_CHxCONF[6] EPOL | 0x1 |
| Clock held high during active state | MCSPI_CHxCONF[1] POL | 0x0 |
| Data latched on odd-numbered edges of the SPI clock | MCSPI_CHxCONF[0] PHA | 0x0 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-9. SPI Transmit-and-Receive Mode Initialization (continued)**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Reset the status bits. | MCSPI_IRQSTATUS | 0x0 |

#### 22.2.12.2.1 Common Transfer Procedures Without FIFO – Polling Method

##### 22.2.12.2.1.1 Receive-Only Procedure – Polling Method

Table 22-10 lists the receive-only procedure using the polling method. The SPI is acting as slave.

**Table 22-10. Receive-Only Procedure – Polling Method**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Configure the channel according to the mode | See Table 22-7. | |
| Start the channel. | MCSPI_CHxCTRL[0] EN | 0x1 |
| Wait until receive register is full? | MCSPI_RXx | = 0x1 |
| Stop the channel. | MCSPI_CHxCTRL[0] EN | 0x0 |

##### 22.2.12.2.1.2 Receive-Only Procedure – Interrupt Method

Table 22-11 lists the receive-only procedure using the interrupt method. The SPI is acting as slave. Channel 0 is used. For the rest channels (1, 2 and 3) the same logic applies.

**Table 22-11. Receive-Only Procedure – Interrupt Method**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Configure the channel according to the mode. | See Table 22-7. | |
| Start the channel. | MCSPI_CHxCTRL[0] EN | 0x1 |
| Enable the interrupt for the receiver register. | MCSPI_IRQENABLE[2] RX0_FULL_ENABLE | 0x1 |
| Read the status register. | MCSPI_IRQSTATUS[2] RX0_FULL | 0x0 |
| Disable the interrupt. | MCSPI_IRQENABLE[2] RX0_FULL_ENABLE | 0x0 |
| Stop the channel. | MCSPI_CHxCTRL[0] EN | 0x0 |
| Read the receiver register. | MCSPI_RX0 | xxxx |

##### 22.2.12.2.1.3 Transmit-Only Procedure – Polling Method

Table 22-12 lists the transmit-only procedure using the polling method. The SPI is acting as master.

**Table 22-12. Transmit-Only Procedure – Polling Method**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Configure the channel according to the mode. | See Table 22-8 | |
| Start the channel. | MCSPI_CHxCTRL[0] EN | 0x1 |
| Wait until end of transfer? | MCSPI_CHxSTAT[2:1] | = 0x2 |
| Stop the channel. | MCSPI_CHxCTRL[0] EN | 0x0 |

##### 22.2.12.2.1.4 Transmit-and-Receive Procedure – Polling Method

Table 22-12 lists the transmit-and-receive procedure using the polling method. The SPI is acting as master and slave.


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-13. Transmit-and-Receive Procedure – Polling Method**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Configure the channel according to the mode. | See Table 22-9. | |
| Start the channel. | MCSPI_CHxCTRL[0] EN | 0x1 |
| Wait until transmit/receive word? | MCSPI_CHxSTAT[2:0] | = 0x3 |
| Stop the channel. | MCSPI_CHxCTRL[0] EN | 0x0 |

#### 22.2.12.2.2 Common Transfer Procedures With FIFO – Polling Method

When using FIFO the SPI module can start the transfer only after the first write request is released by writing the MCSPI_TXx register (where x = 0, 1, 2 and 3). The first write request can be managed by the IRQ routine or DMA handler. The end of transfer is more complex and depends on the transfer type. See Table 22-12 through Table 22-19.

In multi-channel master mode, be careful not to overwrite the bits of the other channels when initializing the MCSPI_IRQSTATUS and MCSPI_IRQENABLE registers.

##### 22.2.12.2.2.1 Receive-Only Procedure With Word Count – Polling Method

**Table 22-14. Receive-Only Procedure With Word Count – Polling Method**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Configure the channel according to the mode. | See Table 22-7. | |
| Start the channel. | MCSPI_CHxCTRL[0] EN | 0x1 |
| Wait until end of word count? | MCSPI_IRQSTATUS[17] EOW | = 0x1 |
| Stop the channel. | MCSPI_CHxCTRL[0] EN | 0x0 |
| Read from the receiver register. | MCSPI_RXx | xxxx |

##### 22.2.12.2.2.2 Transmit-Only Procedure With and Without Word Count – Polling Method

**Table 22-15. Transmit-Only Procedure Without Word Count – Polling Method**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Configure the channel according to the mode. | See Table 22-8. | |
| Start the channel. | MCSPI_CHxCTRL[0] EN | 0x1 |
| Wait until end of word count? | MCSPI_IRQSTATUS[17] EOW | = 0x1 |
| Wait until end of transfer? | MCSPI_CHxSTAT[2] EOT<br>MCSPI_CHxSTAT[3] TXFFE | = 0x1<br>= 0x1 |
| Stop the channel. | MCSPI_CHxCTRL[0] EN | 0x0 |

##### 22.2.12.2.2.3 Transmit-Only Procedure With and Without Word Count – Interrupt Method

**Table 22-16. Transmit-Only Procedure With Word Count – Interrupt Method**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Configure the channel according to the mode. | See Table 22-8. | |
| Start the channel. | MCSPI_CHxCTRL[0] EN | 0x1 |
| Enable the interrupt for the transmit register. | MCSPI_IRQENABLE[4] TX1_EMPTY_ENABLE | 0x1 |
| End of word count. | MCSPI_IRQSTATUS[17] EOW | = 0x1 |
| End of transfer | MCSPI_CHxSTAT[2] EOT<br>MCSPI_CHxSTAT[3] TXFFE | = 0x1<br>= 0x1 |
| Clear the interrupt. | MCSPI_IRQENABLE[17] EOWKE | = 0x0 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-16. Transmit-Only Procedure With Word Count – Interrupt Method (continued)**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Disable the interrupt for the transmit register. | MCSPI_IRQENABLE[4] TX1_EMPTY_ENABLE | 0x0 |
| Stop the channel. | MCSPI_CHxCTRL[0] EN | 0x0 |

#### *22.2.12.2.2.4 Transmit-and-Receive Procedure With Word Count – Polling Method*

**Table 22-17. Transmit-and-Receive Procedure With Word Count – Polling Method**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Configure the channel according to the mode. | See Table 22-9. | |
| Start the channel. | MCSPI_CHxCTRL[0] EN | 0x1 |
| Wait until end of word count? | MCSPI_IRQSTATUS[17] EOW | = 0x1 |
| Stop the channel. | MCSPI_CHxCTRL[0] EN | 0x0 |
| Read from the receiver register. | MCSPI_RXx | xxxx |

#### *22.2.12.2.2.5 Transmit-and-Receive Procedure with Word Count – Interrupt Method*

**Table 22-18. Transmit-and-Receive Procedure With Word Count – Interrupt Method**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Configure the channel according to the mode. | See Table 22-9. | |
| Start the channel. | MCSPI_CHxCTRL[0] EN | 0x1 |
| Enable the interrupt for the receiver register. | MCSPI_IRQENABLE[2] RX0_FULL_ENABLE | 0x1 |
| Enable the interrupt for the transmit register. | MCSPI_IRQENABLE[4] TX1_EMPTY_ENABLE | 0x1 |
| End of word count? | MCSPI_IRQSTATUS[17] EOW | = 0x1 |
| Clear the interrupt. | MCSPI_IRQENABLE[17] EOWKE | = 0x0 |
| Disable the interrupt for the receiver register. | MCSPI_IRQENABLE[2] RX0_FULL_ENABLE | 0x0 |
| Disable the interrupt for the transmit register. | MCSPI_IRQENABLE[4] TX1_EMPTY_ENABLE | 0x0 |
| Stop the channel. | MCSPI_CHxCTRL[0] EN | 0x0 |
| Read from the receiver register. | MCSPI_RXx | xxxx |

#### *22.2.12.2.2.6 Transmit-and-Receive Procedure Without Word Count – Polling Method*

**Table 22-19. Transmit-and-Receive Procedure Without Word Count – Polling Method**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Configure the channel according to the mode. | See Table 22-9. | |
| Start the channel. | MCSPI_CHxCTRL[0] EN | 0x1 |
| Wait until end of transfer? | MCSPI_CHxSTAT[2] EOT<br>MCSPI_CHxSTAT[3] TXFFE | = 0x1<br>= 0x1 |
| Stop the channel. | MCSPI_CHxCTRL[0] EN | 0x0 |
| Read from the receiver register. | MCSPI_RXx | xxxx |

### *22.2.13 Interrupt and DMA Events*

SPI has two DMA requests (Rx and Tx) per channel. It also has one interrupt line for all the interrupt requests.


Copyright © 2011–2016, Texas Instruments Incorporated

## 22.3 SPI Registers

Table 22-20 lists the SPI registers.

**Table 22-20. SPI Registers**

| Offset Address | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | MCSPI_HL_REV | McSPI IP revision register | Section 22.3.1 |
| 4h | MCSPI_HL_HWINFO | McSPI IP hardware information register | Section 22.3.2 |
| 10h | MCSPI_HL_SYSCONFIG | McSPI IP system configuration register | Section 22.3.3 |
| 100h | MCSPI_REVISION | McSPI revision register | Section 22.3.4 |
| 110h | MCSPI_SYSCONFIG | McSPI system configuration register | Section 22.3.5 |
| 114h | MCSPI_SYSSTATUS | McSPI system status register | Section 22.3.6 |
| 118h | MCSPI_IRQSTATUS | McSPI interrupt status register | Section 22.3.7 |
| 11Ch | MCSPI_IRQENABLE | McSPI interrupt enable register | Section 22.3.8 |
| 120h | MCSPI_WAKEUPENABLE | McSPI wakeup enable register | Section 22.3.9 |
| 124h | MCSPI_SYST | McSPI system test register | Section 22.3.10 |
| 128h | MCSPI_MODULCTRL | McSPI module control register | Section 22.3.11 |
| 12Ch | MCSPI_CH0CONF | McSPI channel 0 configuration register | Section 22.3.12 |
| 130h | MCSPI_CH0STAT | McSPI channel 0 status register | Section 22.3.13 |
| 134h | MCSPI_CH0CTRL | McSPI channel 0 control register | Section 22.3.14 |
| 138h | MCSPI_TX0 | McSPI channel 0 FIFO transmit buffer register | Section 22.3.15 |
| 13Ch | MCSPI_RX0 | McSPI channel 0 FIFO receive buffer register | Section 22.3.16 |
| 140h | MCSPI_CH1CONF | McSPI channel 1 configuration register | Section 22.3.12 |
| 144h | MCSPI_CH1STAT | McSPI channel 1 status register | Section 22.3.13 |
| 148h | MCSPI_CH1CTRL | McSPI channel 1 control register | Section 22.3.14 |
| 14Ch | MCSPI_TX1 | McSPI channel 1 FIFO transmit buffer register | Section 22.3.15 |
| 150h | MCSPI_RX1 | McSPI channel 1 FIFO receive buffer register | Section 22.3.16 |
| 154h | MCSPI_CH2CONF | McSPI channel 2 configuration register | Section 22.3.12 |
| 158h | MCSPI_CH2STAT | McSPI channel 2 status register | Section 22.3.13 |
| 15Ch | MCSPI_CH2CTRL | McSPI channel 2 control register | Section 22.3.14 |
| 160h | MCSPI_TX2 | McSPI channel 2 FIFO transmit buffer register | Section 22.3.15 |
| 164h | MCSPI_RX2 | McSPI channel 2 FIFO receive buffer register | Section 22.3.16 |
| 168h | MCSPI_CH3CONF | McSPI channel 3 configuration register | Section 22.3.12 |
| 16Ch | MCSPI_CH3STAT | McSPI channel 3 status register register | Section 22.3.13 |
| 170h | MCSPI_CH3CTRL | McSPI channel 3 control register | Section 22.3.14 |
| 174h | MCSPI_TX3 | McSPI channel 3 FIFO transmit buffer register | Section 22.3.15 |
| 178h | MCSPI_RX3 | McSPI channel 3 FIFO receive buffer register | Section 22.3.16 |
| 17Ch | MCSPI_XFERLEVEL | McSPI transfer levels register | Section 22.3.17 |
| 180h | MCSPI_DAFTX | DMA address aligned FIFO transmitter register | Section 22.3.18 |
| 1A0h | MCSPI_DAFRX | DMA address aligned FIFO receiver register | Section 22.3.19 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.1 McSPI IP Revision Register (MCSPI_HL_REV)

The McSPI IP revision register (MCSPI_HL_REV) holds the IP revision identifier information. The MCSPI_HL_REV is shown in Figure 22-25 and described in Table 22-21.

**Figure 22-25. McSPI IP Revision Register (MCSPI_HL_REV)**

| 31 | 30 | 29 | 28 | 27 | 16 |
|---|---|---|---|---|---|
| SCHEME | | Reserved | | FUNC | |
| R-1 | | R-0 | | R-30h | |

| 15 | 11 | 10 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| R_RTL | | X_MAJOR | | CUSTOM | | Y_MINOR | |
| R-0 | | R-0 | | R-0 | | R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 22-21. McSPI IP Revision Register (MCSPI_HL_REV) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | SCHEME | 0-3h | Identifies scheme. |
| 29-28 | Reserved | 0 | Reserved for module specific status information. Read returns 0. |
| 27-16 | FUNC | 0-FFFh | Indicates a software compatible module family. |
| 15-11 | R_RTL | 0-1Fh | RTL version (R). |
| 10-8 | X_MAJOR | 0-7h | Major revision (X). |
| 7-6 | CUSTOM | 0-3h | Indicates a special version for a particular device. |
| 5-0 | Y_MINOR | 0-3Fh | Minor revision (Y). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.2 McSPI IP Hardware Information Register (MCSPI_HL_HWINFO)

The McSPI IP hardware information register (MCSPI_HL_HWINFO) provides information about the IP hardware configuration. The MCSPI_HL_HWINFO is shown in Figure 22-26 and described in Table 22-22.

**Figure 22-26. McSPI IP Hardware Information Register (MCSPI_HL_HWINFO)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 6 | 5 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | FFNBYTE | | USEFIFO |
| R-0 | | R-4h | | R-1 |

LEGEND: R = Read only; -*n* = value after reset

**Table 22-22. McSPI IP Hardware Information Register (MCSPI_HL_HWINFO) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-6 | Reserved | 0 | Reserved. |
| 5-1 | FFNBYTE | | FIFO number of byte generic parameter. Defines the value of FFNBYTE generic parameter. |
| | | 0 | Reserved |
| | | 1h | FIFO 16 bytes depth |
| | | 2h | FIFO 32 bytes depth |
| | | 3h | Reserved |
| | | 4h | FIFO 64 bytes depth |
| | | 5h-7h | Reserved |
| | | 8h | FIFO 128 bytes depth |
| | | 9h-Fh | Reserved |
| | | 10h | FIFO 256 bytes depth |
| | | 11h-1Fh | Reserved |
| 0 | USEFIFO | | Use of a FIFO enable. |
| | | 0 | FIFO not implemented in design |
| | | 1 | FIFO and its management implemented in design with depth defined by FFNBYTE generic. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.3 McSPI IP System Configuration Register (MCSPI_HL_SYSCONFIG)

The McSPI IP system configuration register (MCSPI_HL_SYSCONFIG) allows control of the clock management. The MCSPI_HL_SYSCONFIG is shown in Figure 22-27 and described in Table 22-23.

**Figure 22-27. McSPI IP System Configuration Register (MCSPI_HL_SYSCONFIG)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | IDLEMODE | | FREEEMU | SOFTRESET |
| R-0 | | R/W-2h | | R/W-0 | R/W-1 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 22-23. McSPI IP System Configuration Register (MCSPI_HL_SYSCONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-4 | Reserved | 0 | Reserved. |
| 3-2 | IDLEMODE | | Configuration of the local target state management mode. The target can handle read/write transactions as long as it is out of IDLE state. |
| | | 0 | Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally,that is, regardless of the IP module's internal requirements. |
| | | 1h | No-idle mode: local target never enters idle state. |
| | | 2h | Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events. |
| | | 3h | Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented. |
| 1 | FREEEMU | | Sensitivity to emulation (debug) suspend input signal. |
| | | 0 | IP module is sensitive to emulation suspend. |
| | | 1 | IP module is not sensitive to emulation suspend. |
| 0 | SOFTRESET | | Software reset. |
| | | Write 0 | No action |
| | | Read 0 | Reset done, no pending action |
| | | Write 1 | Initiate software reset |
| | | Read 1 | Reset (software or other) ongoing |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.4 McSPI Revision Register (MCSPI_REVISION)

The McSPI revision register (MCSPI_REVISION) contains the hard-coded RTL revision number. The MCSPI_REVISION is shown in Figure 22-28 and described in Table 22-24.

**Figure 22-28. McSPI Revision Register (MCSPI_REVISION)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | REV | |
| R-0 | | R-x | |

LEGEND: R = Read only; -*n* = value after reset

**Table 22-24. McSPI Revision Register (MCSPI_REVISION) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-8 | Reserved | 0 | Reserved. |
| 7-0 | REV | 0-FFh | Identifies revision of peripheral. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.5 McSPI System Configuration Register (MCSPI_SYSCONFIG)

The McSPI system configuration register (MCSPI_SYSCONFIG) allows control of various parameters of the module interface. It is not sensitive to software reset. The MCSPI_SYSCONFIG is shown in Figure 22-29 and described in Table 22-25.

**Figure 22-29. McSPI System Configuration Register (MCSPI_SYSCONFIG)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R/W-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | CLOCKACTIVITY | |
| R/W-0 | | | | | | R/W-0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | SIDLEMODE | | ENAWAKEUP | SOFTRESET | AUTOIDLE |
| R/W-0 | | | R/W-2h | | R/W-1 | R/W-0 | R/W-1 |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 22-25. McSPI System Configuration Register (MCSPI_SYSCONFIG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-10 | Reserved | 0 | Reads returns 0 |
| 9-8 | CLOCKACTIVITY | | Clocks activity during wake-up mode period. |
| | | 0 | OCP and Functional clocks may be switched off. |
| | | 1h | OCP clock is maintained. Functional clock may be switched-off. |
| | | 2h | Functional clock is maintained. OCP clock may be switched-off. |
| | | 3h | OCP and Functional clocks are maintained. |
| 7-5 | Reserved | 0 | Reads returns 0 |
| 4-3 | SIDLEMODE | | Power management |
| | | 0 | If an idle request is detected, the McSPI acknowledges it unconditionally and goes in Inactive mode. Interrupt, DMA requests are unconditionally de-asserted. |
| | | 1h | If an idle request is detected, the request is ignored and keeps on behaving normally. |
| | | 2h | Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. |
| | | 3h | Reserved |
| 2 | ENAWAKEUP | | Enable Wakeup. |
| | | 0 | Wakeup capability is disabled. |
| | | 1 | Wakeup capability is enabled. |
| 1 | SOFTRESET | | Software reset. During reads it always returns 0. |
| | | 0 | (write) Normal mode |
| | | 1 | (write) Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. |
| 0 | AUTOIDLE | | Internal OCP Clock gating strategy |
| | | 0 | OCP clock is free-running |
| | | 1 | Automatic OCP clock gating strategy is applied, based on the OCP interface activity |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.6 McSPI System Status Register (MCSPI_SYSSTATUS)

The McSPI system status register (MCSPI_SYSSTATUS) provides status information about the module excluding the interrupt status information. The MCSPI_SYSSTATUS is shown in Figure 22-30 and described in Table 22-26.

**Figure 22-30. McSPI System Status Register (MCSPI_SYSSTATUS)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | RESETDONE |
| R-0 | | | | | | | | | | | | | | | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 22-26. McSPI System Status Register (MCSPI_SYSSTATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved for module specific status information. Read returns 0. |
| 0 | RESETDONE | | Internal Reset Monitoring |
| | | 0 | Internal module reset is on-going |
| | | 1 | Reset completed |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.7 McSPI Interrupt Status Register (MCSPI_IRQSTATUS)

The McSPI interrupt status register (MCSPI_IRQSTATUS) regroups all the status of the module internal events that can generate an interrupt. The MCSPI_IRQSTATUS is shown in Figure 22-31 and described in Table 22-27.

**NOTE:** In SYSTEST mode, the bits of this register have no meaning and always read 0.

**Figure 22-31. McSPI Interrupt Status Register (MCSPI_IRQSTATUS)**

| 31 | | | | | | | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | EOW | Rsvd |
| R/W-0 | | | | | | | | R/W-0 | R-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Rsvd | RX3_FULL | TX3_UNDERFLOW | TX3_EMPTY | Reserved | RX2_FULL | TX2_UNDERFLOW | TX2_EMPTY |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | RX1_FULL | TX1_UNDERFLOW | TX1_EMPTY | RX0_OVERFLOW | RX0_FULL | TX0_UNDERFLOW | TX0_EMPTY |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; *-n* = value after reset

**Table 22-27. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-18 | Reserved | 0 | Reads returns 0 |
| 17 | EOW | | End of word (EOW) count event when a channel is enabled using the FIFO buffer and the channel has sent the number of McSPI words defined by the MCSPI_XFERLEVEL[WCNT]. |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 16 | Reserved | 0 | Reserved |
| 15 | Reserved | 0 | Reads returns 0 |
| 14 | RX3_FULL | | Receiver register is full or almost full. Only when Channel 3 is enabled. This bit indicate FIFO almost full status when built-in FIFO is used for receive register (MCSPI_CH3CONF[FFE3R] is set). |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 13 | TX3_UNDERFLOW | | Transmitter register underflow. Only when Channel 3 is enabled. The transmitter register is empty (not updated by Host or DMA with new data) before its time slot assignment.<br>Exception: No TX_underflow event when no data has been loaded into the transmitter register since channel has been enabled. |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-27. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 12 | TX3_EMPTY | | Transmitter register is empty or almost empty. This bit indicate FIFO almost full status when built-in FIFO is used for transmit register (MCSPI_CH3CONF[FFE3W] is set). |
| | | | **Note:** Enabling the channel automatically raises this event. |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 11 | Reserved | 0 | Reads returns 0 |
| 10 | RX2_FULL | | Receiver register full or almost full. Channel 2 This bit indicate FIFO almost full status when built-in FIFO is used for receive register (MCSPI_CH3CONF[FFE2R] is set). |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 9 | TX2_UNDERFLOW | | Transmitter register underflow. Channel 2 |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 8 | TX2_EMPTY | | Transmitter register empty or almost empty. Channel 2. This bit indicate FIFO almost full status when built-in FIFO is used for transmit register (MCSPI_CH3CONF[FFE2W] is set). |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 7 | Reserved | 0 | Reads returns 0 |
| 6 | RX1_FULL | | Receiver register full or almost full. Channel 1. This bit indicate FIFO almost full status when built-in FIFO is use for receive register (MCSPI_CH3CONF[FFE1R] is set). |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 5 | TX1_UNDERFLOW | | Transmitter register underflow. Channel 1. |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 4 | TX1_EMPTY | | Transmitter register empty or almost empty. Channel 1. This bit indicate FIFO almost full status when built-in FIFO is use for transmit register (MCSPI_CH3CONF[FFE1W] is set). |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 3 | RX0_OVERFLOW | | Receiver register overflow (slave mode only). Channel 0. |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-27. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 2 | RX0_FULL | | Receiver register full or almost full. Channel 0. Receiver register full or almost full. Channel 0 |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 1 | TX0_UNDERFLOW | | Transmitter register underflow. Channel 0. |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |
| 0 | TX0_EMPTY | | Transmitter register empty or almost empty. Channel 0. This bit indicate FIFO almost full status when built-in FIFO is use for transmit register (MCSPI_CH3CONF[FFE0W] is set). |
| | | Write 0 | Event status bit is unchanged. |
| | | Read 0 | Event false. |
| | | Write 1 | Event status bit is reset. |
| | | Read 1 | Event is pending. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.8 McSPI Interrupt Enable Register (MCSPI_IRQENABLE)

This McSPI interrupt enable register (MCSPI_IRQENABLE) enables/disables the module internal sources of interrupt, on an event-by-event basis. The MCSPI_IRQENABLE is shown in Figure 22-32 and described in Table 22-28.

**Figure 22-32. McSPI Interrupt Enable Register (MCSPI_IRQENABLE)**

| 31 | | | | | | | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | EOWKE | Rsvd |
| R/W-0 | | | | | | | | R/W-0 | R-0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Rsvd | RX3_FULL_ ENABLE | TX3_UNDERFLOW_ ENABLE | TX3_EMPTY_ ENABLE | Reserved | RX2_FULL_ ENABLE | TX2_UNDERFLOW_ ENABLE | TX2_EMPTY_ ENABLE |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Rsvd | RX1_FULL_ ENABLE | TX1_UNDERFLOW_ ENABLE | TX1_EMPTY_ ENABLE | RX0_OVERFLOW_ ENABLE | RX0_FULL_ ENABLE | TX0_UNDERFLOW_ ENABLE | TX0_EMPTY_ ENABLE |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; *-n* = value after reset

**Table 22-28. McSPI Interrupt Enable Register (MCSPI_IRQENABLE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-18 | Reserved | 0 | Reads return 0 |
| 17 | EOWKE | | End of word count interrupt enable. |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 16 | Reserved | | Reserved |
| 15 | Reserved | 0 | Reads return 0 |
| 14 | RX3_FULL_ ENABLE | | MCSPI_RX3 receiver register full or almost full interrupt enable (channel 3). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 13 | TX3_UNDERFLOW_ ENABLE | | MCSPI_TX3 transmitter register underflow interrupt enable (channel 3). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 12 | TX3_EMPTY_ ENABLE | | MCSPI_TX3 transmitter register empty or almost empty interrupt enable (channel 3). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 11 | Reserved | 0 | Reads return 0 |
| 10 | RX2_FULL_ ENABLE | | MCSPI_RX2 receiver register full or almost full interrupt enable (channel 2). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 9 | TX2_UNDERFLOW_ ENABLE | | MCSPI_TX2 transmitter register underflow interrupt enable (channel 2). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 8 | TX2_EMPTY_ ENABLE | | MCSPI_TX2 transmitter register empty or almost empty interrupt enable (channel 2). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 7 | Reserved | 0 | Reads return 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-28. McSPI Interrupt Enable Register (MCSPI_IRQENABLE) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 6 | RX1_FULL_ ENABLE | | MCSPI_RX1 receiver register full or almost full interrupt enable (channel 1) |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 5 | TX1_UNDERFLOW_ ENABLE | | MCSPI_TX1 transmitter register underflow interrupt enable (channel 1). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 4 | TX1_EMPTY_ ENABLE | | MCSPI_TX1 transmitter register empty or almost empty interrupt enable (channel 1). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 3 | RX0_OVERFLOW_ ENABLE | | MCSPI_RX0 receivier register overflow interrupt enable (channel 0). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 2 | RX0_FULL_ ENABLE | | MCSPI_RX0 receiver register full or almost full interrupt enable (channel 0). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 1 | TX0_UNDERFLOW_ ENABLE | | MCSPI_TX0 transmitter register underflow interrupt enable (channel 0). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |
| 0 | TX0_EMPTY_ ENABLE | | MCSPI_TX0 transmitter register empty or almost empty interrupt enable (channel 0). |
| | | 0 | Interrupt is disabled. |
| | | 1 | Interrupt is enabled. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.9 McSPI Wakeup Enable Register (MCSPI_WAKEUPENABLE)

The McSPI wakeup enable register (MCSPI_WAKEUPENABLE) allows you to enable/disable the module internal sources of wakeup on an event-by-event basis. The MCSPI_WAKEUPENABLE is shown in Figure 22-33 and described in Table 22-29.

**Figure 22-33. McSPI Wakeup Enable Register (MCSPI_WAKEUPENABLE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | WKEN |
| R-0 | | | | | | | | | | | | | | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 22-29. McSPI Wakeup Enable Register (MCSPI_WAKEUPENABLE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-1 | Reserved | 0 | Reserved. Read returns 0. |
| 0 | WKEN | | Wakeup functionality in slave mode when an active control signal is detected on the $\overline{\text{SPI_SCS[n]}}$ line programmed in MCSPI_CH(i)CONF[SPIENSLV]. |
| | | 0 | The event is not allowed to wakeup the system, even if MCSPI_SYSCONFIG[ENAWAKEUP] is set. |
| | | 1 | The event is allowed to wakeup the system if MCSPI_SYSCONFIG[ENAWAKEUP] is set. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.10 McSPI System Test Register (MCSPI_SYST)

This McSPI system test register (MCSPI_SYST) is used to configure the system interconnect either internally to the peripheral bus or externally to the device I/O pads, when the module is configured in the system test (SYSTEST) mode. The MCSPI_SYST is shown in Figure 22-34 and described in Table 22-30.

**Figure 22-34. McSPI System Test Register (MCSPI_SYST)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R/W-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | SSB | SPIENDIR | SPIDATDIR1 | SPIDATDIR0 |
| R/W-0 | | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | SPICLK | SPIDAT_1 | SPIDAT_0 | SPIEN_3 | SPIEN_2 | SPIEN_1 | SPIEN_0 |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 22-30. McSPI System Test Register (MCSPI_SYST) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-12 | Reserved | 0 | Reads returns 0 |
| 11 | SSB | | Set status bit. This bit must be cleared prior attempting to clear a status bit of the MCSPI_ IRQSTATUS register. |
| | | 0 | No action. Writing 0 does not clear already set status bits. |
| | | | This bit must be cleared prior attempting to clear a status bit of the <MCSPI_IRQSTATUS> register. |
| | | 1 | Writing 1 sets to 1 all status bits contained in the MCSPI_IRQSTATUS register. |
| | | | Writing 1 into this bit sets to 1 all status bits contained in the <MCSPI_IRQSTATUS> register. |
| 10 | SPIENDIR | | Sets the direction of the $\overline{\text{SPI_SCS}}$[3:0] lines and SPI_SCLK line. |
| | | 0 | Output (as in master mode). |
| | | 1 | Input (as in slave mode). |
| 9 | SPIDATDIR1 | | Sets the direction of the SPI_D[1]. |
| | | 0 | Output |
| | | 1 | Input |
| 8 | SPIDATDIR0 | | Sets the direction of the SPI_D[0]. |
| | | 0 | Output |
| | | 1 | Input |
| 7 | Reserved | 0 | Reserved |
| 6 | SPICLK | | SPI_SCLK line (signal data value) |
| | | | If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the CLKSPI line (high or low), and a write into this bit has no effect. |
| | | | If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the CLKSPI line is driven high or low according to the value written into this register. |
| 5 | SPIDAT_1 | | SPI_D[1] line (signal data value) |
| | | | If MCSPI_SYST[SPIDATDIR1] = 0 (output mode direction), the SPI_D[1] line is driven high or low according to the value written into this register. |
| | | | If MCSPI_SYST[SPIDATDIR1] = 1 (input mode direction), this bit returns the value on the SPI_D[1] line (high or low), and a write into this bit has no effect. |
| 4 | SPIDAT_0 | | SPI_D[0] line (signal data value) |
| | | | If MCSPI_SYST[SPIDATDIR0] = 0 (output mode direction), the SPI_D[0] line is driven high or low according to the value written into this register. |
| | | | If MCSPI_SYST[SPIDATDIR0] = 1 (input mode direction), this bit returns the value on the SPI_D[0] line (high or low), and a write into this bit has no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-30. McSPI System Test Register (MCSPI_SYST) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 3 | SPIEN_3 | | $\overline{\text{SPI_SCS}}$[3] line (signal data value)<br>If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the $\overline{\text{SPI_SCS}}$[3] line is driven high or low according to the value written into this register.<br>If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the $\overline{\text{SPI_SCS}}$[3] line (high or low), and a write into this bit has no effect. |
| 2 | SPIEN_2 | | $\overline{\text{SPI_SCS}}$[2] line (signal data value)<br>If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the $\overline{\text{SPI_SCS}}$[2] line is driven high or low according to the value written into this register.<br>If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the $\overline{\text{SPI_SCS}}$[2] line (high or low), and a write into this bit has no effect. |
| 1 | SPIEN_1 | | $\overline{\text{SPI_SCS}}$[1] line (signal data value)<br>If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the $\overline{\text{SPI_SCS}}$[1] line is driven high or low according to the value written into this register.<br>If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the $\overline{\text{SPI_SCS}}$[1] line (high or low), and a write into this bit has no effect. |
| 0 | SPIEN_0 | | $\overline{\text{SPI_SCS}}$[0] line (signal data value)<br>If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the $\overline{\text{SPI_SCS}}$[0] line is driven high or low according to the value written into this register.<br>If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the $\overline{\text{SPI_SCS}}$[0] line (high or low), and a write into this bit has no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.11 McSPI Module Control Register (MCSPI_MODULCTRL)

This McSPI module control register (MCSPI_MODULCTRL) is used to configure the serial port interface. The MCSPI_MODULCTRL is shown in Figure 22-35 and described in Table 22-31.

**Figure 22-35. McSPI Module Control Register (MCSPI_MODULCTRL)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R/W-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | FDAA |
| R/W-0 | | | | | | | R/W-0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MOA | INITDLY | | | SYSTEM_TEST | MS | PIN34 | SINGLE |
| R/W-0 | R/W-0 | | | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 22-31. McSPI Module Control Register(MCSPI_MODULCTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | 0 | |
| 8 | FDAA | | FIFO DMA Address 256-bit aligned. This register is used when a FIFO is managed by the module and DMA connected to the controller provides only 256 bit aligned address. If this bit is set the enabled channel which uses the FIFO has its datas managed through MCSPI_DAFTX and MCSPI_DAFRX registers instead of MCSPI_TX(i) and MCSPI_RX(i) registers. |
| | | 0 | FIFO data managed by MCSPI_TX(i) and MCSPI_RX(i) registers. |
| | | 1 | FIFO data managed by MCSPI_DAFTX and MCSPI_DAFRX registers. |
| 7 | MOA | | Multiple word ocp access. This register can only be used when a channel is enabled using a FIFO. It allows the system to perform multiple SPI word access for a single 32-bit OCP word access. This is possible for WL < 16. |
| | | 0 | Multiple word access disabled |
| | | 1 | Multiple word access enabled with FIFO |
| 6-4 | INITDLY | | Initial SPI delay for first transfer. This register is an option only available in single master mode, The controller waits for a delay to transmit the first SPI word after channel enabled and corresponding TX register filled. This delay is based on SPI output frequency clock, No clock output provided to the boundary and chip select is not active in 4 pin mode within this period. |
| | | 0 | No delay for first SPI transfer |
| | | 1h | The controller wait 4 SPI bus clock |
| | | 2h | The controller wait 8 SPI bus clock |
| | | 3h | The controller wait 16 SPI bus clock |
| | | 4h | The controller wait 32 SPI bus clock |
| | | 5h-Fh | Reserved |
| 3 | SYSTEM_TEST | | Enables the system test mode |
| | | 0 | Functional mode |
| | | 1 | System test mode (SYSTEST) |
| 2 | MS | | Master/ Slave |
| | | 0 | Master - The module generates the SPI_SCLK and $\overline{\text{SPI_SCS[3:0]}}$ |
| | | 1 | Slave - The module receives the SPI_SCLK and $\overline{\text{SPI_SCS[3:0]}}$ |
| 1 | PIN34 | | Pin mode selection. This register is used to configure the SPI pin mode, in master or slave mode. If asserted, the controller only use SPI_D[0], SPI_D[1], and SPI_SCLK clock pin for SPI transfers. |
| | | 0 | $\overline{\text{SPI_SCS[n]}}$ is used as a chip select. |
| | | 1 | $\overline{\text{SPI_SCS[n]}}$ is not used. In this mode all related option to chip select have no meaning. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-31. McSPI Module Control Register(MCSPI_MODULCTRL) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 0 | SINGLE | | Single channel / Multichannel (master mode only) For master mode, the user has to set this bit to 0 for Multi or 1 for Single. For Slave mode, this bit has to be set to 0. |
| | | 0 | More than one channel will be used in master mode. |
| | | 1 | Only one channel will be used in master mode. This bit must be set in Force $\overline{\text{SPI_SCS[n]}}$ mode. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.12 McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF)

The McSPI channel *i* configuration register (MCSPI_CH(i)CONF) is used to configure channel*i*. The (MCSPI_CH(i)CONF) is shown in Figure 22-36 and described in Table 22-32.

**Figure 22-36. McSPI Channel *(i )* Configuration Register (MCSPI_CH(i)CONF)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | CLKG | FFER | FFEW | TCS | | SBPOL |
| R/W-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | | R/W-0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SBE | SPIENSLV | | FORCE | TURBO | IS | DPE1 | DPE0 |
| R/W-0 | R/W-0 | | R/W-0 | R/W-0 | R/W-1 | R/W-1 | R/W-0 |
| 15 | 14 | 13 | 12 | 11 | | | 8 |
| DMAR | DMAW | TRM | | WL | | | |
| R/W-0 | R/W-0 | R/W-0 | | R/W-0 | | | |
| 7 | 6 | 5 | | | 2 | 1 | 0 |
| WL | EPOL | CLKD | | | | POL | PHA |
| R/W-0 | R/W-0 | R/W-0 | | | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 22-32. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Read returns 0 |
| 29 | CLKG | | Clock divider granularity. This register defines the granularity of channel clock divider: power of two or one clock cycle granularity. |
| | | | When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio. Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values. |
| | | 0 | Clock granularity of power of 2 |
| | | 1 | 1 clock cycle granularity |
| 28 | FFER | | FIFO enabled for receive. Only one channel can have this bit set. |
| | | 0 | The buffer is not used to receive data. |
| | | 1 | The buffer is used to receive data. |
| 27 | FFEW | | FIFO enabled for transmit. Only one channel can have this bit set. |
| | | 0 | The buffer is not used to transmit data. |
| | | 1 | The buffer is used to transmit data. |
| 26-25 | TCS | | Chip select time control. This 2-bit field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock. |
| | | 0 | 0.5 clock cycles |
| | | 1h | 1.5 clock cycles |
| | | 2h | 2.5 clock cycles |
| | | 3h | 3.5 clock cycles |
| 24 | SBPOL | | Start bit polarity. |
| | | 0 | Start bit polarity is held to 0 during SPI transfer. |
| | | 1 | Start bit polarity is held to 1 during SPI transfer. |
| 23 | SBE | | Start bit enable for SPI transfer. |
| | | 0 | Default SPI transfer length as specified by WL bit field. |
| | | 1 | Start bit D/CX added before SPI transfer. Polarity is defined by MCSPI_CH(i)CONF[SBPOL]. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-32. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 22-21 | SPIENSLV | | Channel 0 only and slave mode only: SPI slave select signal detection. |
| | | 0 | Detection enabled only on $\overline{\text{SPI_SCS[0]}}$ |
| | | 1h | Detection enabled only on $\overline{\text{SPI_SCS[1]}}$ |
| | | 2h | Detection enabled only on $\overline{\text{SPI_SCS[2]}}$ |
| | | 3h | Detection enabled only on $\overline{\text{SPI_SCS[3]}}$ |
| 20 | FORCE | | Manual $\overline{\text{SPI_SCS[n]}}$ assertion to keep $\overline{\text{SPI_SCS[n]}}$ active between SPI words. (single channel master mode only) |
| | | 0 | Writing 0 into this bit drives the $\overline{\text{SPI_SCS[n]}}$ line when MCSPI_CHCONF(i)[EPOL] = 0, and drives it high when MCSPI_CHCONF(i)[EPOL] = 1. |
| | | 1 | Writing 1 into this bit drives the $\overline{\text{SPI_SCS[n]}}$ line when MCSPI_CHCONF(i)[EPOL] = 0, and drives it low when MCSPI_CHCONF(i)[EPOL] = 1. |
| 19 | TURBO | | Turbo mode. |
| | | 0 | Turbo is deactivated (recommended for single SPI word transfer). |
| | | 1 | Turbo is activated to maximize the throughput for multi-SPI word transfers. |
| 18 | IS | | Input select |
| | | 0 | Data line 0 (SPI_D[0]) selected for reception. |
| | | 1 | Data line 1 (SPI_D[1]) selected for reception. |
| 17 | DPE1 | | Transmission enable for data line 1 (SPIDATAGZEN[1]) |
| | | 0 | Data line 1 (SPI_D[1]) selected for transmission |
| | | 1 | No transmission on data line 1 (SPI_D[1]) |
| 16 | DPE0 | | Transmission enable for data line 0 (SPIDATAGZEN[0]) |
| | | 0 | Data line 0 (SPI_D[0]) selected for transmission |
| | | 1 | No transmission on data line 0 (SPI_D[0]) |
| 15 | DMAR | | DMA read request. The DMA read request line is asserted when the channel is enabled and new data is available in the receive register of the channel. The DMA read request line is deasserted on read completion of the receive register of the channel. |
| | | 0 | DMA read request is disabled. |
| | | 1 | DMA read request is enabled. |
| 14 | DMAW | | DMA write request. The DMA write request line is asserted when the channel is enabled and the MCSPI_TX*n* register of the channel is empty. The DMA write request line is deasserted on load completion of the MCSPI_TX*n* register of the channel. |
| | | 0 | DMA write request is disabled. |
| | | 1 | DMA write request is enabled. |
| 13-12 | TRM | | Transmit/receive modes. |
| | | 0 | Transmit and receive mode |
| | | 1h | Receive-only mode |
| | | 2h | Transmit-only mode |
| | | 3h | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-32. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 11-7 | WL | | SPI word length. |
| | | 0-2h | Reserved |
| | | 3h | The SPI word is 4-bits long. |
| | | 4h | The SPI word is 5-bits long |
| | | 5h | The SPI word is 6-bits long |
| | | 6h | The SPI word is 7-bits long |
| | | 7h | The SPI word is 8-bits long |
| | | 8h | The SPI word is 9-bits long |
| | | 9h | The SPI word is 10-bits long |
| | | Ah | The SPI word is 11-bits long |
| | | Bh | The SPI word is 12-bits long |
| | | Ch | The SPI word is 13-bits long |
| | | Dh | The SPI word is 14-bits long |
| | | Eh | The SPI word is 15-bits long |
| | | Fh | The SPI word is 16-bits long |
| | | 10h | The SPI word is 17-bits long |
| | | 11h | The SPI word is 18-bits long |
| | | 12h | The SPI word is 19-bits long |
| | | 13h | The SPI word is 20-bits long |
| | | 14h | The SPI word is 21-bits long |
| | | 15h | The SPI word is 22-bits long |
| | | 16h | The SPI word is 23-bits long |
| | | 17h | The SPI word is 24-bits long |
| | | 18h | The SPI word is 25-bits long |
| | | 19h | The SPI word is 26-bits long |
| | | 1Ah | The SPI word is 27-bits long |
| | | 1Bh | The SPI word is 28-bits long |
| | | 1Ch | The SPI word is 29-bits long |
| | | 1Dh | The SPI word is 30-bits long |
| | | 1Eh | The SPI word is 31-bits long |
| | | 1Fh | The SPI word is 32-bits long |
| 6 | EPOL | | $\overline{\text{SPI_SCS}}$[n] polarity |
| | | 0 | $\overline{\text{SPI_SCS}}$[n] is held high during the active state. |
| | | 1 | $\overline{\text{SPI_SCS}}$[n] is held low during the active state. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 22-32. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) Field Descriptions (continued)**

| Bit | Field | Value | Description |
|---|---|---|---|
| 5-2 | CLKD | | Frequency divider for SPI_SCLK. (only when the module is a Master SPI device). A programmable clock divider divides the SPI reference clock (CLKSPIREF) with a 4-bit value, and results in a new clock SPI_SCLK available to shift-in and shift-out data. By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared, Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] register. |
| | | | The value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is cleared to 0. |
| | | 0 | Divide by 1 |
| | | 1h | Divide by 2 |
| | | 2h | Divide by 4 |
| | | 3h | Divide by 8 |
| | | 4h | Divide by 16 |
| | | 5h | Divide by 32 |
| | | 6h | Divide by 64 |
| | | 7h | Divide by 128 |
| | | 8h | Divide by 256 |
| | | 9h | Divide by 512 |
| | | Ah | Divide by 1024 |
| | | Bh | Divide by 2048 |
| | | Ch | Divide by 4096 |
| | | Dh | Divide by 8192 |
| | | Eh | Divide by 16384 |
| | | Fh | Divide by 32768 |
| 1 | POL | | SPI_SCLK polarity |
| | | 0 | SPI_SCLK is held high during the active state |
| | | 1 | SPI_SCLK is held low during the active state |
| 0 | PHA | | SPI_SCLK phase |
| | | 0 | Data are latched on odd numbered edges of SPI_SCLK |
| | | 1 | Data are latched on even numbered edges of SPI_SCLK |

**Table 22-33. Data Lines Configurations**

| | | | TRMi | | |
|---|---|---|---|---|---|
| ISi | DPEi1 | DPEi0 | Transmit and Receive | Receive Only | Transmit Only |
| 0 | 0 | 0 | Supported | Supported | Supported |
| 0 | 0 | 1 | Supported | Supported | Supported |
| 0 | 1 | 0 | Supported | Supported | Supported |
| 0 | 1 | 1 | Not supported (unpredictable result) | Supported | Not supported (unpredictable result) |
| 1 | 0 | 0 | Supported | Supported | Supported |
| 1 | 0 | 1 | Supported | Supported | Supported |
| 1 | 1 | 0 | Supported | Supported | Supported |
| 1 | 1 | 1 | Not supported (unpredictable result) | Supported | Not supported (unpredictable result) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.13 McSPI Channel (*i*) Status Register (MCSPI_CH(i)STAT)

The McSPI channel *i* status register register (MCSPI_CH(i)STAT) provides status information about the McSPI channel *i* FIFO transmit buffer register (MCSPI_TX*n*) and the McSPI channel *i* FIFO receive buffer register (MCSPI_RX*n*) of channel *i*. The (MCSPI_CH(i)STAT) is shown in Figure 22-37 and described in Table 22-34.

**Figure 22-37. McSPI Channel *(i)* Status Register (MCSPI_CH(i)STAT)**

| 31 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | RXFFF | RXFFE | TXFFF | TXFFE | EOT | TXS | RXS |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 22-34. McSPI Channel *(i)* Status Register (MCSPI_CH(i)STAT) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-7 | Reserved | 0 | Read returns 0 |
| 6 | RXFFF | | Channel *i* FIFO receive buffer full status. |
| | | 0 | FIFO receive buffer is not full. |
| | | 1 | FIFO receive buffer is full. |
| 5 | RXFFE | | Channel *i* FIFO receive buffer empty status. |
| | | 0 | FIFO receive buffer is not empty. |
| | | 1 | FIFO receive buffer is empty. |
| 4 | TXFFF | | Channel *i* FIFO transmit buffer full status. |
| | | 0 | FIFO transmit buffer is not full. |
| | | 1 | FIFO transmit buffer is full. |
| 3 | TXFFE | | Channel *i* FIFO transmit buffer empty status. |
| | | 0 | FIFO transmit buffer is not empty. |
| | | 1 | FIFO transmit buffer is empty. |
| 2 | EOT | | Channel *i* end-of-transfer status. The definitions of beginning and end of transfer vary with master versus slave and the transfer format (transmit/receive mode, turbo mode). |
| | | 0 | This flag is automatically cleared when the shift register is loaded with the data from the transmitter register (beginning of transfer). |
| | | 1 | This flag is automatically set to one at the end of an SPI transfer. |
| 1 | TXS | | Channel *i* transmitter register status. The bit is cleared when the host writes the most significant byte of the SPI word in the MCSPI_TX(i) register. The bit is set when enabling the channel *i* , and also when the SPI word is transferred from the MCSPI_TX(i) register to the shift register. |
| | | 0 | Register is full. |
| | | 1 | Register is empty. |
| 0 | RXS | | Channel *i* receiver register status. The bit is cleared when enabling the channel *i*, and also when the host reads the most significant byte of the received SPI word from the MCSPI_RX(i) register. The bit is set when the received SPI word is transferred from the shift register to the MCSPI_RX(i) register. |
| | | 0 | Register is empty. |
| | | 1 | Register is full. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.14 McSPI Channel (i) Control Register (MCSPI_CH(I)CTRL)

The McSPI channel *i* control register (MCSPI_CH*i*CTRL) is used to enable channel *i*. The MCSPI_CH*i*CTRL is shown in Figure 22-38 and described in Table 22-35.

**Figure 22-38. McSPI Channel (*i*) Control Register (MCSPI_CH(I)CTRL)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 8 | 7 | 1 | 0 |
|---|---|---|---|---|
| EXTCLK | | Reserved | | EN |
| R/W-0 | | R/W-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 22-35. McSPI Channel (*i*) Control Register (MCSPI_CH(I)CTRL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | Reserved | 0 | Reserved |
| 15-8 | EXTCLK | | Clock ratio extension. Used to concatenate with the CLKD bit field in MCSPI_CH*n*CONF for clock ratio only when granularity is 1 clock cycle (CLKG bit in MCSPI_CH*n*CONF set to 1). Then the maximum value reached is a 4096 clock divider ratio. |
| | | 0 | Clock ratio is CLKD + 1 |
| | | 1h | Clock ratio is CLKD + 1 + 16 |
| | | ... | ... |
| | | FFh | Clock ratio is CLKD + 1 + 4080 |
| 7-1 | Reserved | 0 | Reserved |
| 0 | EN | | Channel *n* enable. |
| | | 0 | Channel *n* is not active. |
| | | 1 | Channel *n* is active. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.15 McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*)

The McSPI channel *i* transmit register (MCSPI_TX*(i)*) contains a single McSPI word to transmit on the serial link. The (MCSPI_TX*(i)*) is shown in Figure 22-39 and described in Table 22-36.

- Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h.
- The SPI words are transferred with MSB first.

**Figure 22-39. McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*)**

| 31 | 0 |
|---|---|
| TDATA | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 22-36. McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | TDATA | 0-FFFF FFFFh | Channel *i* data to transmit. |

### 22.3.16 McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*)

The McSPI channel *i* FIFO receive buffer register (MCSPI_RX*(i)*) contains a single McSPI word received through the serial link. The (MCSPI_RX*(i)*) is shown in Figure 22-40 and described in Table 22-37.

- Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h.

**Figure 22-40. McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*)**

| 31 | 0 |
|---|---|
| RDATA | |
| R-0 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 22-37. McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | RDATA | 0-FFFF FFFFh | Channel *i* received data. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.17 McSPI Transfer Levels Register (MCSPI_XFERLEVEL)

The McSPI transfer levels register (MCSPI_XFERLEVEL) provides the transfer levels needed while using the FIFO buffer during transfer. The MCSPI_XFERLEVEL is shown in Figure 22-41 and described in Table 22-38.

**Figure 22-41. McSPI Transfer Levels Register (MCSPI_XFERLEVEL)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WCNT | | | | | | | | | | | | | | | |
| R/W-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AFL | | | | | | | | AEL | | | | | | | |
| R/W-0 | | | | | | | | R/W-0 | | | | | | | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 22-38. McSPI Transfer Levels Register (MCSPI_XFERLEVEL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-16 | WCNT | | SPI word counter. Holds the programmable value of the number of SPI words to be transferred on the channel that is using the FIFO buffer. When the transfer has started, a read back of this register returns the current SPI word transfer index. |
| | | 0 | Counter not used |
| | | 1 | 1 SPI word |
| | | ... | ... |
| | | FFFEh | 65534 SPI word |
| | | FFFFh | 65535 SPI word |
| 15-8 | AFL | | Buffer almost full. Holds the programmable almost full level value used to determine almost full buffer condition. If you want an interrupt or a DMA read request to be issued during a receive operation when the data buffer holds at least *n* bytes, then the buffer MCSPI_XFERLEVEL[AFL] must be set with n - 1. |
| | | 0 | 1 byte |
| | | 1 | 2 bytes |
| | | ... | ... |
| | | FFh | 256 bytes |
| 7-0 | AEL | | Buffer almost empty. Holds the programmable almost empty level value used to determine almost empty buffer condition. If you want an interrupt or a DMA write request to be issued during a transmit operation when the data buffer is able to receive *n* bytes, then the buffer MCSPI_XFERLEVEL[AEL] must be set with n - 1. |
| | | 0 | 1 byte |
| | | 1 | 2 bytes |
| | | ... | ... |
| | | FFh | 256 bytes |


Copyright © 2011–2016, Texas Instruments Incorporated

### 22.3.18 DMA Address Aligned FIFO Transmitter Register (MCSPI_DAFTX)

The DMA address aligned FIFO transmitter register (MCSPI_DAFTX) contains the SPI words to transmit on the serial link when the FIFO used and the DMA address is aligned on 256 bits. MCSPI_DAFTX is an image of one of MCSPI_TX*(i)* register corresponding to the channel that has its FIFO enabled. The MCSPI_DAFTX is shown in Figure 22-42 and described in Table 22-39.

- Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h.
- The SPI words are transferred with MSB first.

**Figure 22-42. DMA Address Aligned FIFO Transmitter Register (MCSPI_DAFTX)**

| 31 | 0 |
|---|---|
| DAFTDATA | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 22-39. DMA Address Aligned FIFO Transmitter Register (MCSPI_DAFTX) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | DAFTDATA | 0-FFFF FFFFh | FIFO data to transmit with DMA 256 bit aligned address. This register is only is used when MCSPI_MODULCTRL[FDAA] is set to 1 and only one of the MCSPI_CH(i)CONF[FFEW] of enabled channels is set. If these conditions are not respected, any access to this register returns a null value. |

### 22.3.19 DMA Address Aligned FIFO Receiver Register (MCSPI_DAFRX)

The DMA address aligned FIFO receiver register (MCSPI_DAFRX) contains the SPI words to receive on the serial link when the FIFO used and the DMA address is aligned on 256 bits. MCSPI_DAFRX is an image of one of MCSPI_RX*(i)* register corresponding to the channel that has its FIFO enabled. The MCSPI_DAFRX is shown in Figure 22-43 and described in Table 22-40.

- Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h.

**Figure 22-43. DMA Address Aligned FIFO Receiver Register (MCSPI_DAFRX)**

| 31 | 0 |
|---|---|
| DAFRDATA | |
| R/W-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 22-40. DMA Address Aligned FIFO Receiver Register (MCSPI_DAFRX) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-0 | DAFRDATA | 0-FFFF FFFFh | FIFO received data with DMA 256 bit aligned address. This register is only is used when MCSPI_MODULCTRL[FDAA] is set to 1 and only one of the MCSPI_CH(i)CONF[FFER] of enabled channels is set. If these conditions are not respected, any access to this register returns a null value. |


Copyright © 2011–2016, Texas Instruments Incorporated

# Timers

This chapter describes the operation of the software-programmable timer.

**Topic** **Page**

**23.1 Introduction** ..... **3553**
**23.2 Architecture** ..... **3556**


Copyright © 2011–2016, Texas Instruments Incorporated

## 23.1 Introduction

### 23.1.1 Overview

The timer module contains a free running upward counter with auto reload capability on overflow. The timer counter can be read and written in real-time (while counting). The timer module includes compare logic to allow an interrupt event on a programmable counter matching value.

A dedicated output signal can be pulsed or toggled on overflow and a matched event. This output can be used as a timing stamp trigger signal or PWM (pulse-width modulation) signal source. An additional dedicated output signal (PORGPOCFG) can be used for general-purpose IO, directly driven by bit 14 of the Timer Control Register (TCLR). Based on the programmable input signal transition type, a dedicated signal can be used to trigger an automatic timer counter capture event and generate an interrupt. A programmable clock divider (prescaler) allows reduction of the timer input clock frequency. All internal timer interrupt sources are merged in one module interrupt line and one wake-up line. Each internal interrupt source can be independently enabled or disabled.

This module is controllable through the OCP peripheral bus.

As two clock domains are managed inside this module, resynchronization is done by special logic between the OCP clock domain and the Timer clock domain. At reset, synchronization logic allows utilization of all ratios between the OCP clock and the Timer clock. A drawback of this mode is that the full-resynchronization path is used with access latency performance impact in terms of OCP clock cycles. In order to improve module access latency, and under restricted conditions on clock ratios, write-posted mode can be used by setting the POSTED bit of the Timer Synchronous Interface Control Register (TSICR). Under posted mode, the OCP write command is granted before the write process is complete in the timer clock domain. This mode allows software to do concurrent writes on Dual Mode timer registers and to observe write process completion (synchronization) at the software level by reading independent write posted status bits in the Write Posted Status Register (TWPS).

### 23.1.2 Features

The timer consists of the following features:

- Counter timer with compare and capture modes
- Auto-reload mode
- Start-stop mode
- Programmable divider clock source
- 16-32 bit addressing
- “On the fly” read/write registers
- Interrupts generated on overflow, compare and capture
- Interrupt enable
- Wake-up enable
- Write posted mode
- Dedicated input trigger for capture mode and dedicated output trigger/PWM signal
- Dedicated output signal for general purpose use PORGPOCFG
- OCP interface compatible

The Timer resolution and interrupt period are dependent on the selected input clock and clock prescale value. Example resolutions for common clock values are shown in Table 23-1.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 23-1. Timer Resolution and Maximum Range**

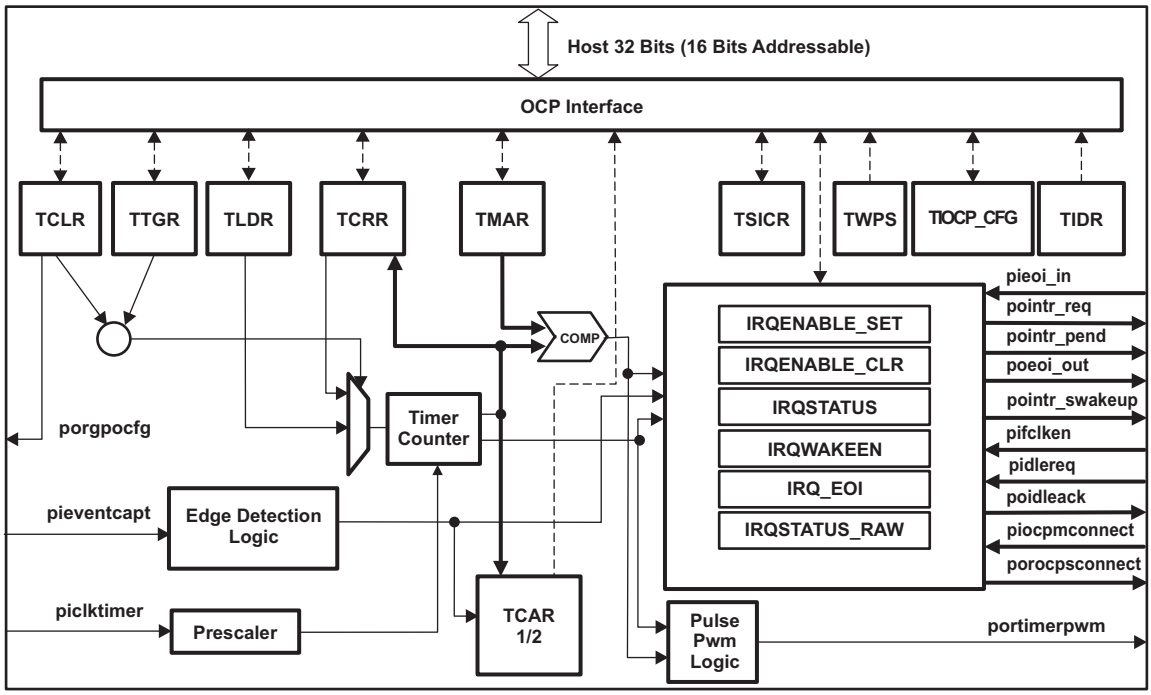
| Clock | Prescaler | Resolution | Interrupt Period Range[1] |
|---|---|---|---|
| 32 KHz | 1 (min) | 31.25 us | 62.5 us to ~37h 17m |
| | 256 (max) | 8 ms | 16 ms to ~397d 16h 22m |
| 27 MHz | 1 (min) | ~37 ns | ~74 ns to ~159s |
| | 256 (max) | ~9.48 us | ~18.9 us to ~11h 18m |
| 38.4 MHz | 1 (min) | ~26 ns | ~52 ns to ~112 s |
| | 256 (max) | ~6.7 us | ~13.3 us to ~7h 57m |

(1) Minimum calculation adheres to recommendation of 2 clock cycle minimum between interrupts.

### 23.1.3 Functional Block Diagram

Figure 23-1 shows a block diagram of the timer.

**Figure 23-1. Timer Block Diagram**





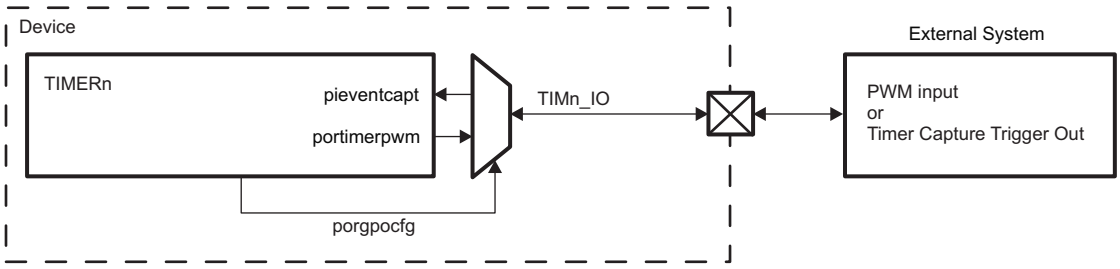
Copyright © 2011–2016, Texas Instruments Incorporated

### 23.1.4 GP Timer External System Interface

The general-purpose (GP) timer can send or receive stimulus to/from the external (off-chip) system. In the device, however, only certain timers are configured to output a PWM pulse or receive an external event signal used as a trigger to capture the current timer count. See the device-specific data manual details on which timers have IO capabilities.

Figure 23-2 shows the external system interface for the GP timers.

**Figure 23-2. GP Timers External System Interface**




**NOTE:** For the timers with IO capabilities, the PORGPOCFG output signal selects the timer PWM output or the timer capture input signal to the TIMn_IO pad at the top level. When TCLR[GPO_CFG] = 1, TIMn_IO functions as a capture input. When TCLR[GPO_CFG] = 0, TIMn_IO functions as a PWM output.


Copyright © 2011–2016, Texas Instruments Incorporated

## 23.2 Architecture

### 23.2.1 Functional Description

The general-purpose timer is an upward counter. It supports 3 functional modes:

- Timer mode
- Capture mode
- Compare mode

By default, after core reset, the capture and compare modes are disabled.

#### 23.2.1.1 Timer Mode Functionality

The timer is an upward counter that can be started and stopped at any time through the ST bit of the Timer Control Register (TCLR). The Timer Counter Register (TCRR) can be loaded when stopped or on the fly (while counting). TCRR can be loaded directly by a TCRR write access with the new timer value. TCRR can also be loaded with the value held in the Timer Load Register (TLDR), due to a write access to the Timer Trigger Register (TTGR). TCRR is loaded regardless of the TTGR written value. The TCRR value can be read when stopped or captured on-the-fly by a TCRR read access. The timer is stopped and the counter value is cleared to “0” when the module’s reset is asserted. The timer is maintained in a stopped state after reset is released. When the timer is stopped, TCRR does not increment. The counter can be restarted from the frozen value unless TCRR has been reloaded with a new value.

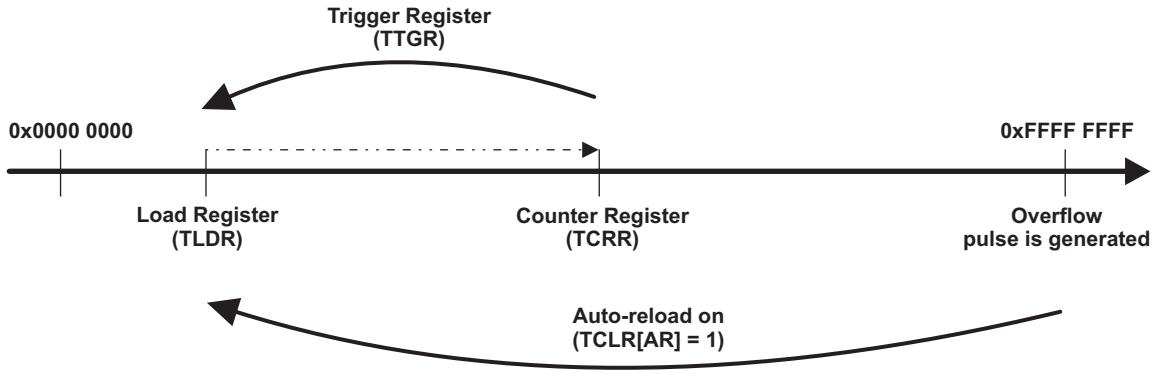
When the one-shot mode is enabled (TCLR[AR] = 0), the counter is stopped after a counting overflow (counter value remains at zero).

When the auto-reload mode is enabled (TCLR[AR] = 1), TCRR is reloaded with the Timer Load Register (TLDR) value after a counting overflow.

It is not recommended to put the overflow value (FFFF FFFFh) in TLDR because it can lead to undesired results.

An interrupt can be issued on overflow, if the overflow interrupt enable bit is set in the Timer IRQENABLE Set Register (IRQENABLE_SET[OVF_EN_FLAG] = 1). In addition, a dedicated output pin can be used to generate one positive pulse (piclktimer duration) or to invert the current value (toggle mode) when an overflow occurs. Refer to Section 23.2.1.5 for more details related to the dedicated pulse-width modulation (PWM) output pin.

**Figure 23-3. TCRR Timing Value**





Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.1.2 Capture Mode Functionality

The timer value in TCRR can be captured and saved in TCAR1 or TCAR2 function of the mode selected in TCLR through the field CAPT_MODE when a transition is detected on the module input pin (PIEVENTCAPT). The edge detection circuitry monitors transitions on the input pin (PIEVENTCAPT).

Rising transition, falling transition or both can be selected in TCLR (TCM bit) to trig the timer counter capture. The module sets the IRQSTATUS (TCAR_IT_FLAG bit) when an active transition is detected and at the same time the counter value TCRR is stored in one of the timer capture registers TCAR1 or TCAR2 as follows:

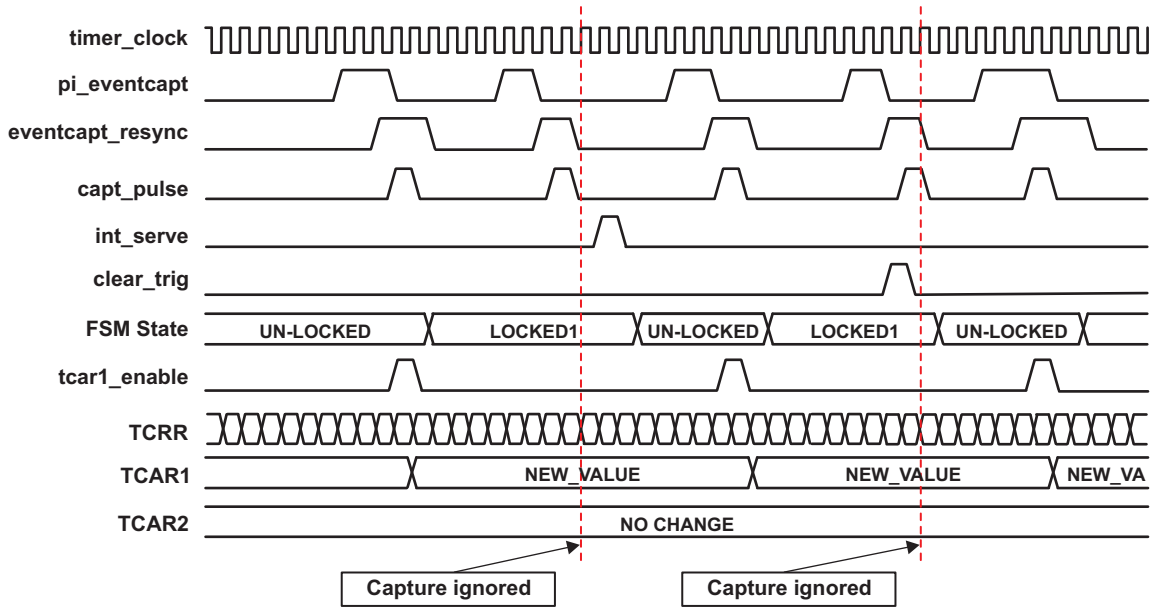
- If TCLR's CAPT_MODE field is 0 then, on the first enabled capture event, the value of the counter register is saved in TCAR1 register and all the next events are ignored (no update on TCAR1 and no interrupt triggering) until the detection logic is reset or the interrupt status register is cleared on TCAR's position writing a 1 in it.
- If TCLR's CAPT_MODE field is 1 then, on the first enabled captured event, the counter value is saved in TCAR1 register and, on the second enabled capture event, the value of the counter register is saved in TCAR2 register. All the other events are ignored (no update on TCAR1/2 and no interrupt triggering) until the detection logic is reset or the interrupt status register is cleared on TCAR's position writing a 1 in it. This mechanism is useful for period calculation of a clock if that clock is connected to the PIEVENTCAPT input pin.

The edge detection logic is reset (a new capture is enabled) when the active capture interrupt is served - TCAR_IT_FLAG bit of IRQSTATUS (previously 1) is cleared. The timer functional clock (input to prescaler) is used to sample the input pin (PIEVENTCAPT). Input negative or positive pulse can be detected when pulse time is above functional clock period. An interrupt can be issued on transition detection if the capture interrupt enable bit is set in the Timer Interrupt Enable Register IRQENABLE_SET (TCAR_IT_FLAG bit).

In Figure 23-4, the TCM value is 01 and CAPT_MODE is 0 - only rising edge of the PIEVENTCAPT will trigger a capture in TCAR and only TCAR1 will update.

In Figure 23-5, the TCM value is 01 and CAPT_MODE is 1 - only rising edge of the PIEVENTCAPT will trigger a capture in TCAR1 on first enabled event and TCAR2 will update on the second enabled event.

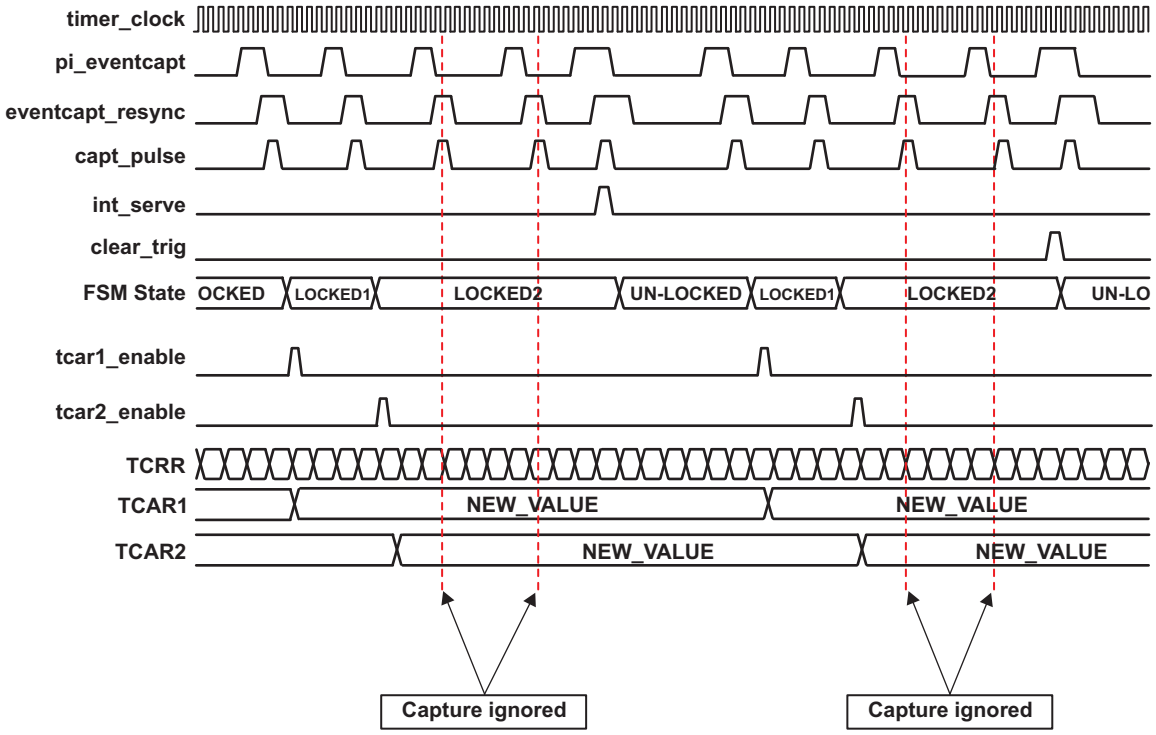
**Figure 23-4. Capture Wave Example for CAPT_MODE = 0**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 23-5. Capture Wave Example for CAPT_MODE = 1**




### 23.2.1.3 Compare Mode Functionality

When the Compare Enable bit of the Timer Control Register (TCLR[CE]) is set to 1, the timer value in the Timer Counter Register (TCRR) is permanently compared to the value held in the Timer Match Register (TMAR). The TMAR value can be loaded at any time (timer counting or stopped). When the TCRR and the TMAR values match, an interrupt can be issued if the match interrupt enable bit is set in the Timer IRQENABLE Set Register (IRQENABLE_SET[MAT_EN_FLAG] = 1). Software should write a compare value in the TMAR register before setting the TCLR[CE] bit to avoid any unwanted interrupt due to a reset value matching effect.

A dedicated output pin can be used to generate one positive pulse (piclktimer duration) or to invert the current value (toggle mode) when an overflow and a match occur. Refer to Section 23.2.1.5 for more details related to the dedicated pulse-width modulation (PWM) output pin.

### 23.2.1.4 Prescaler Functionality

A prescaler counter can be used to divide the timer counter input clock frequency. The prescaler is enabled when the PRE bit of the Timer Control Register (TCLR[PRE]) is set. The $2^n$ division ratio value (PTV) can be configured in the TCLR register. The prescaler counter is reset when the timer counter is stopped or reloaded on-the-fly. Refer to Section 23.2.1.6 for more details on the prescaler clock ratio values.

**Table 23-2. Prescaler Functionality**

| Contexts | Prescaler Counter | Timer Counter |
|---|---|---|
| Overflow (when Auto-reload on) | Reset | TLDR |
| TCRR Write | Reset | TCRR |
| TTGR Write | Reset | TLDR |
| Stop | Reset | Frozen |


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.1.5 Pulse-Width Modulation

The timer can be configured to provide a programmable pulse-width modulation (PWM) output on a dedicated output pin (PORTIMERPWM). The PORTIMERPWM output pin can be configured to toggle on a specified event. The TRG field of the Timer Control Register (TCLR[TRG]) determines the trigger event when the PORTIMERPWM pin toggles. Either overflow or the combination of both overflow and match can be used to toggle the PORTIMERPWM pin.

When both overflow and match are configured to trigger the PWM pin, the matched event will be ignored from the moment the mode was set-up until the first overflow event occurs (see Figure 23-7).

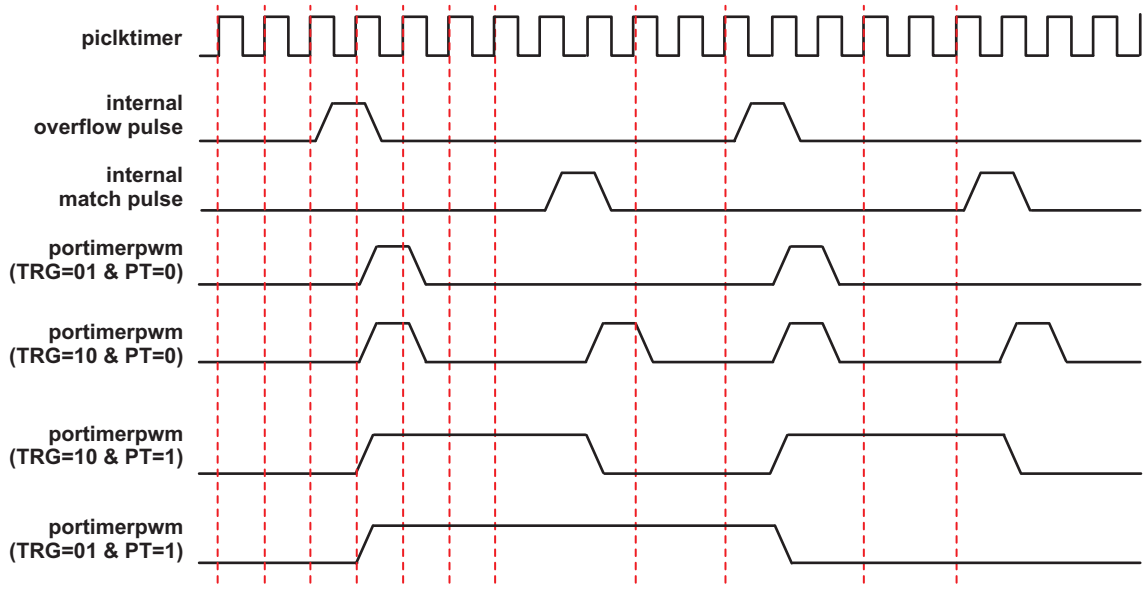
The SCPWM bit in the TCLR register can be programmed to set or clear the PORTIMERPWM output signal only while the counter is stopped or the triggering is disabled. This allows fixing a deterministic state of the output pin while modulation is stopped. The modulation is synchronously stopped when the TRG bit is cleared and overflow occurs.

In the following timing diagrams, the internal overflow pulse is set each time an overflow occurs in the Timer Counter Register (TCRR), and the internal match pulse is set when the counter reaches the Timer Match Register (TMAR) value. Depending on the configured value of the TRG and PT bits of the TCLR register, the timer generates one pulse or inverts the current value on the output pin (PORTIMERPWM).

The Timer Load Register (TLDR) and Timer Match Register (TMAR) must keep values smaller than the overflow value (FFFF FFFFh) with at least 2 units. In case the PWM trigger events are both overflow and match, the difference between the values kept in the TLDR and TMAR registers must be at least 2 units. When match event is used, the CE bit of the TCLR register must be set.

In Figure 23-6, TCLR[SCPWM] is cleared to 0. In Figure 23-7, TCLR[SCPWM] is set to 1.

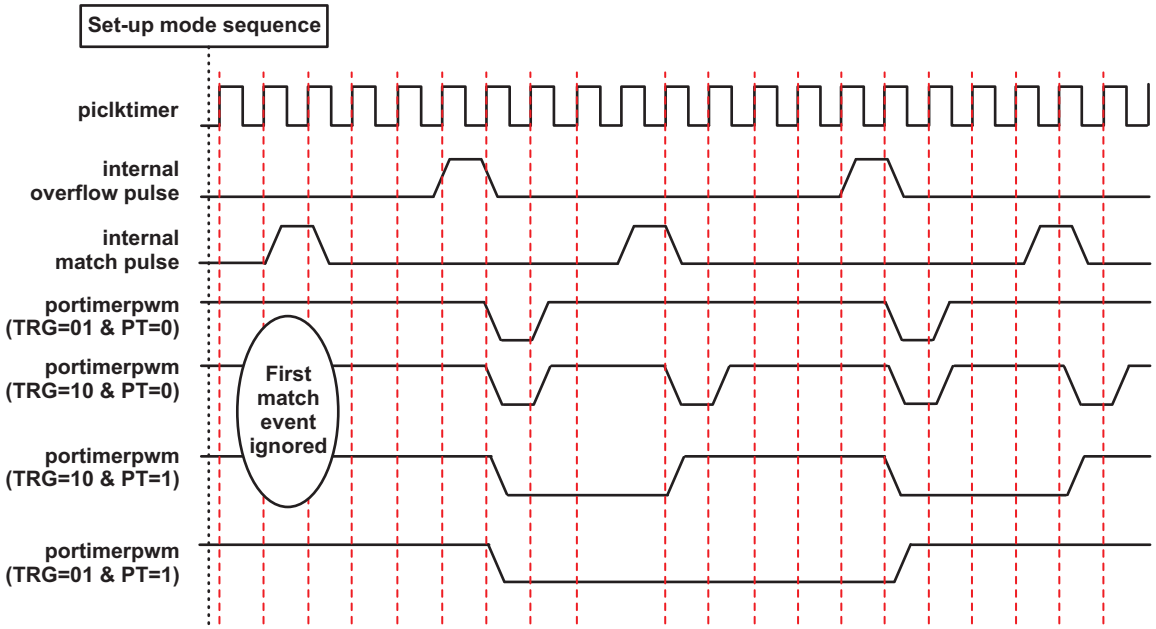
**Figure 23-6. Timing Diagram of Pulse-Width Modulation with SCPWM = 0**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 23-7. Timing Diagram of Pulse-Width Modulation with SCPWM = 1**





Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.1.6 Timer Counting Rate

The timer counter is composed of a prescaler stage and a timer counter. The prescaler stage is clocked with the timer input clock and acts as a clock divider for the timer counter stage. The ratio can be managed by accessing the ratio definition field of the control register (PTV and PRE of TCLR). See Table 23-3.

The timer rate is defined by:

- The value of the prescaler fields (PRE and PTV of TCLR register)
- The value loaded into the Timer Load Register (TLDR).

**Table 23-3. Prescaler Clock Ratios Value**

| PRE | PTV | Divisor (PS) |
|---|---|---|
| 0 | X | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 4 |
| 1 | 2 | 8 |
| 1 | 3 | 16 |
| 1 | 4 | 32 |
| 1 | 5 | 64 |
| 1 | 6 | 128 |
| 1 | 7 | 256 |

The timer rate equation is as follows:

*(FFFF FFFFh – TLDR + 1) × timer Clock period × Clock Divider (PS)*

With timer Clock period = 1/ timer Clock frequency and PS = $2^{(PTV + 1)}$.

As an example, if we consider a timer clock input of 32 kHz, with a PRE field equal to 0, the timer output period is:

**Table 23-4. Value and Corresponding Interrupt Period**

| TLDR | Interrupt period |
|---|---|
| 0000 0000h | ~37 h |
| FFFF 0000h | ~2 s |
| FFFF FFF0h | 500 µs |
| FFFF FFFEh | 62.5 µs |

### 23.2.1.7 Dual Mode Timer Under Emulation

During emulation mode (when PINSUSPENDN signal is active), the timer can/cannot continue running according to the value of the EmuFree bit of the timer OCP configuration register (TIOCP_CFG).

If EmuFree is 1, timer execution is not stopped and, regardless of the value of PINSUSPENDN signal, and the interrupt assertion is still generated when overflow is reached.

If EmuFree is 0, counters (prescaler/timer) are frozen and an increment start occurs again as soon as PINSUSPENDN becomes inactive. The asynchronous input pin is internally synchronized on 2 TIMER clock rising edges.


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.2 Accessing Registers

All registers are 32-bit wide, accessible via OCP interface with 16-bit or 32-bit OCP access (Read/Write). The 32-bit registers write update in 16 bits access must be LSB16 first and the second write access must be MSB16. For the write operation, the module allows skipping the MSB access if the user does not need to update the 16 MSB bits of the register, but only for the OCP registers (TIDR, TIOCP_CFG, IRQ_EOI, IRQSTATUS_RAW, IRQSTATUS, IRQENABLE_SET, IRQENABLE_CLR, IRQWAKEEN and TSICR). The write operation on any functional register (TCLR, TCRR, TLDR, TTGR and TMAR) must be complete (the MSB must be written even if the MSB data is not used).

#### 23.2.2.1 Programming the Timer Registers

The TLDR, TCRR, TCLR, TIOCP_CFG, IRQ_EOI, IRQSTATUS, IRQENABLE_SET, IRQENABLE_CLR, IRQWAKEEN, TTGR, TSICR and TMAR registers write is done synchronously with OCP clock, by the host, using the OCP bus protocol.

#### 23.2.2.2 Reading the Timer Registers

The counter register (TCRR) is a 32-bit “atomic datum” and 16-bit capture is done on the 16-bit LSB first to allow atomic LSB16 + MSB16 capture. Atomic capture is also performed for the TCAR1 and TCAR2 registers as they may change due to internal processes. DSP 16 bit accesses can be interleaved with MCU 32 bit accesses.

#### 23.2.2.3 OCP Error Generation

The timer module responds with error indication in the following cases:

**Error on write transactions**

- Assert the PORSRESP = ERR signal in the same cycle as PORSCMDACCEPTED.
- Use the ERR code for PORSRESP during the response phase.

**Error on read transactions**

- Assert the PORSRESP = ERR signal in the same cycle as PORSCMDACCEPTED.
- Use the ERR code for PORSRESP during the response phase. PORSDATA in this case is not valid.

**Table 23-5. OCP Error Reporting**

| Error Type | Response: SRESP = ERR |
|---|---|
| Unsupported PIOCPMCMD command | Yes |
| Address error: Read or write to a non-existing internal address | No |
| Read to write-only registers and write to read-only registers | No |
| Unaligned address (PIOCPMADDR ≠ 00) on read/write transaction | Yes |
| Unsupported PIOCPMBYTEEN on read/write transaction | Yes |

**NOTE:** Byte enable “0000” is a supported byte enable.

### 23.2.3 Posted Mode Selection

A choice between the two synchronization modes will be made taking into account the frequency ratio and the stall periods that can be supported by the system, without impacting the global performance.

The posted mode selection applies only to functional registers that require synchronization on/from timer clock domain. For write operation the registers affected by this posted/non-posted selection are: TCLR, TLDR, TCRR, TTGR and TMAR. For read operation the register affected by this posted/non-posted selection are: TCRR, TCAR1 and TCAR2.


Copyright © 2011–2016, Texas Instruments Incorporated

The OCP clock domain synchronous registers TIDR, TIOCP_CFG, TISTAT, IRQ_EOI,IRQSTATUS,IRQSTATUS_RAW, IRQENABLE_SET, IRQENABLE_CLR, IRQWAKEEN, TWPS and TSICR are not affected by the posted/non-posted mode selection; the write/read operation is effective and acknowledged (command accepted) after one OCP clock cycle from the command assertion.

The configuration posted/non-posted is made by setting the PIFREQRATIO when the module is integrated. The PIFREQRATIO signal should be tied to '1' when the freq (timer) < freq (OCP)/4 frequency, and tied to 0 when it is the opposite frequency ratio. The PIFREQRATIO represent the reset value of the TSICR (POSTED bit). The configuration can be changed (overwritten) by software, writing the TSICR (POSTED bit) register.

The following cases are possible:

- Posted Mode can be used when the functional frequency range is: freq (timer) < freq (OCP)/4.
- Non-Posted Mode can be used regardless of the frequency range. Recommended frequency is: freq (timer) >= freq (OCP)/4.

**NOTE:** The Non-Posted Mode can be also used when freq (timer) < freq (OCP)/4, but it is recommended to use the Posted Mode. Using Non-Posted Mode will delay the command accept and the transaction latency will be as described in the below chapters. Posted mode offers an OCP interface latency improvement and can be used only if the frequencies respect the freq (timer) < freq (OCP)/4 formula.

### *23.2.4 Write Registers Access*

#### 23.2.4.1 Write Posted

This mode can be used only if the functional frequency range is freq (timer) < freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is set to 1 in the timer control register.

This mode uses a posted-write scheme to update any internal register. The write transaction is immediately acknowledged on the OCP interface, although the effective write operation will occur later, due to a resynchronisation in the timer clock domain. This has the advantage of not stalling either the interconnect system, or the CPU that requested the write transaction. For each register, a status bit is provided, that is set if there is a pending write access to this register.

In this mode, it is mandatory that the CPU checks the status bit prior to any write access. In case a write is attempted to a register with a previous access pending, the previous access is discarded without notice (this can lead to unexpected results also).

There is one status bit per register, accessible in the Timer Write Posted Status Register. When the timer module operates in this mode, there is an automatic sampling of the current timer counter value, in an OCP-synchronized capture register. Consequently, any read access to the timer counter register does not add any re-synchronization latency; the current value is always available.

A register read following a write posted register (on the same register) is not insured to read the previous write value if the write posted process is not completed. Software synchronization should be used to avoid non-coherent read.

The drawback of this automatic update mechanism is that it assumes a given relationship between the OCP interface frequency and the timer functional frequency.

This posted period is defined as the interval between the posted write access request and the reset of the posted bit in TWPS register, and can be quantified:

T (reset posted max.) = 3 OCP clock + 2.5 TIMER clock

The time when the write accomplishes is:

T (write accomplish) = 1 OCP clock + 2.5 TIMER clock


Copyright © 2011–2016, Texas Instruments Incorporated

#### 23.2.4.2 Write Non-Posted

This mode is functional regardless of the ratio between the OCP interface frequency and the functional clock frequency. Recommended functional frequency range is freq (timer) >= freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is cleared to 0 in the timer control register.

This mode uses a non posted-write scheme to update any internal register. That means the write transaction will not be acknowledged on the OCP interface, until the effective write operation occurs, after the resynchronisation in the timer clock domain. The drawback is that both the interconnect system and the CPU are stalled during this period.

- The latency of the interrupt serving is increased, as the interconnect system and CPU are stalled.
- An interconnect logic, including time-out logic to detect erroneous transactions, can generate an unwanted system abort event.

The stall period is defined as the interval between the non-posted write access request and the rise of the command accept signal and can be quantified:

T (stall max.) = 3 OCP clock + 2.5 TIMER clock

The time when the write accomplishes is:

T (write accomplish) = 1 OCP clock + 2.5 TIMER clock

A register read following a write to the same register is always coherent.

### *23.2.5 Read Registers Access*

#### 23.2.5.1 Read Posted

This mode can be used only if the functional frequency range is freq (timer) < freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is set to 1 in the timer control register.

This mode uses a posted-read scheme, for reading any internal register. The read transaction is immediately acknowledged on the OCP interface, and the value to be read has been previously resynchronised. This has the advantage of not stalling either the interconnect system, or the CPU that requested the read transaction.

#### 23.2.5.2 Read Non-Posted

This mode is functional whatever the ratio between the OCP interface frequency and the functional clock frequency. Recommended functional frequency range is freq (timer) >= freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is cleared to 0 in the timer control register.

This mode uses a non posted-read scheme, for reading any internal register. The read transaction will not be acknowledged on the OCP interface, until the effective read operation occurs, after the resynchronisation in the timer clock domain. The drawback is that both the interconnect system and the CPU are stalled during this period.

- The latency of the interrupt serving is increased, as the interconnect system and the CPU are stalled.
- An interconnect system including time-out logic to detect erroneous transactions can generate an unwanted system abort event.

This mode only applies only to three registers: TCRR, TCAR1 and TCAR2, which need resynchronisation from functional to OCP clock domains.

The stall period is defined as the interval between the non-posted read access request and the rise of the command accept signal and can be quantified:

T (stall max.) = 3 OCP clock + 2.5 TIMER clock

The time when the value is sampled is:

T (read sample) = 1 OCP clock + 2.5 TIMER clock


Copyright © 2011–2016, Texas Instruments Incorporated

## Timer Registers

The timer registers are listed in Table 23-6. All registers are:

- 32-bit register accessible in 16-bit mode
- Little-endian addressing

**Table 23-6. Timer Registers**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 00h | TIDR | Identification Register | Section 23.2.6 |
| 10h | TIOCP_CFG | Timer OCP Configuration Register | Section 23.2.7 |
| 20h | IRQ_EOI | Timer IRQ End-Of-Interrupt Register | Section 23.2.8 |
| 24h | IRQSTATUS_RAW | Timer IRQSTATUS Raw Register | Section 23.2.9 |
| 28h | IRQSTATUS | Timer IRQSTATUS Register | Section 23.2.10 |
| 2Ch | IRQENABLE_SET | Timer IRQENABLE Set Register | Section 23.2.11 |
| 30h | IRQENABLE_CLR | Timer IRQENABLE Clear Register | Section 23.2.12 |
| 34h | IRQWAKEEN | Timer IRQ Wakeup Enable Register | Section 23.2.13 |
| 38h | TCLR | Timer Control Register | Section 23.2.14 |
| 3Ch | TCRR | Timer Counter Register | Section 23.2.15 |
| 40h | TLDR | Timer Load Register | Section 23.2.16 |
| 44h | TTGR | Timer Trigger Register | Section 23.2.17 |
| 48h | TWPS | Timer Write Posted Status Register | Section 23.2.18 |
| 4Ch | TMAR | Timer Match Register | Section 23.2.19 |
| 50h | TCAR1 | Timer Capture Register | Section 23.2.20 |
| 54h | TSICR | Timer Synchronous Interface Control Register | Section 23.2.21 |
| 58h | TCAR2 | Timer Capture Register | Section 23.2.22 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.6 TIDR Register

This read only register contains the revision number of the module. A write to this register has no effect. This register is used by software to track features, bugs, and compatibility.

**Figure 23-8. TIDR Register**

| 31 | 30 | 29 | 28 | 27 | 16 |
|---|---|---|---|---|---|
| SCHEME | | Reserved | | FUNC | |
| R-1 | | R-0 | | R-FFFh | |

| 15 | 11 | 10 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| R_RTL | | X_MAJOR | | CUSTOM | | Y_MINOR | |
| R-2h | | R-3h | | R-0 | | R-1 | |

LEGEND: R = Read only; -*n* = value after reset

**Table 23-7. TIDR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | SCHEME | R | 1 | Used to distinguish between old scheme and current. |
| 29-28 | Reserved | R | 0 | Reads return 0 |
| 27-16 | FUNC | R | FFFh | Function indicates a software compatible module family. |
| 15-11 | R_RTL | R | 2h | RTL Version (R). |
| 10-8 | X_MAJOR | R | 3h | Major Revision (X). |
| 7-6 | CUSTOM | R | 0 | Indicates a special version for a particular device. |
| 5-0 | Y_MINOR | R | 1 | Minor Revision (Y). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.7 TIOCP_CFG Register

This register allows controlling various parameters of the OCP interface.

**Figure 23-9. TIOCP_CFG Register**

| 31 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| Reserved | | IDLEMODE | | EMUFREE | SOFTRESET |
| R-0 | | R/W-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 23-8. TIOCP_CFG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | Reserved | R | 0 | Reserved |
| 3-2 | IDLEMODE | R/W | 0 | Power management, req/ack control<br>0 = Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only.<br>1h = No-idle mode: local target never enters idle state. Backup mode, for debug only.<br>2h = Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events.<br>3h = Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented. |
| 1 | EMUFREE | R/W | 0 | Emulation mode<br>0 = The timer is frozen in emulation mode (PINSUSPENDN signal active).<br>1 = The timer runs free, regardless of PINSUSPENDN value. |
| 0 | SOFTRESET | R/W | 0 | Software reset<br>Read 0 = Reset done, no pending action<br>Write 0 = No action<br>Read 1 = Reset ongoing<br>Write 1 = Initiate software reset |


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.8 IRQ_EOI Register

Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if a new interrupt event is pending, when using the pulsed output.

Unused when using the level interrupt line (depending on module integration).

**Figure 23-10. IRQ_EOI Register**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | LINE_NUMBER |
| R-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 23-9. IRQ_EOI Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0 | Reserved |
| 0 | LINE_NUMBER | R/W | 0 | Write the number of the interrupt line to apply a SW EOI to it.<br>Note that there is only a single line (that is, number 0)<br>Read 0 = Always returns 0<br>Write 0 = SW EOI on interrupt line<br>Write 1 = No action |


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.9 IRQSTATUS_RAW Register

Component interrupt request status.

Check the corresponding secondary status register. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.

**Figure 23-11. IRQSTATUS_RAW Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | TCAR_IT_FLAG | OVF_IT_FLAG | MAT_IT_FLAG |
| R-0 | | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 23-10. IRQSTATUS_RAW Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0 | Reserved |
| 2 | TCAR_IT_FLAG | R/W | 0 | IRQ status for Capture<br>Read 0 = No event pending<br>Write 0 = No action<br>Read 1 = IRQ event pending<br>Write 1 = Trigger IRQ event by software |
| 1 | OVF_IT_FLAG | R/W | 0 | IRQ status for Overflow<br>Read 0 = No event pending<br>Write 0 = No action<br>Read 1 = IRQ event pending<br>Write 1 = Trigger IRQ event by software |
| 0 | MAT_IT_FLAG | R/W | 0 | IRQ status for Match<br>Read 0 = No event pending<br>Write 0 = No action<br>Read 1 = IRQ event pending<br>Write 1 = Trigger IRQ event by software |

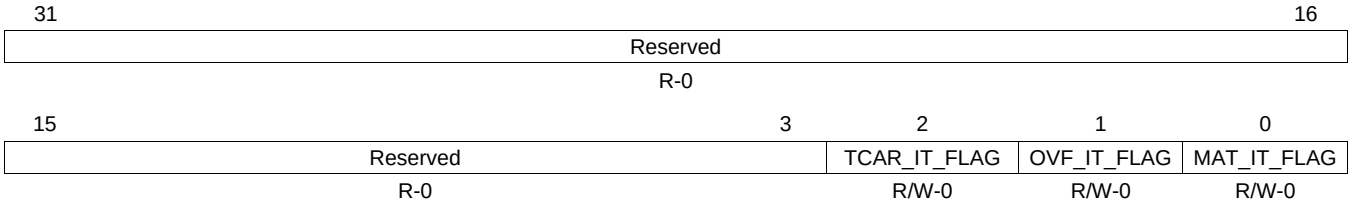

Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.10 IRQSTATUS Register

Component interrupt request status.

Check the corresponding secondary status register. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).

**Figure 23-12. IRQSTATUS Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 23-11. IRQSTATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0 | Reserved |
| 2 | TCAR_IT_FLAG | R/W | 0 | IRQ status for Capture<br>Read 0 = No event pending<br>Write 0 = No action<br>Read 1 = IRQ event pending<br>Write 1 = Clear pending event, if any |
| 1 | OVF_IT_FLAG | R/W | 0 | IRQ status for Overflow<br>Read 0 = No event pending<br>Write 0 = No action<br>Read 1 = IRQ event pending<br>Write 1 = Clear pending event, if any |
| 0 | MAT_IT_FLAG | R/W | 0 | IRQ status for Match<br>Read 0 = No event pending<br>Write 0 = No action<br>Read 1 = IRQ event pending<br>Write 1 = Clear pending event, if any |


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.11 IRQENABLE_SET Register

Component interrupt request enable.

Write 1 to set (enable interrupt).

Readout equal to corresponding _CLR register.

**Figure 23-13. IRQENABLE_SET Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | TCAR_EN_FLAG | OVF_EN_FLAG | MAT_EN_FLAG |
| R-0 | | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 23-12. IRQENABLE_SET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0 | Reserved |
| 2 | TCAR_EN_FLAG | R/W | 0 | IRQ enable for Compare<br>Read 0 = IRQ event is disabled<br>Write 0 = No action<br>Read 1 = IRQ event is enabled<br>Write 1 = Set IRQ enable |
| 1 | OVF_EN_FLAG | R/W | 0 | IRQ enable for Overflow<br>Read 0 = IRQ event is disabled<br>Write 0 = No action<br>Read 1 = IRQ event is enabled<br>Write 1 = Set IRQ enable |
| 0 | MAT_EN_FLAG | R/W | 0 | IRQ enable for Match<br>Read 0 = IRQ event is disabled<br>Write 0 = No action<br>Read 1 = IRQ event is enabled<br>Write 1 = Set IRQ enable |


Copyright © 2011–2016, Texas Instruments Incorporated

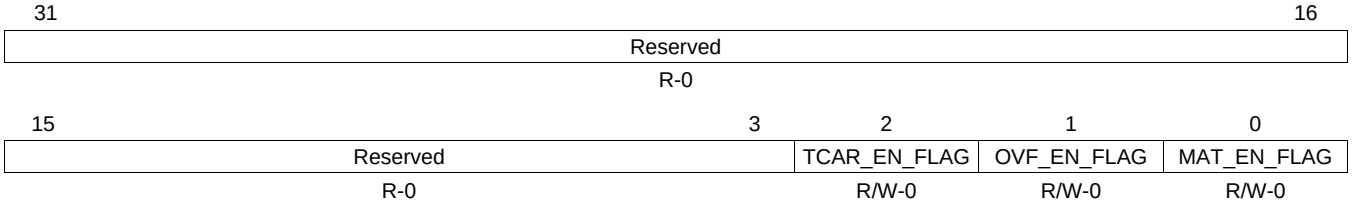
### 23.2.12 IRQENABLE_CLR Register

Component interrupt request enable.

Write 1 to clear (disable interrupt).

Readout equal to corresponding _SET register.

**Figure 23-14. IRQENABLE_CLR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 23-13. IRQENABLE_CLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0 | Reserved |
| 2 | TCAR_EN_FLAG | R/W | 0 | IRQ enable for Compare<br>Read 0 = IRQ event is disabled<br>Write 0 = No action<br>Read 1 = IRQ event is enabled<br>Write 1 = Clear IRQ enable |
| 1 | OVF_EN_FLAG | R/W | 0 | IRQ enable for Overflow<br>Read 0 = IRQ event is disabled<br>Write 0 = No action<br>Read 1 = IRQ event is enabled<br>Write 1 = Clear IRQ enable |
| 0 | MAT_EN_FLAG | R/W | 0 | IRQ enable for Match<br>Read 0 = IRQ event is disabled<br>Write 0 = No action<br>Read 1 = IRQ event is enabled<br>Write 1 = Clear IRQ enable |


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.13 IRQWAKEEN Register

Wakeup-enabled events taking place when module is idle will generate an asynchronous wakeup.

**Figure 23-15. IRQWAKEEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | TCAR_WUP_FLAG | OVF_WUP_FLAG | MAT_WUP_FLAG |
| R-0 | | | | | | | | | | | | | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 23-14. IRQWAKEEN Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0 | Reserved |
| 2 | TCAR_WUP_ENA | R/W | 0 | Wakeup generation for Compare<br>0 = Wakeup disabled<br>1 = Wakeup enabled |
| 1 | OVF_WUP_ENA | R/W | 0 | Wakeup generation for Overflow<br>0 = Wakeup disabled<br>1 = Wakeup enabled |
| 0 | MAT_WUP_ENA | R/W | 0 | Wakeup generation for Match<br>0 = Wakeup disabled<br>1 = Wakeup enabled |

### 23.2.14 TCLR Register

When the TCM field passed from (00) to any other combination then the TCAR_IT_FLAG and the edge detection logic are cleared.

The ST bit of TCLR register may be updated from the OCP interface or reset due to one-shot overflow. The OCP interface update has the priority.

**Figure 23-16. TCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0 | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | GPO_CFG | CAPT_MODE | PT | TRG | | TCM | | SCPWM | CE | PRE | PTV | | | AR | ST |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | | R/W-0 | | R/W-0 | R/W-0 | R/W-0 | R/W-0 | | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 23-15. TCLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | Reserved | R | 0 | Reserved |
| 14 | GPO_CFG | R/W | 0 | General purpose output—this register directly drives the PORGPOCFG output pin<br>0 = PORGPOCFG drives 0<br>1 = PORGPOCFG drives 1<br>Note: For specific use of GPO_CFG bit, see Section 23.1.4. |
| 13 | CAPT_MODE | R/W | 0 | Capture mode.<br>0 = Single capture<br>1 = Capture on second event |
| 12 | PT | R/W | 0 | Pulse or toggle mode on PORTIMERPWM output pin<br>0 = Pulse<br>1 = Toggle |
| 11-10 | TRG | R/W | 0 | Trigger output mode on PORTIMERPWM output pin<br>0 = No trigger<br>1h = Trigger on overflow<br>2h = Trigger on overflow and match<br>3h = Reserved |
| 9-8 | TCM | R/W | 0 | Transition Capture Mode on PIEVENTCAPT input pin<br>0 = No capture<br>1h = Capture on low to high transition<br>2h = Capture on high to low transition<br>3h = Capture on both edge transition |
| 7 | SCPWM | R/W | 0 | This bit should be set or cleared while the timer is stopped or the trigger is off<br>0 = Clear the PORTIMERPWM output pin and select positive pulse for pulse mode<br>1 = Set the PORTIMERPWM output pin and select negative pulse for pulse mode |
| 6 | CE | R/W | 0 | 0 = Compare mode is disabled<br>1 = Compare mode is enabled |
| 5 | PRE | R/W | 0 | Prescaler enable<br>0 = The TIMER clock input pin clocks the counter<br>1 = The divided input pin clocks the counter |
| 4-2 | PTV | R/W | 0 | Pre-scale clock Timer value. See Table 23-3 for more details. |
| 1 | AR | R/W | 0 | 0 = One shot timer<br>1 = Auto-reload timer |
| 0 | ST | R/W | 0 | In the case of one-shot mode selected (AR = 0), this bit is automatically reset by internal logic when the counter is overflowed.<br>0 = Stop timer: Only the counter is frozen<br>1 = Start timer |


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.15 TCRR Register

The TCRR register is a 32-bit register, 16-bit addressable. The MCU can perform a 32-bit access or two 16-bit accesses to access the register while the DSP performs 2 consecutive 16-bit transactions. Note that since the OCP clock is completely asynchronous with the timer clock, some synchronization is done in order to make sure that the TCRR value is not read while it is being incremented.

In 16-bit mode the following sequence must be followed to read the TCRR register properly:

First, perform an OCP Read Transaction to Read the lower 16-bit of the TCRR register (offset = 3Ch). When the TCRR is read and synchronized, the lower 16-bit LSBs are driven onto the output OCP data bus and the upper 16-bit MSBs of the TCRR register are stored in a temporary register.

Second, perform an OCP Read Transaction to read the upper 16-bit of the TCRR register (offset = 3Eh). During this Read, the value of the upper 16-bit MSBs that has been temporary register is forwarded onto the output OCP data bus.

So, to read the value of TCRR correctly, the first OCP read access has to be to the lower 16-bit (offset = 3Ch), followed by OCP read access to the upper 16-bit (offset = 3Eh).

As the TCRR is updated using more sources (shadow_in_tcrr, incremented value of tcrr, TLDR and “0”), a priority order will be defined:

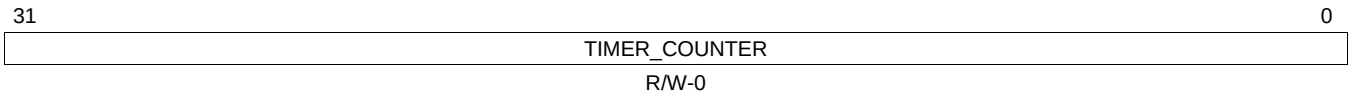
The first priority is the OCP update.

The second is the reload way (triggered through TTGR reg. or following an auto-reload overflow).

The third is the one-shot overflow reset to 0.

The last is the incremented value.

**Figure 23-17. TCRR Register**




LEGEND: R = Read only; -*n* = value after reset

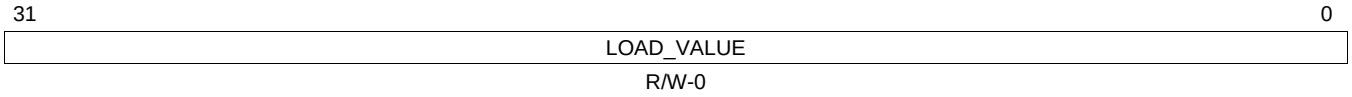
**Table 23-16. TCRR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | TIMER_COUNTER | R/W | 0 | Value of TIMER counter |

### 23.2.16 TLDR Register

LOAD_VALUE must be different than the timer overflow value (FFFF FFFFh).

**Figure 23-18. TLDR Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 23-17. TLDR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | LOAD_VALUE | R/W | 0 | Timer counter value loaded on overflow in auto-reload mode or on TTGR write access |

### 23.2.17 TTGR Register

The read value of this register is always FFFF FFFFh.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 23-19. TTGR Register**

| 31 | 0 |
|---|---|
| TTGR_VALUE | |
| R/W-FFFF FFFFh | |

LEGEND: R = Read only; -*n* = value after reset

**Table 23-18. TTGR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | TTGR_VALUE | R/W | FFFF FFFFh | Any write to the TTGR register will load the value from TLDR into TCRR and clear the prescaler counter. This will occur regardless of the AR field value in the TCLR register. A read access of TTGR will always return FFFF FFFFh. |

### *23.2.18 TWPS Register*

In posted mode the software must read the pending write status bits (Timer Write Posted Status register bits [4:0]) to insure that following write access will not be discarded due to on going write synchronization process. These bits are automatically cleared by internal logic when the write to the corresponding register is acknowledged.

**Figure 23-20. TWPS Register**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0 | |

| 15 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | W_PEND_TMAR | W_PEND_TTGR | W_PEND_TLDR | W_PEND_TCRR | W_PEND_TCLR |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R = Read only; -*n* = value after reset

**Table 23-19. TWPS Register Field Descriptions**

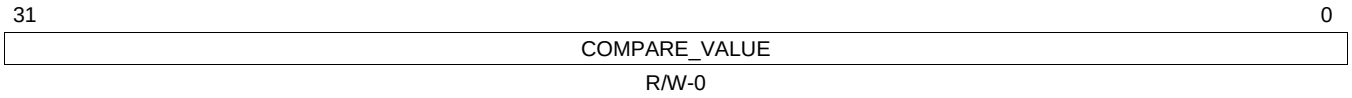
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | Reserved | R | 0 | Reserved |
| 4 | W_PEND_TMAR | R | 0 | When equal to 1, a write is pending to the TMAR register |
| 3 | W_PEND_TTGR | R | 0 | When equal to 1, a write is pending to the TTGR register |
| 2 | W_PEND_TLDR | R | 0 | When equal to 1, a write is pending to the TLDR register |
| 1 | W_PEND_TCRR | R | 0 | When equal to 1, a write is pending to the TCRR register |
| 0 | W_PEND_TCLR | R | 0 | When equal to 1, a write is pending to the TCLR register |


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.19 TMAR Register

The TMAR register stores the value to be compared against the counter's current value using the timer's compare logic.

**Figure 23-21. TMAR Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 23-20. TMAR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | COMPARE_VALUE | R/W | 0 | Value to be compared to the timer counter |

### 23.2.20 TCAR1 Register

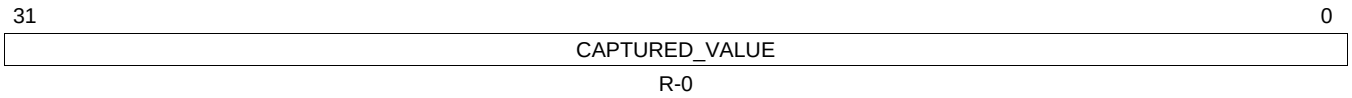
When the appropriate (rising, falling or both) transition is detected in the edge detection logic the current counter value is stored to the TCAR1 register. Note that since the OCP clock is completely asynchronous with the timer clock, some synchronization is done in order to make sure that the TCAR1 value is not read while it is being updated due to some capture event.

In 16-bit mode the following sequence must be followed to read the TCAR1 register properly:

First, perform an OCP Read Transaction to Read the lower 16-bits of the TCAR1 register.

Second, perform an OCP Read Transaction to read the upper 16-bits of the TCAR1 register.

**Figure 23-22. TCAR1 Register**




LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 23-21. TCAR1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CAPTURED_VALUE | R | 0 | Timer counter value captured on an external event trigger |


Copyright © 2011–2016, Texas Instruments Incorporated

### 23.2.21 TSICR Register

Access to this register is not stalled even if the timer is in non-posted mode configuration. To abort any wrong behavior, software can permanently reset the functional part of the module. Also in case of a wrong hardware PIFREQRATIO tied the POSTED field can be reprogrammed on the fly, so deadlock situation cannot happen.

Reset value of POSTED depends on hardware integration module at design time. Software must read POSTED field to get the hardware module configuration.

**Figure 23-23. TSICR Register**

| 31 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | POSTED | SFT | Reserved |
| R-0 | | R/W-0 | R/W-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 23-22. TSICR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | Reserved | R | 0 | Reserved |
| 2 | POSTED | R/W | 0 | PIFREQRATIO<br>0 = Posted mode inactive: will delay the command accept output signal.<br>1 = Posted mode active (clocks ratio needs to fit freq (timer) less than freq (OCP)/4 frequency requirement) |
| 1 | SFT | R/W | 0 | This bit resets all the function parts of the module. During reads it always returns 0.<br>0 = Software reset is disabled<br>1 = Software reset is enabled |
| 0 | Reserved | R | 0 | Reserved |

### 23.2.22 TCAR2 Register

When the appropriate (rising, falling or both) transition is detected in the edge detection logic and the capture on second event is activated from the control register (TCLR), the current counter value is stored to the TCAR2 register. Note that since the OCP clock is completely asynchronous with the timer clock, some synchronization is done in order to make sure that the TCAR2 value is not read while it is being updated due to some capture event.

In 16-bit mode the following sequence must be followed to read the TCAR2 register properly:

First, perform an OCP Read Transaction to Read the lower 16-bits of the TCAR2 register.

Second, perform an OCP Read Transaction to read the upper 16-bits of the TCAR2 register.

**Figure 23-24. TCAR2 Register**

| 31 | 0 |
|---|---|
| CAPTURED_VALUE | |
| R-0 | |

LEGEND: R/W = Read/Write; -*n* = value after reset

**Table 23-23. TCAR2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CAPTURED_VALUE | R | 0 | Timer counter value captured on an external event trigger |


Copyright © 2011–2016, Texas Instruments Incorporated

# UART/IrDA/CIR Module

This chapter describes the function, operation, and configuration of the universal asynchronous receiver/transmitter (UART)/infrared data association (IrDA)/consumer infrared (CIR) module.

**Topic** **Page**

**24.1 Introduction** ..... **3580**
**24.2 Architecture** ..... **3582**
**24.3 UART Registers** ..... **3614**


Copyright © 2011–2016, Texas Instruments Incorporated

## 24.1 Introduction

### 24.1.1 Main Features

The main features of each of the UARTs are:

- Selectable UART/IrDA/CIR modes
- 16C750 compatibility
- Dual 64 entry FIFOs for received and transmitted data payload
- Programmable and selectable transmit and receive FIFO trigger levels for DMA and interrupt generation
- Programmable sleep mode
- Complete status reporting capabilities in both normal and sleep mode
- Frequency prescaler values from 0 to 16383 to generate the appropriate baud rates
- Internal system clock reference for baud setting (In this document, a 48 MHz reference clock is considered. For more information on reference clock support, see your device-specific data manual)
- Two DMA requests, 1 interrupt request to the system
- Break character detection and generation
- Configurable data format
  - Data bit: 5, 6, 7, or 8 bits
  - Parity bit: Even, odd, none
  - Stop-bit: 1, 1.5, 2 bit(s)
- Flow control: Hardware (RTS/CTS) or software (XON/XOFF)

The UART/IrDA module can operate in six different modes:

- UART 16× mode
- UART 16× mode with autobauding (greater than or equal to 1200 bits/s and less than or equal to 115.2 Kbits/s)
- UART 13× mode
- IrDA SIR mode (less than or equal to 115.2 Kbits/s)
- IrDA MIR mode (0.576 and 1.152 Mbits/s)
- IrDA FIR mode (4 Mbits/s)
- CIR mode (programmable modulation rates specific to remote control applications)


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.1.2 UART/Modem Functions

The UART/Modem has the following functions:

- Baud-rate from 300 bits/s up to 3.6864 Mbits/s (with a 48 MHz reference clock)
- Auto-baud between 1200 bits/s and 115.2 Kbits/s
- Software/Hardware flow control
  - Programmable Xon/Xoff characters
  - Programmable auto-RTS and auto-CTS
- Programmable serial interface characteristics
  - 5, 6, 7, or 8 bit characters
  - Even, odd, mark (always = 1), space (always = 0) or no parity (non parity bit frame) bit generation and detection
  - 1, 1.5, or 2 stop bit generation
- False start bit detection
- Line break generation and detection
- Fully prioritized interrupt system controls
- Internal test and loopback capabilities
- Modem control functions ($\overline{\text{CTS}}$, $\overline{\text{RTS}}$, $\overline{\text{DSR}}$, $\overline{\text{DTR}}$, RIN, and $\overline{\text{DCD}}$)

### 24.1.3 IrDA Functions

The IrDA has the following functions:

- Slow infrared (SIR 115.2 KBAUD), medium infrared (MIR 0.576 MBAUD) and fast infrared (FIR 4.0 MBAUD) operations (very fast infrared (VFIR) is not supported)
- Framing error, cyclic redundancy check (CRC) error, illegal symbol (FIR), abort pattern (SIR, MIR) detection
- 8-entry status FIFO (with selectable trigger levels) available to monitor frame length and frame errors

### 24.1.4 CIR Features

The CIR mode uses a variable pulse-width modulation (PWM) technique (based on multiples of a programmable *t* period) to encompass the various formats of infrared encoding for remote-control applications. The CIR logic transmits data packets based on a user-definable frame structure and packet content.

The CIR includes the following key features to provide CIR support for remote control applications:

- Consumer Infrared remote control mode with programmable data encoding
- Transmit and receive modes supported
- Free data format (supports any remote-control private standards)
- Selectable bit rate
- Configurable carrier frequency
- 1/2, 5/12, 1/3, or 1/4 carrier duty cycle


Copyright © 2011–2016, Texas Instruments Incorporated

## 24.2 Architecture

### 24.2.1 UART Signal Descriptions

Table 24-1 describes the UART interface.

**Table 24-1. UART Signal Description**

| Signal | I/O | Description | Reset |
|---|---|---|---|
| RX | I | Serial data input for all modes. | Unknown |
| TX | O | Serial data output in UART modes. In other modes, this pin is set to the reset value (inactive state). | 1 |
| IRTX | O | Serial data output in IrDA modes (SIR, MIR and FIR). In other modes, this pin is set to the reset value (inactive state). | 0 |
| RCTX | O | Serial data output in CIR mode. In other modes, this pin is set to the reset value (inactive state). | 0 |
| $\overline{CTS}$ | I | Clear to send. Active-low modem status signal. Reading bit 4 of the modem status register checks the condition of $\overline{CTS}$. Reading bit 0 of that register checks a change of state of $\overline{CTS}$ since the last read of the modem status register. $\overline{CTS}$ is used in auto-CTS mode to control the transmitter. | Unknown |
| $\overline{RTS}$ | O | Request to send. When active (low), the module is ready to receive data. Setting modem control register bit 1 activates $\overline{RTS}$. It becomes inactive as a result of a module reset, loop back mode or by clearing the MCR[1]. In auto-RTS mode, it becomes inactive as a result of the receiver threshold logic. | 1 |
| SD | O | SD mode is used to configure the transceivers. The SD pin out is an inverted value of ACREG[6]. | 1 |
| $\overline{DSR}$ | I | Data set ready: Active-low modem status signal. Reading bit 5 of the modem status register checks the condition of $\overline{DSR}$. Reading bit 1 of that register checks a change of state of $\overline{DSR}$ since the last read of the modem status register. | Unknown |
| $\overline{DTR}$ | O | Data terminal ready: When active (low), this signal informs the modem that the module is ready to communicate. It is activated by setting bit 0 of the modem control register. | 1 |
| $\overline{DCD}$ | I | Data carrier detect: Active-low modem status signal. The condition of $\overline{DCD}$ can be checked by reading bit 7 of the modem status register and any change in its state can be detected by reading bit 3 of that register. | Unknown |
| RIN | I | Ring indicator: Active-low modem status signal. The condition of RIN can be checked by reading bit 6 of the modem status register and any change in its state can be detected by reading bit 2 of that register. | Unknown |

NOTE: Not all UART instances on a specific device support the full pinout shown here. Refer to the device data manual for details

### 24.2.2 UART/IrDA/CIR Mode Selection

To select a mode, set the MODESELECT field in MDR1[2:0] (see Table 24-2).

**Table 24-2. UART Mode Selection**

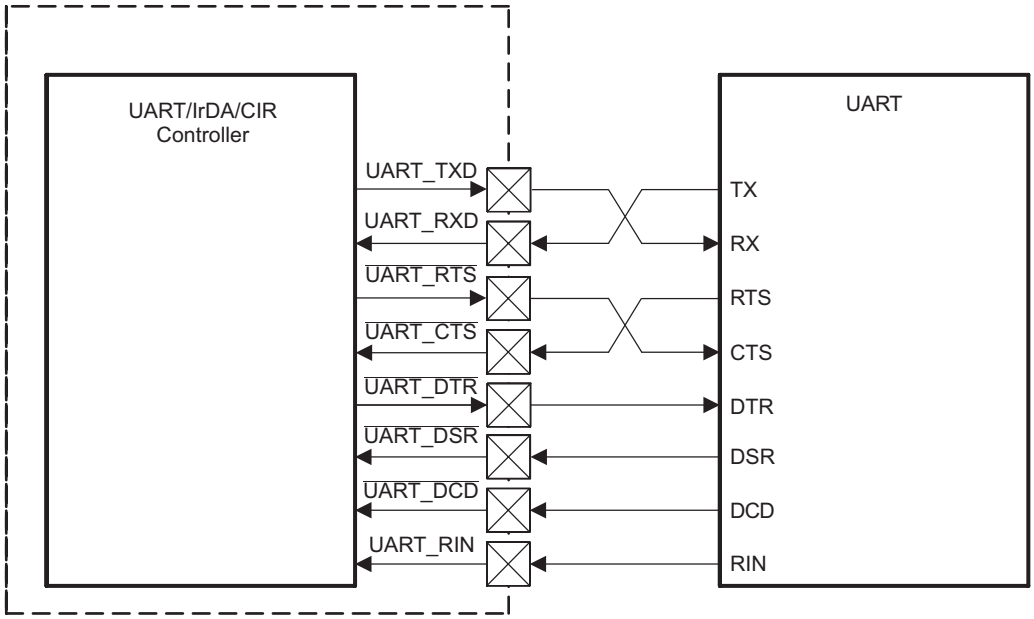
| MDR1[2:0] Value | Mode |
|---|---|
| 0h | UART 16x mode |
| 1h | SIR mode |
| 2h | UART 16x auto-baud |
| 3h | UART 13x mode |
| 4h | MIR mode |
| 5h | FIR mode |
| 6h | CIR mode |
| 7h | Disable (default state) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.2.3 UART Mode

This section describes how the UART functions in UART wired mode. Figure 24-1 shows how to connect the UART operating in UART mode to another UART on an external device.

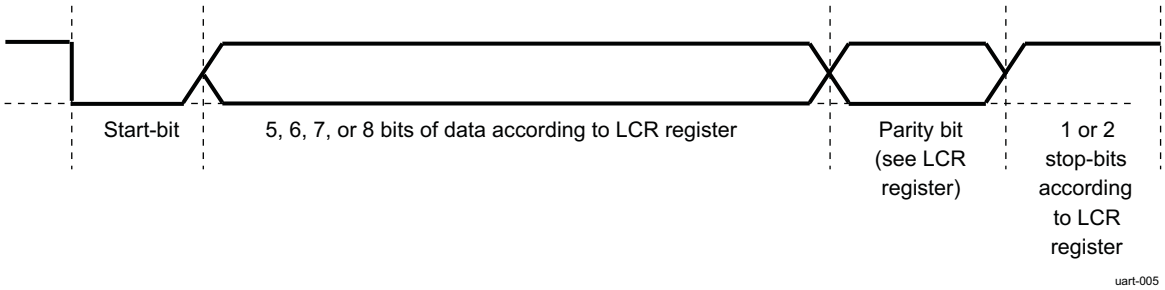
**Figure 24-1. UART to UART Connection With Full Handshaking**




The UART uses a wired interface for serial communication with a remote device. It can use hardware or software flow control to manage transmission/reception. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the $\overline{\text{RTS}}$ output and $\overline{\text{CTS}}$ input signals. Software flow control automatically controls data flow by using programmable XON/XOFF characters.

The UART modem module is enhanced with an autobauding functionality, which in control mode allows to automatically set the speed, the number of bit per character, and the parity selected. Figure 24-2 shows the UART frame data format. Each frame consists of a start bit, 5 to 8 data bits (user configurable), an optional parity bit (user configurable), and 1 or 2 stop bits (user configurable).

**Figure 24-2. UART Frame Data Format**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 24.2.3.1 Hardware Flow Control

Hardware flow control is composed of auto-CTS and auto-RTS. Auto-CTS and auto-RTS can be enabled/disabled independently by programming EFR[7:6]. With auto-CTS, $\overline{CTS}$ must be active before the module can transmit data.

Auto-RTS only activates the $\overline{RTS}$ output when there is enough room in the RX FIFO to receive data and de-activates the $\overline{RTS}$ output when the RX FIFO is sufficiently full. The HALT and RESTORE trigger levels in the TCR determine the levels at which $\overline{RTS}$ is activated/de-activated.

If both auto-CTS and auto-RTS are enabled, data transmission does not occur unless the receiver FIFO has empty space. Thus, overrun errors are eliminated during hardware flow control. If not enabled, overrun errors occur if the transmit data rate exceeds the receive FIFO latency.

##### *24.2.3.1.1 Auto-RTS*

Auto-RTS data flow control originates in the receiver block (see functional block diagram). The receiver FIFO trigger levels used in auto-RTS are stored in the TCR. $\overline{RTS}$ is active if the RX FIFO level is below the HALT trigger level in TCR[3:0]. When the receiver FIFO HALT trigger level is reached, $\overline{RTS}$ is de-asserted. The sending device (for example, another UART) may send an additional byte after the trigger level is reached because it may not recognize the de-assertion of $\overline{RTS}$ until it has begun sending the additional byte. $\overline{RTS}$ is automatically reasserted once the receiver FIFO reaches the RESUME trigger level programmed via TCR[7:4]. This reassertion requests the sending device to resume transmission.

##### *24.2.3.1.2 Auto-CTS*

The transmitter circuitry checks $\overline{CTS}$ before sending the next data byte. When $\overline{CTS}$ is active, the transmitter sends the next byte. To stop the transmitter from sending the following byte, $\overline{CTS}$ must be de-asserted before the middle of the last stop bit that is currently being sent. The auto-CTS function reduces interrupts to the host system. When auto-CTS flow control is enabled, the $\overline{CTS}$ state changes need not trigger host interrupts because the device automatically controls its own transmitter. Without auto-CTS, the transmitter sends any data present in the transmit FIFO and a receiver overrun error can result.

#### 24.2.3.2 Software Flow Control

Software flow control is enabled through the enhanced feature register (EFR) and the modem control register (MCR). Different combinations of software flow control can be enabled by setting different combinations of EFR[3:0]. There are two other enhanced features relating to software flow control:

- XON any function (MCR[5]): operation will resume after receiving any character after recognizing the XOFF character. The XON-any character is written into the RX FIFO even if it is a software flow character.
- Special character (EFR[5]): Incoming data is compared to XOFF2. Detection of the special character sets the XOFF interrupt (IIR[4]) but does not halt transmission. The XOFF interrupt is cleared by a read of the IIR. The special character is transferred to the RX FIFO.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 24.2.3.2.1 Receive (RX)

When software flow control operation is enabled, the UART compares incoming data with XOFF1/2 programmed characters (in certain cases XOFF1 and XOFF2 must be received sequentially). When the correct XOFF characters are received, transmission is halted after completing transmission of the current character. XOFF detection also sets IIR[4] (if enabled via IER[5]) and causes the interrupt line to go low. To resume transmission a XON1/2 character must be received (in certain cases XON1 and XON2 must be received sequentially). When the correct XON characters are received IIR[4] is cleared and the XOFF interrupt disappears.

**NOTE:** If a parity, framing or break error occurs while receiving a software flow control character, this character will be treated as normal data and will be written to the RX FIFO.

When XON-any and special character detect are disabled and software flow control is enabled no valid XON or XOFF characters are written to the RX FIFO. For example, EFR[1:0] = 10, if XON1 and XOFF1 characters are received they do not get written to the RX FIFO. In the case where pairs of software flow characters are programmed to be received sequentially (EFR[1:0] = 11) the software flow characters are not written to the RX FIFO if they are received sequentially. However, received XON1/XOFF1 characters must be written to the RX FIFO if the subsequent character is not XON2/XOFF2.

#### 24.2.3.2.2 Transmit (TX)

- XOFF1: Two characters are transmitted when the RX FIFO has passed the programmed trigger level TCR[3:0].
- XON1: Two characters are transmitted when the RX FIFO reaches the trigger level programmed via TCR[7:4].

**NOTE:** If after an XOFF character has been sent, software flow control is disabled, the module transmits XON characters automatically to enable normal transmission to proceed.

The transmission of XOFF/XON (s) follows the exact same protocol as transmission of an ordinary byte from the TX FIFO. This means that even if the word length is set to be 5, 6 or 7 characters then the 5, 6 or 7 least significant bits of XOFF1,2/XON1,2 are transmitted. Note that the transmission of 5, 6, or 7 bits of a character is seldom done but this functionality is included to maintain compatibility with earlier designs. It is assumed that software flow control and hardware flow control will never be enabled simultaneously.

### 24.2.3.3 Break and Time-Out Conditions

#### 24.2.3.3.1 Time-Out Counter

An RX idle condition is detected when the receiver line, RX, has been high for a time equivalent to 4X programmed word length+12 bits. The receiver line is sampled midway through each bit. For sleep mode the counter is reset when there is activity on the RX line. For the timeout interrupt, the counter only counts when there is data in the RX FIFO and the count is reset when there is activity on the RX line or when the RHR is read.

#### 24.2.3.3.2 Break Condition

When a break condition occurs, the TX line is pulled low. A break condition is activated by setting LCR[6]. Be aware that the break condition is not aligned on word stream; that is, a break condition can occur in the middle of a character. The only way to send a break condition on a full character, is:

- Reset transmit FIFO (if enabled)
- Wait for transmit shift register becomes empty (LSR[6] = 1)
- Take a guard time according to stop bit definition
- Set LCR[6] to 1


Copyright © 2011–2016, Texas Instruments Incorporated

A break condition is asserted as long as LCR[6] is set to 1. The above functionality (timeout counter and break condition) only applies to the UART Modem operation and does not extend to the IrDA/CIR modes of operation.

**NOTE:** A break condition is asserted irrespective of CTS state when auto-CTS is programmed.

#### 24.2.3.4 UART Configuration Example

This section proposes outlines the programming stages to operate one UART module with FIFO, interrupt and no DMA capabilities. This is a three-step procedure that ensures quick start of these modules (obviously it does not cover every UART module feature). The first stage covers SW reset of the module (interrupts, status and controls); the second stage deals with FIFO configuration and enable. Finally, last stage deals with the baud rate data and stop configuration. The procedure below is programming language agnostic.

##### *24.2.3.4.1 UART Software Reset*

Goal: To clear IER and MCR registers: remove UART breaks (LCR[6] = 0) and put module in reset (MDR1[2:0] = 7h).

Procedure: To write into both the IER and MCR register, EFR[4] must first be set to 1. To be able to access the EFR register, BFh must first be written to LCR register. Hence LCR = BFh; first write to the LCR register EFR[4] = 1. When LCR = BFh, enable the enhanced feature register LCR[7] = 0. Here, access to IER and MCR is allowed. IER = 0; disable interrupt MCR = 0. Force control signals inactive LCR[6] = 0. Here, UART breaks are removed MDR1 = 7h. Here, UART is in reset or disabled. Alternatively, the SYSC[1] can be set to 1 to start a hardware reset from the generic synchronous reset module. The reset progress can be monitored via the SYSS[0]. Once complete the above sequence should ensure the UART is in the equivalent disabled mode with reference to MDR1[2:0].

##### *24.2.3.4.2 UART FIFO Configuration*

Goal: To set trigger level for halt/restore (TCR register), set trigger level for transmit/receive (TLR register), and configure FIFO (FCR register).

Procedure: To write into both the TLR and TCR registers, EFR[4] must be set to 1 and MCR[6] to 1. To write into FCR, EFR[4] must be set to 1. Note that EFR[4] = 1 has already been done in the previous section, hence a write to MCR[6] is necessary. MCR[6] = 1; set TCR, TLR, and FCR to the desired value. Here, accesses to TCR, TLR, and FCR must be disabled to avoid any further undesired writes to these registers. Hence, LCR = BFh, provides access to EFR, EFR[4] = 0; LCR[7] = 0; MCR[6] = 0.

##### *24.2.3.4.3 Baud Rate Data and Stop Configuration*

Goal: To configure UART data and stop (LCR register). baud rate (DLH and DLL registers), and enable UART operation. In case interrupt capability is added, configuration must be added right before UART enable.

Procedure: Set LCR to desired value; LCR[7] to 1, gives access to DLH and DLL registers. Set DLH and DLL; LCR[7] = 0, removes access to DLH and DLL registers. Set IER to desired value. Set interrupts MDR1[2:0] = 0; enables UART without autobauding.

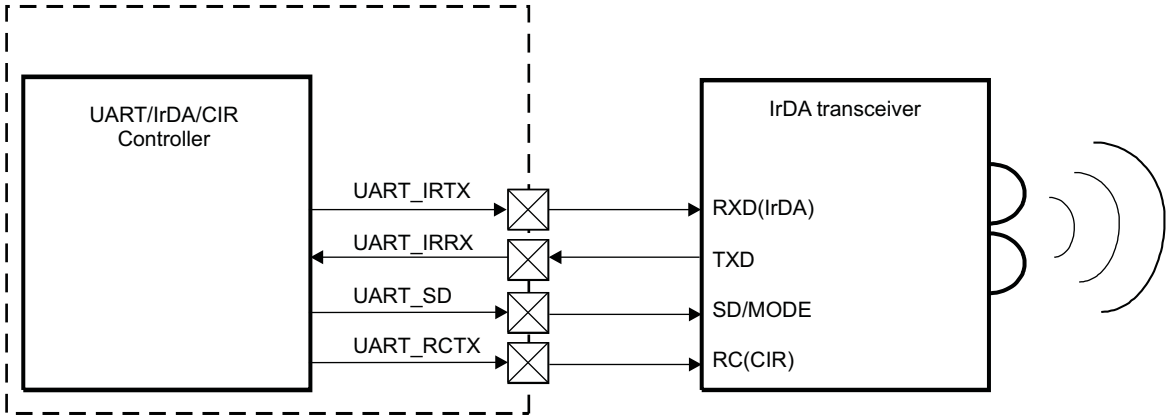
The UART module is operational.


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.2.4 IrDA Mode

This section describes the IrDA connection with an external device. Figure 24-3 shows how UART3 can be connected to an external infrared transceiver in the IrDA mode.

**Figure 24-3. UART IrDA to External IR Device**




The module performs serial-to-parallel conversion on received data characters and parallel-to-serial conversion on transmitted data characters by the processor. The complete status of each channel of the module and each received character/frame can be read at any time during functional operation via the line status register (LSR).

The module can be placed in an alternate mode (FIFO mode) relieving the processor of excessive software overhead by buffering received/transmitted characters. Both the receiver and transmitter FIFOs can store up to 64 bytes of data (plus three additional bits of error status per byte for the receiver FIFO) and have selectable trigger levels. Both interrupts and DMA are available to control the data-flow between the LH and the module.

#### 24.2.4.1 SIR Mode

In slow infrared (SIR) mode, data transfer takes place between the LH and peripheral devices at speeds of up to 115200 bauds speed. A SIR transmit frame starts with start flags (either a single C0h, multiple C0h, or a single C0h preceded by a number of FFh flags), followed by frame data, CRC-16 and ends with a stop flag (C1h). The bit format for a single word uses a single start bit, eight data bits and one stop bit and is unaffected by the use and settings of the LCR register.

Note that BLR[6] is used to select whether C0h or FFh start patterns is to be used, when multiple start flags are required. The SIR transmit state machine attaches start flags, CRC-16 and stop flags. It checks the outgoing data to establish if data transparency is required. SIR transparency is carried out if the outgoing data, between the start and stop flags, contains C0h, C1h, or 7Dh. If one of these is about to be transmitted, then the SIR state machine sends an escape character (7Dh) first, then inverts the fifth bit of the real data to be sent, and sends this data immediately after the 7Dh character.

The SIR receive state machine recovers the receive clock, removes the start flags, removes any transparency from the incoming data and determines frame boundary with reception of the stop flag. It also checks for errors such as: frame abort (0x7D character followed immediately by a C1h stop flag, without transparency), CRC error and frame-length error. At the end of a frame reception, the LH reads the line status register (LSR) to find out possible errors of the received frame.

**NOTE:** Data can be transferred both ways by the module, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware. See the UART3.ACREG[5] DIS_IR_RX bit description. This applies to all three modes: SIR, MIR, and FIR.


Copyright © 2011–2016, Texas Instruments Incorporated

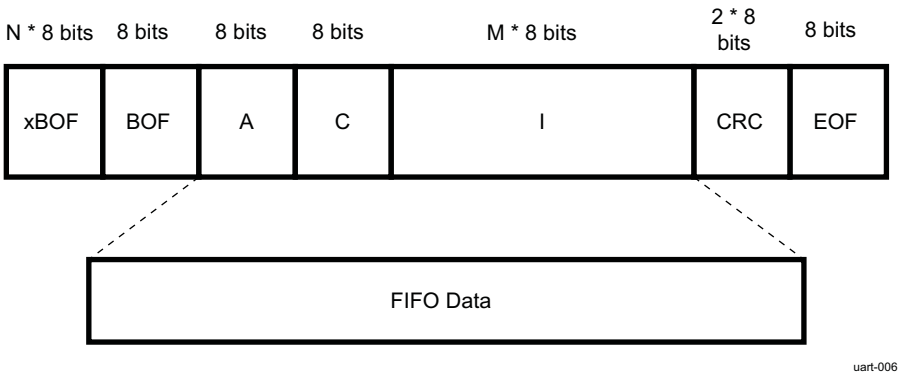
The infrared output in SIR mode can either be 1.6 µs or 3/16 encoding, selected by the PULSETYPE bit of the auxiliary control register (ACREG[7]). In 1.6 µs encoding, the infrared pulse width is 1.6 µs and in 3/16 encoding the infrared pulse width is 3/16 of a bit duration (1/baud-rate). The transmitting device must send at least two start flags at the start of each frame for back-to-back frames.

**NOTE:** Reception supports variable-length stop-bits.

#### 24.2.4.1.1 Frame Format

Figure 24-4 shows the IrDA SIR frame format.

**Figure 24-4. IrDA SIR Frame Format**




The CRC is applied on the address (A), control (C), and information (I) bytes.

**NOTE:** The two words of CRC are written to the FIFO in reception.

#### 24.2.4.1.2 Asynchronous Transparency

Before transmitting a byte, the UART IrDA controller examines each byte of the payload and the CRC field (between BOF and EOF). For each byte equal to C0h (BOF), C1h (EOF), or 7Dh (control escape), the controller performs certain tasks:

- In transmission:
  - Inserts a control escape (CE) byte preceding the byte
  - Complements bit 5 of the byte (that is, exclusive ORs the byte with 20h)

  The byte sent for the CRC computation is the initial byte written in the TX FIFO (before the XOR with 20h).
- In reception:

  For the A, C, I, CRC field:
  - Compares the byte with the CE byte; if they are not equal, sends the byte to the CRC detector and stores it in the RX FIFO
  - If the byte is equal to the CE byte, discards the CE byte
  - Complements bit 5 of the byte following the CE
  - Sends the complemented byte to the CRC detector and stores it in the RX FIFO


Copyright © 2011–2016, Texas Instruments Incorporated

#### 24.2.4.1.3 Abort Sequence

The transmitter may decide to prematurely close a frame. The transmitter aborts by sending following sequence: 7DC1h. The abort pattern closes the frame without a CRC field or an ending flag. It is possible to abort a transmission frame by programming the ABORTEN bit of the auxiliary control register (ACREG[1]). When this bit is set to 1, 7Dh and C1h are transmitted and the frame is not terminated with CRC or stop flags. The receiver treats a frame as an aborted frame when a 7Dh character followed immediately by a C1h character have been received without transparency.

#### 24.2.4.1.4 Pulse Shaping

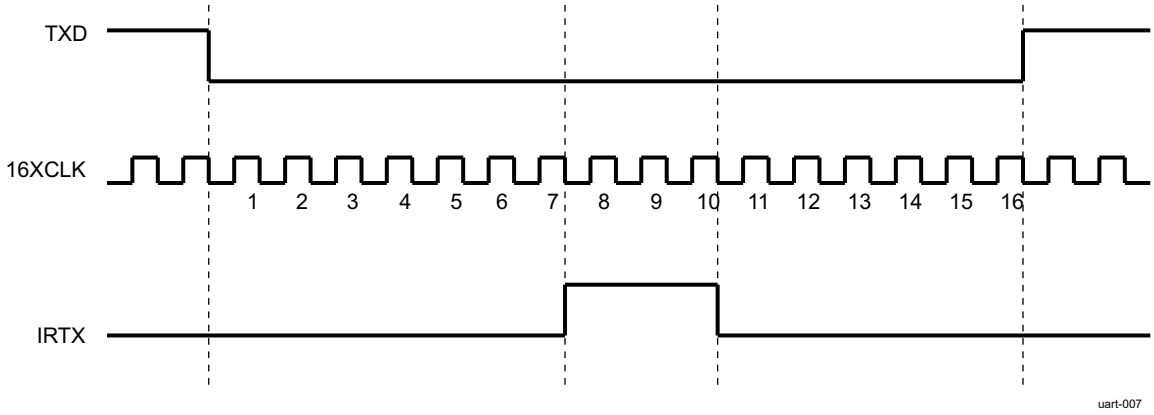
The SIR mode supports both the 3/16th and the 1.6-µs pulse duration methods. The PULSETYPE bit of the auxiliary control register (ACREG[7]) selects the pulse width method in transmit mode.

#### 24.2.4.1.5 Encoder

Serial data from transmit state-machine is encoded to transmit data to the optoelectronics. While the serial data input to the (TXD) is high, the output (IRTX) is always low, and the counter used to form a pulse on IRTX is continuously cleared. After TXD resets to 0, IRTX rises on the falling edge of the 7th 16XCLK. On the falling edge of the 10th 16XCLK pulse, IRTX falls, creating a 3-clock-wide pulse. While TXD stays low, a pulse is transmitted during the 7th to the 10th clock of each 16-clock bit cycle.

Figure 24-5 shows the IrDA SIR encoding mechanism.

**Figure 24-5. IrDA SIR Encoding Mechanism**



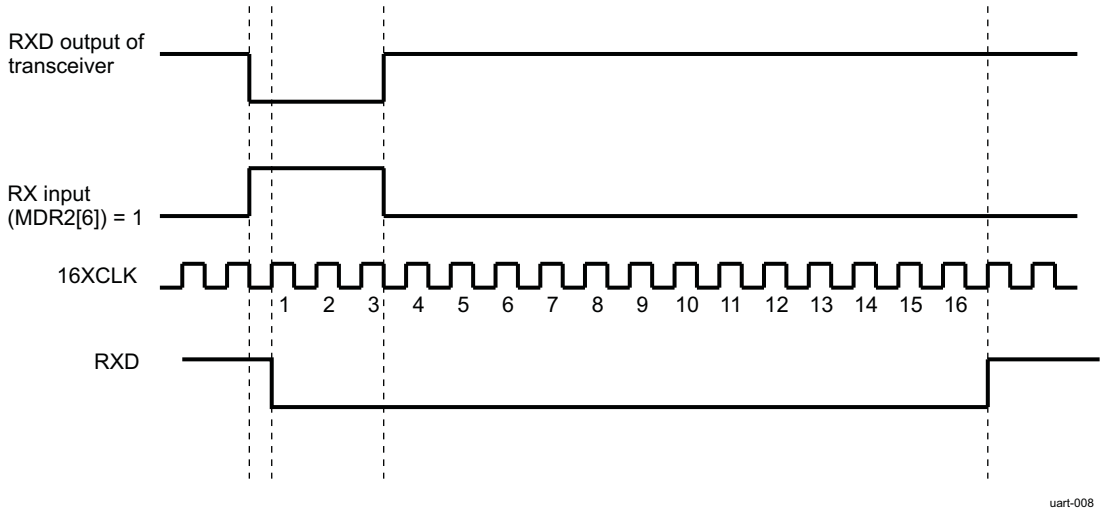

#### 24.2.4.1.6 Decoder

After reset, RXD is high and the 4-bit counter is cleared. When a rising edge is detected on RX, RXD falls on the next rising edge of 16XCLK with sufficient setup time. RXD stays low for 16 cycles (16XCLK) and then returns to high as required by the IrDA specification. As long as no pulses (rising edges) are detected on the RX, RXD remains high.

Figure 24-6 shows the IrDA SIR decoding mechanism.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 24-6. IrDA SIR Decoding Mechanism**




The operation of the RX input can be disabled with DISIRRX bit of the auxiliary control register (ACREG[5]). Furthermore, the MDR2[6] can be used to invert the signal from the transceiver (RX output) pin to the IRRX logic inside the UART.

#### *24.2.4.1.7 IR Address Checking*

In all IR modes, if address checking is enabled, only frames intended for the device are written to the RX FIFO. This is to avoid receiving frames not meant for this device in a multi-point infrared environment. It is possible to program two frame addresses that the UART IrDA receives with XON1/ADDR1 and XON2/ADDR2 registers.

Selecting address1 checking is done by setting EFR[0] to 1; selecting address2 checking is done by setting EFR[1] to 1. Clearing EFR[1:0] to 0 disables all address checking operations. If both bits are set, then the incoming frame is checked for both private and public addresses. If address checking is disabled, then all received frames are written into the reception FIFO.

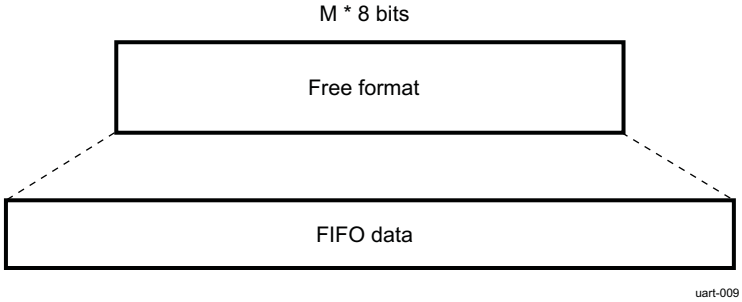
#### *24.2.4.1.8 SIR Free Format Mode*

To allow complete software flexibility in the transmission and reception of Infrared data packets, the SIR Free Format mode is a sub function of the existing SIR mode such that all frames going to and from the FIFO buffers are untouched with respect to appending and removing control characters and CRC values. In transmission phase, it uses the IRTX pin as in SIR mode.

This mode corresponds to a UART mode with a pulse modulation of 3/16 of baud-rate pulse width. For example, a normal SIR packet has BOF control and CRC error checking data appended (transmitting) or removed (receiving) from the data going to and from the FIFOs. In SIR Free Format mode only the data termed the FIFO DATA area as shown in Figure 24-7 (and in Figure 24-4) would be transmitted and received.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 24-7. SIR Free Format Mode**




In this mode, the entire FIFO data packet is to be constructed (encoded and decoded) by the LH software.

The SIR Free Format mode is selected by setting the module in UART mode (MDR1[2:0] = 000) and the MDR2[3] register bit to one to allow the pulse shaping.

As the bit format is to remain the same, some UART mode configuration register need to be set at specific value:

- LCR[1:0] = 11 (8 data bits)
- LCR[2] = 0 (1 stop bit)
- LCR[3] = 0 (no parity)
- ACREG[7] = 0 (3/16 of baud-rate pulse width)

The features defined through MDR2[6] and ACREG[5] are also supported.

**NOTE:** All other configuration registers need to be at the reset value.

The UART mode interrupts are used for the SIR Free Format mode but many of them are not relevant (XOFF, RTS, CTS, Modem status register, …….).

#### *24.2.4.1.9 Programming Models*

***Example 24-1. Receive IrDA***

Receive IrDA frame with parity forced to 1, baud-rate = 112.5Kbs, FIFOs disabled.

- MDR1 = 01h
  - MDR1[2:0] = 001b: SIR mode
- LCR = A8h
  - LCR[7] = 1: access to write DLL and DLH
  - LCR[5:3] = 101: set parity type to forced 1 (default: no parity).
  - optional: LCR[2] = 1: set number of stop bits to 2 (default: 1)
  - optional: LCR[1:0] = 11b: set word length to 8 bits (default: 5)
- DLL = 1Ah: 115.2 Kbs
- DLH = 0h: 115.2 Kbs
- LCR = 28h: LCR[7] = 0 disables access to DLL and DLH and gives access to MCR, FCR, IER, BLR, EBLR, RHR
- optional: IER = 1: enable RHR interrupt


Copyright © 2011–2016, Texas Instruments Incorporated

***Example 24-2. Transmit IrDA***

Transmit IrDA 6bytes frame with no parity, baud-rate = 112.5Kbs, FIFOs disabled, 3/16 encoding.

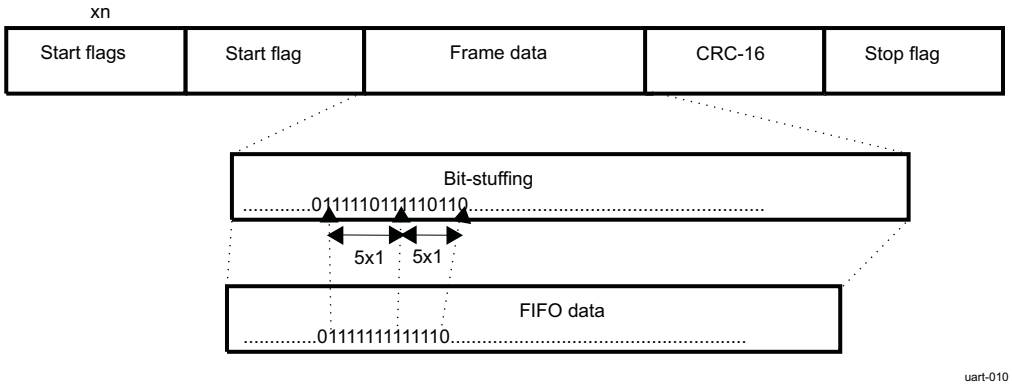
- MDR1 = 41h
  - MDR1[2:0] = 001b: SIR mode
  - MDR1[6] = 1: SIP is generated automatic after each transmission
- LCR = BFh: gives you access to EFR
- EFR = 10h: enhanced functions write enable
- LCR = 86h
  - LCR[7] = 1: access to write DLL and DLH
  - LCR[2] = 1: set number of stop bits to 2 (default: 1)
  - LCR[1:0] = 10b: set word length to 7 bits (default: 5)
- DLL = 1Ah: 115.2 Kbs
- DLH = 0h: 115.2 Kbs
- LCR = 06h: LCR[7] = 0 disables access to DLL and DLH and gives access to MCR, FCR, IER, BLR, EBLR, THR
- MCR =01h: force $\overline{\text{DTR}}$ output to active (low)
- optional: IER = 02h: enable THR interrupt
- TXFLL = 06h: transmit frame length is 6 bytes
- EBLR = 08: transmit 7 starts of frame
- optional: set ACREG[7] = 1: selects SIR pulse width to be 1.6 µs (default 3/16 bit period).
- THR = desired data to be transmitted


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.2.4.2 MIR Mode

In medium infrared (MIR) mode, data transfer takes place between LH and peripheral devices at 0.576 or 1.152 Mbits/s speed. A MIR transmit frame starts with start flags (at least two), followed by a frame data, CRC-16 and ends with a stop flag (see Figure 24-8).

**Figure 24-8. MIR Transmit Frame Format**




On transmit, the MIR state machine attaches start flags, CRC-16, and stop flags. It also looks for 5 consecutive 1s in the frame data and automatically inserts 0 after 5 consecutive 1s (this is called bit stuffing).

On receive, the MIR receive state machine recovers the receive clock, removes the start flags, de-stuffs the incoming data and determines frame boundary with reception of the stop flag. It also checks for errors such as: frame abort, CRC error or frame-length error. At the end of a frame reception, the LH reads the line status register (LSR) to find out possible errors of received frame.

Data can be transferred both ways by the module but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware. Refer to the DISIRRX bit of the auxiliary control register (ACREG[5]) for a description of the logical operation.

**NOTE:** This applies to all three modes SIR, MIR, and FIR.

#### *24.2.4.2.1 MIR Encoder/Decoder*

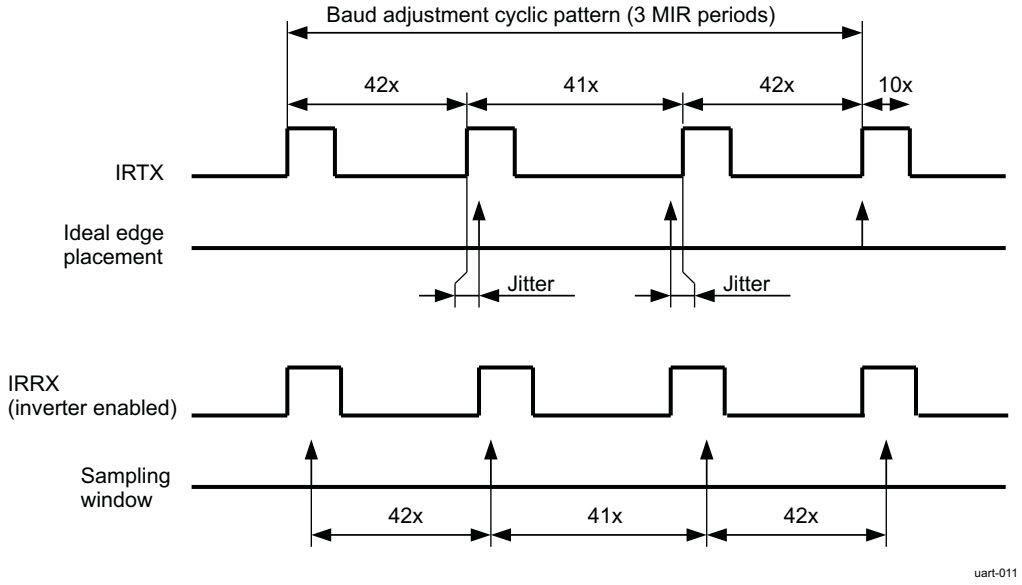
To meet the MIR baud rate tolerance of +/–0.1 percent with a 48-MHz clock input, a 42-41-42 encoding/decoding adjustment is performed. The reference start point is the first start flag, and the 42-41-42 cyclic pattern is repeated until the stop flag is sent or detected.

The jitter created this way is within MIR tolerances. The pulse width is not exactly 1/4, but it is within the tolerances defined by the IrDA specifications.

Figure 24-9 shows the MIR baud rate adjustment mechanism.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 24-9. MIR Baud Rate Adjustment Mechanism**



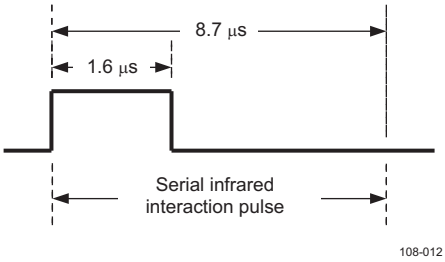

#### *24.2.4.2.2 SIP Generation*

In the MIR and FIR operation modes, the transmitter must send a serial infrared interaction pulse (SIP) at least once every 500 ms. The SIP informs slow devices (operating in SIR mode) that the medium is currently occupied.

When the SIPMODE bit of the mode definition register 1 (MDR1[6]) equals 1, the TX state machine always sends one SIP at the end of a transmission frame. But when MDR1[6] = 0, the transmission of the SIP depends on the SENDSIP bit of the auxiliary control register (ACREG[3]). The system (LH) can set ACREG[3] at least once every 500 ms. The advantage of this approach over the default approach is that the TX state machine does not need to send the SIP at the end of each frame which may reduce the overhead required.

Figure 24-10 shows the SIP.

**Figure 24-10. Serial Infrared Interaction Pulse (SIP)**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 24.2.4.2.3 Programming Models

**Example 24-3. Receive IrDA**

Receive IrDA frame with no parity, baud-rate = 1.152 Mbs, FIFOs disabled.

- MDR1 = 04h
  - MDR1[2:0] = 100b: MIR mode
- LCR = 86h
  - LCR[7] = 1: access to write DLL and DLH
  - LCR[2] = 1: set number of stop bits to 2 (default: 1)
  - LCR[1:0] = 10b: set word length to 7 bits (default: 5)
- DLL = 01h: 1.152 Mbs
- DLH = 0h: 1.152 Mbs
- LCR = 06h: LCR[7] = 0 disables access to DLL and DLH and gives access to MCR, FCR, IER, BLR, EBLR, RHR
- MCR =03h: forces $\overline{\text{DTR}}$ and $\overline{\text{RTS}}$ output to active (low).
- optional: IER = 1: enable RHR interrupt

**Example 24-4. Transmit IrDA**

Transmit IrDA 60 bytes frame with no parity, baud-rate = 1.152 Mbs, FIFOs disabled.

- MDR1 = 04h
  - MDR1[2:0] = 100b: MIR mode
- LCR = 82h
  - LCR[7] = 1: access to write DLL and DLH
  - LCR[1:0] = 10b: set word length to 7 bits (default: 5)
  - optional: LCR[2] = 1: set number of stop bits to 2 (default: 1)
- DLL = 01h: 1.152 Mbs
- DLH = 0h: 1.152 Mbs
- LCR = 02h: LCR[7] = 0 disables access to DLL and DLH and gives access to MCR, FCR, IER, BLR, EBLR, THR
- MCR =01h: force $\overline{\text{DTR}}$ output to active (low)
- optional: IER = 02h: enable THR interrupt
- TXFLL = 3Ch: transmit frame length is 60 bytes
- optional: EBLR = 08: transmit 8 additional starts of frame (MIR mode requires 2 starts anyway)
- optional: set ACREG[3] = 1: SIP sends at the end of transmission
- THR = desired data to be transmitted


Copyright © 2011–2016, Texas Instruments Incorporated

#### 24.2.4.3 FIR Mode

In fast infrared mode (FIR), data transfer takes place between LH and peripheral devices at 4 Mbits/s speed. A FIR transmit frame starts with preamble, followed by a start flag, frame data, CRC-32 and ends with a stop flag.

Table 24-3 shows the FIR transmit frame format.

**Table 24-3. FIR Transmit Frame Format**

| Preamble (16x) | Start flag | Frame data | CRC-32 | Stop flag |
|---|---|---|---|---|

On transmit, the FIR transmit state machine attaches the preamble, start flag, CRC-32, and stop flag. It also encodes the transmit data into 4PPM format. It also generates the serial infrared interaction pulse (SIP).

On receive, the FIR receive state machine recovers the receive clock, removes start flag, decodes the 4PPM incoming data and determines frame boundary with reception of stop flag. It also checks for errors such as: illegal symbol, CRC error and frame-length error. At the end of a frame reception, the LH reads the line status register (LSR) to find out possible errors of the received frame.

Data can be transferred both ways by the module but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware. Refer to the DISIRRX bit of the auxiliary control register (ACREG[5]) for a description of the logical operation.

**NOTE:** This applies to all three modes SIR, MIR, and FIR.

### *24.2.5 CIR Mode*

In consumer infrared (CIR) mode, the infrared operation is designed to function as a programmable (universal) remote control. By setting the MDR1 register, the UART can be set to CIR mode in the same way as the other IrDA modes are set using the MDR1 register. The CIR mode uses a variable pulse width modulation technique (based on multiples of a programmable T period) to encompass the various formats of infrared encoding for remote control applications. The CIR logic is to transmit and receive data packets according to the user definable frame structure and packet content.

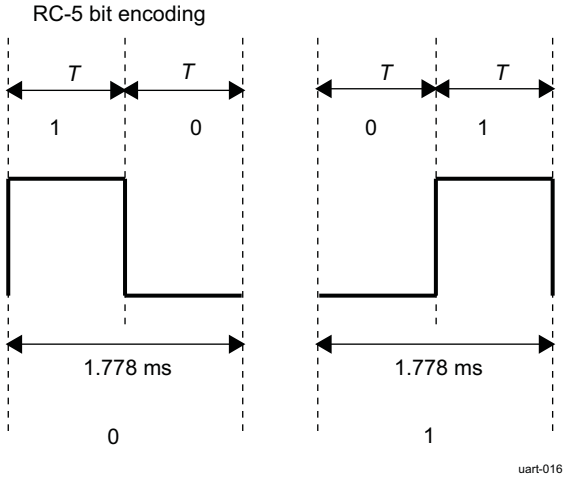
#### 24.2.5.1 Consumer IR Encoding

There are two distinct methods of encoding for remote control applications. The first uses time extended bit forms, that is, a variable pulse distance (or duration) whereby the difference between a logic 1 and logic 0 is the length of the pulse width; and the second is the use of a bi-phase where the encoding of the logic 0 and logic 1 is in the change of signal level from 1 -> 0 or 0 -> 1, respectively. Japanese manufacturers tend to favor the use of pulse duration encoding whereas European manufacturers favor the use of bi-phase encoding.

The CIR mode is designed to use a completely flexible free format encoding where a digit ‘1’ from the TX/RX FIFO is to be transmitted/received as a modulated pulse with duration T. Equally, a ‘0’ is to be transmitted/received as a blank duration T. The protocol of the data is to be constructed and deciphered by the host CPU. For example, the RC-5 protocol (Figure 24-11) using Manchester encoding can be emulated as using a “01” pair for one and “10” pair for a zero.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 24-11. RC-5 Bit Encoding**



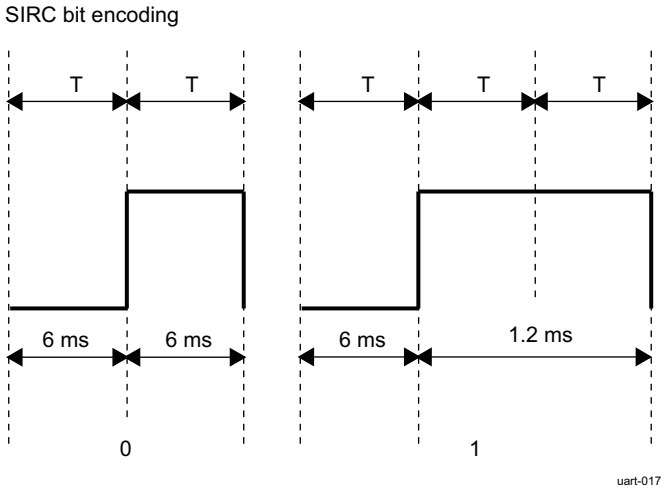

Since the CIR mode logic does not impose a fixed format for infrared packets of data, the CPU software is at liberty to define the format through the use of simple data structures that will then be modulated into an industry standard such as RC-5 or SIRC to name a few. To send a sequence of “0101” in RC-5, the host software must write an eight bit binary character of “10011001” to the data TX FIFO of the UART.

For SIRC, the modulation length (multiples of T) is the method to distinguish between a “1” or a “0”. The following SIRC digits show the difference in encoding between this and RC-5 for example.

**NOTE:** The pulse width is extended for “1” digits.

Figure 24-12 shows SIRC bit encoding.

**Figure 24-12. SIRC Bit Encoding**

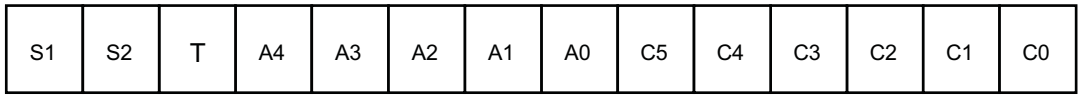



To construct comprehensive packets that constitute remote control commands, the host software must combine a number of 8-bit data characters in a sequence that follows one of the universally accepted formats.

Figure 24-13 shows a standard RC-5 frame as detected by the UART3 in CIR mode (the SIRC format follows this). Each field in RC-5 can be considered as two T pulses (digital bits) from the TX and RX FIFOs.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 24-13. RC-5 Standard Packet Format**




Where:

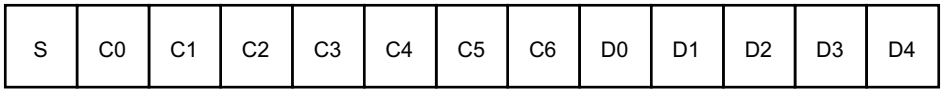
| | |
|---|---|
| S1, S2: | Start bits (always 1) |
| T: | Toggle bit |
| A4..A0: | Address (or system) bits |
| C5..C0: | Command bits |

The toggle bit T changes each time a new command is transmitted to allow detection of pressing the same key twice (or effectively receiving the same data from the host consecutively). Since a code is being sent as long as the CPU transmits characters to the UART for transmission, a brief delay in the transmission of the same command would be detected by the use of the toggle bit. The address bits define the machine or device that the Infrared transmission is intended for and the command defines the operation.

To accommodate an extended RC-5 format, the S2 bit is replaced by an additional command bit (C6) that allows the command range to increase to 7 bits. This format is known as the extended RC-5 format.

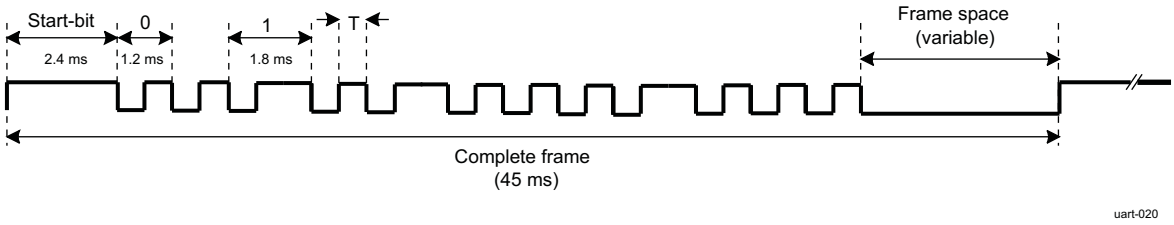
The SIRC encoding uses the duration of modulation for mark and space; hence the duration of data bits inside the standard frame length will vary depending upon the logic 1 content. Figure 24-14 shows the packet format and bit encoding. As Figure 24-14 shows, 1 start bit of 2 ms and control codes are followed by data that constitute the entire frame.

**Figure 24-14. SIRC Packet Format**




**NOTE:** The encoding must take a standard duration, but the contents of the data can vary. This implies that the control software for emitting and receiving data packets must exercise a scheme of interpacket delay, where the emission of successive packets can be done only after a real-time delay has expired.

**Figure 24-15. SIRC Bit Transmission Example**




This document does not describe all encoding methods and techniques; the preceding information shows the considerations required to employ different encoding methods for different industry-standard protocols. See industry-standard documentation for specific methods of encoding and protocol usage.


Copyright © 2011–2016, Texas Instruments Incorporated

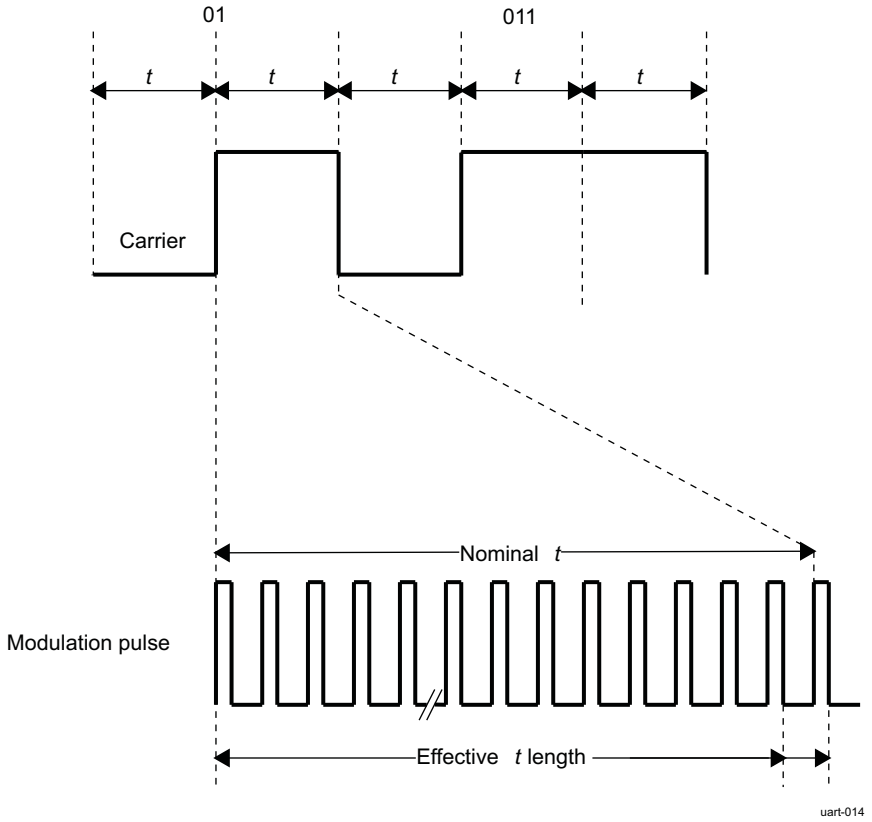
### 24.2.5.2 CIR Mode Operation

In CIR mode, the infrared operation functions as a programmable (universal) remote control. CIR mode uses a variable PWM technique (based on multiples of a programmable t period) to encompass the various formats of infrared encoding for remote-control applications. The CIR logic transmits data packets based on user-defined frame structure and packet content.

### 24.2.5.3 Carrier Modulation

Looking closely at the actual modulation pulses of the infrared data stream, each modulated pulse that constitutes a digit is a train of on/off pulses (see Figure 24-16).

**Figure 24-16. CIR Pulse Modulation**




A minimum of 4 modulation pulses per bit is required by the module. Based on the requested modulation frequency, the CFPS register must be set with the correct dividing value to provide the more accurate pulse frequency:

*Dividing value = (FCLK/12)/MODfreq*

Where:

| | |
|---|---|
| FCLK = | System clock frequency (48 MHz) |
| 12 = | real value of BAUD multiple |
| MODfreq = | Effective frequency of the modulation (MHz) |


Copyright © 2011–2016, Texas Instruments Incorporated

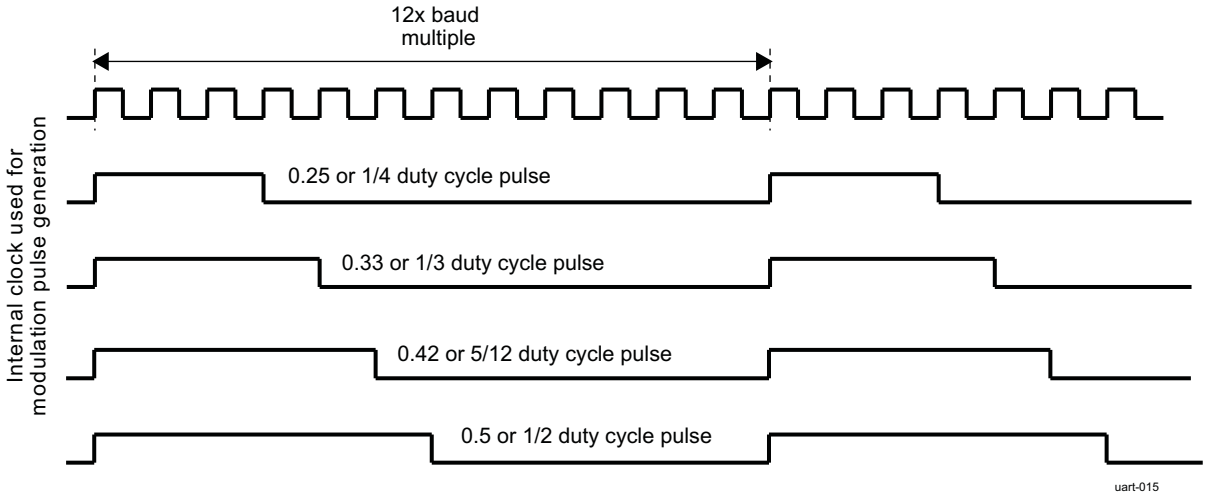
For example, for a targeted modulation frequency of 36 kHz, the CFPS value must be set to 111 in decimal, which provides a modulation frequency of 36.04 kHz.

**NOTE:** The CFPS register starts with a reset value of 105 (decimal), which translates to a frequency of 38.1 kHz.

The duty cycle of these pulses is user-defined by the CIRPULSEMODE field in the mode definition register 2 (MDR2[5:4]).

Figure 24-17 shows the CIR modulation duty cycles.

**Figure 24-17. CIR Modulation Duty Cycle**




The transmission logic ensures that all pulses are transmitted completely (that is, no cutoff during transmission). Furthermore, while transmitting continuous bytes back-to-back, no delay is inserted between 2 transmitted bytes.

NOTE: The CIR RX demodulation can be bypassed by setting the MDR3[0] register bit. This bit will not affect the transmission modulation.

### 24.2.5.4 Frequency Divider Values

As previously explained, the data transferred is a succession of pulses with a T period. Depending on the standards used, the T period is defined through the DLL and DLH registers, which defined the value to divide the functional clock (48 MHz):

*Dividing value = (FCLK/16)/Tfreq*

Where:

| | |
|---|---|
| FCLK = | System clock frequency (48 MHz) |
| 16 = | real value of BAUD multiple |
| Tfreq = | Effective frequency of the T pulse (MHz) |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.2.6 FIFO Management

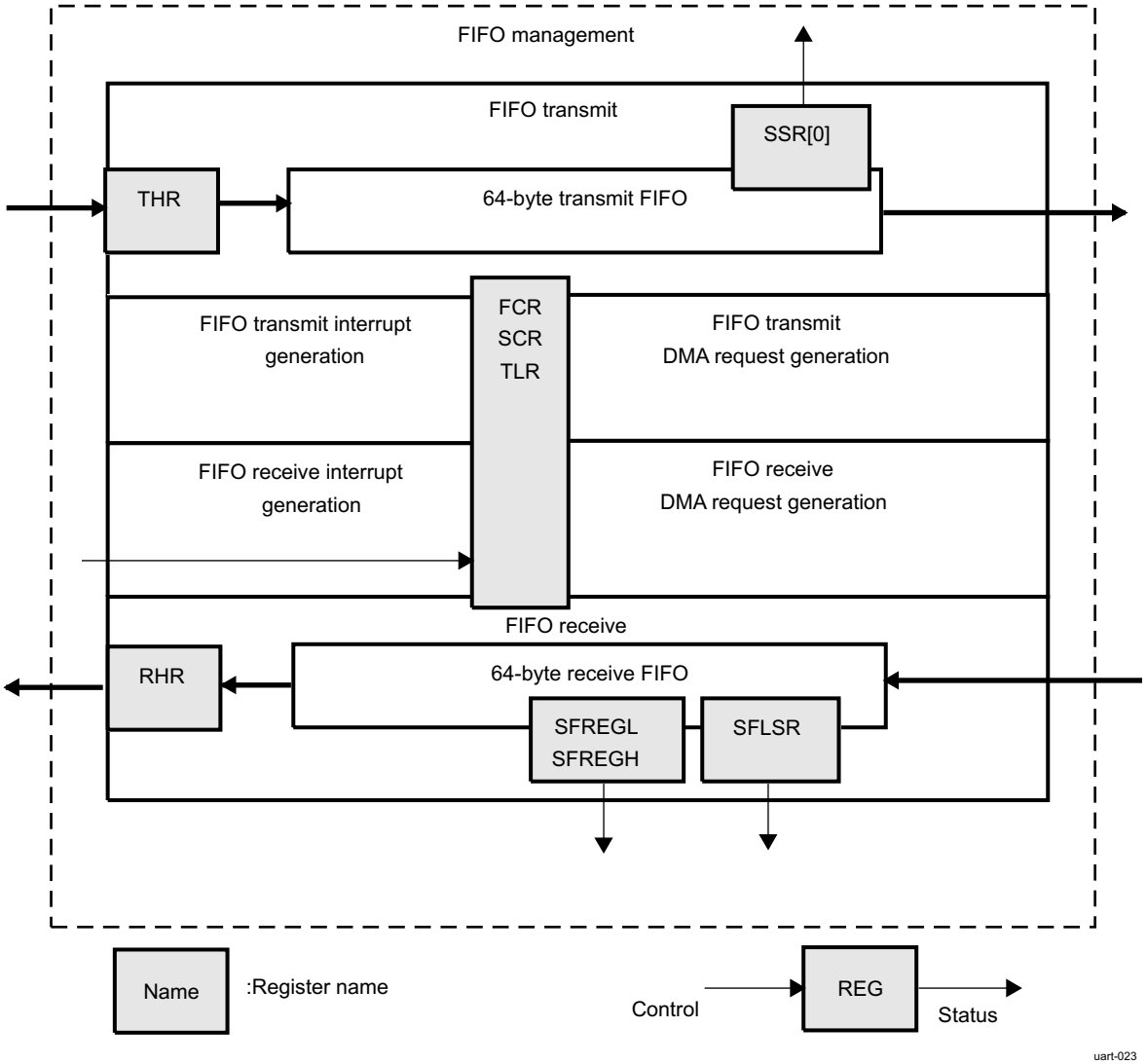
The FIFO is accessed by reading/writing the RHR and THR registers. Parameters are controlled using the FIFO control register (FCR) and supplementary control register (SCR). Reading the TXFIFOFULL bit in SSR[0] as 1 means the FIFO is full.

The TLR register controls the FIFO trigger level, which enables the DMA and interrupt generation. After reset, both transmitter and receiver FIFOs are disabled; so, in effect, the trigger level is the default value of 1 byte. Figure 24-18 shows the FIFO management registers.

**NOTE:** Data in the RHR register is not overwritten when an overflow occurs.

The SFLSR, SFREGL, and SFREGH status registers are used in IrDA mode only.

**Figure 24-18. FIFO Management**





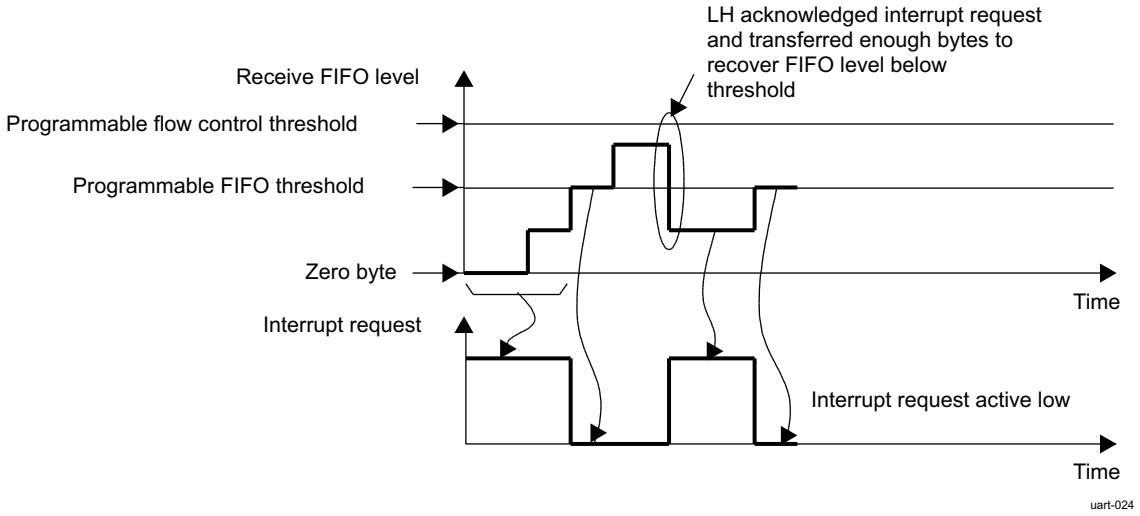
Copyright © 2011–2016, Texas Instruments Incorporated

### 24.2.6.1 FIFO Interrupt Mode

In the FIFO interrupt mode (FIFO_EN bit in the FIFO control register (FCR[0]) is set to 1 and relevant interrupts are enabled via the interrupt enable register (IER)), an interrupt signal informs the processor of the status of the receiver and transmitter. These interrupts are raised when the receive/transmit FIFO threshold (TLR[7:4] and TLR[3:0] fields or the FCR[7:6] and FCR[5:4] fields, respectively) are reached; the interrupt signals instruct the Local Host to transfer data to the destination (from the UART module in receive mode and/or from any source to the UART FIFO in transmit mode). Note also that in the case of the UART flow control being enabled along with the interrupt capabilities, the user must ensure that the UART flow control FIFO threshold (TCR[3:0]) is greater than or equal to the receive FIFO threshold.

Figure 24-19 shows the generation of the receive FIFO interrupt request.

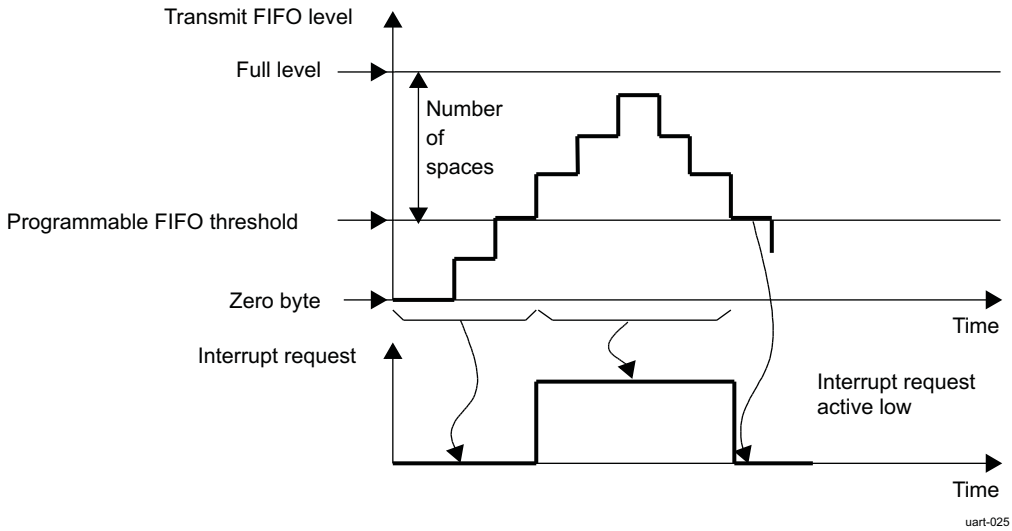
**Figure 24-19. Receive FIFO Interrupt Request Generation**




In receive mode, no interrupt is generated until the receive FIFO reaches its threshold. Once low, the interrupt can be de-asserted only when the Local Host has handled enough bytes to make the FIFO level below threshold. The flow control threshold is set at a higher value than the FIFO threshold.

Figure 24-20 shows the generation of the transmit FIFO interrupt request.

**Figure 24-20. Transmit FIFO Interrupt Request Generation**




In transmit mode, an interrupt request is automatically asserted when the TX FIFO is empty. This request is de-asserted when the TX FIFO crosses the threshold level. The interrupt line is de-asserted until a sufficient number of elements is transmitted to go below the TX FIFO threshold.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 24.2.6.2 FIFO Polled Mode Operation

In FIFO polled mode (FCR[0] = 0, relevant interrupts disabled via interrupt enable register (IER)) the status of the receiver and transmitter can then be checked by polling the line status register (LSR). This mode is an alternative to the FIFO interrupt mode of operation where the status of the receiver and transmitter is automatically known by means of interrupts sent to the LH.

#### 24.2.6.3 FIFO DMA Mode Operation

##### *24.2.6.3.1 DMA Signaling*

There are four modes of DMA operation, DMA mode 0/1/2/3. They can be selected as follows:

- When SCR[0] = 0: setting FCR[3] to 0 enables DMA mode 0, setting FCR[3] to 1 enables DMA mode 1.
- When SCR[0] = 1: SCR[2:1] determines DMA mode 0 to 3, according to supplementary control register (SCR).

So for instance:

- if no DMA operation is desired: set SCR[0] to 1 and SCR[2:1] to 00 (FCR[3] is discarded)
- if DMA mode 1 is desired: either set SCR[0] to 0 and FCR[3] to 1 or set SCR[0] to 1 and SCR[2:1] to 01 (FCR[3] is discarded)

If the FIFOs are disabled (FCR[0] = 0), DMA occurs in single character transfers.

**NOTE:** Note that when DMA mode 0 has been programmed, the signals associated with DMA operation are not active.

##### *24.2.6.3.2 DMA Transfers (DMA Mode 1, 2, or 3)*

Figure 24-21, Figure 24-22, Figure 24-23, and Figure 24-24 show the supported DMA operations.

In receive mode, a DMA request is generated as soon as the receive FIFO reaches its threshold level defined in the trigger level register (TLR). This request is de-asserted when the number of bytes defined by the threshold level has been read by the system DMA.

In transmit mode, a DMA request is automatically asserted when the transmit FIFO is empty. This request is de-asserted when the number of bytes defined by the number of spaces in the trigger level register (TLR) has been written by the system DMA. If an insufficient number of characters are written, then the DMA request will remain active.

The DMA request is again asserted if the FIFO can receive the number of bytes defined by the TLR register.

There are a number of ways the threshold can be programmed. The example in Figure 24-22 is of a DMA transfer that operates with a space setting of 56, which could arise from the use of the auto settings in the FCR[5:4] or the use of the TLR[3:0] concatenated with the FCR[5:4]. The setting of 56 spaces in the UART/IrDA/CIR module should correlate with settings of the system DMA so that the buffer does not overflow (program the DMA request size of the Local Host controller to be equal to the number of spaces value in the UART/IrDA/CIR module).


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 24-21. Receive FIFO DMA Request Generation (32 Characters)**

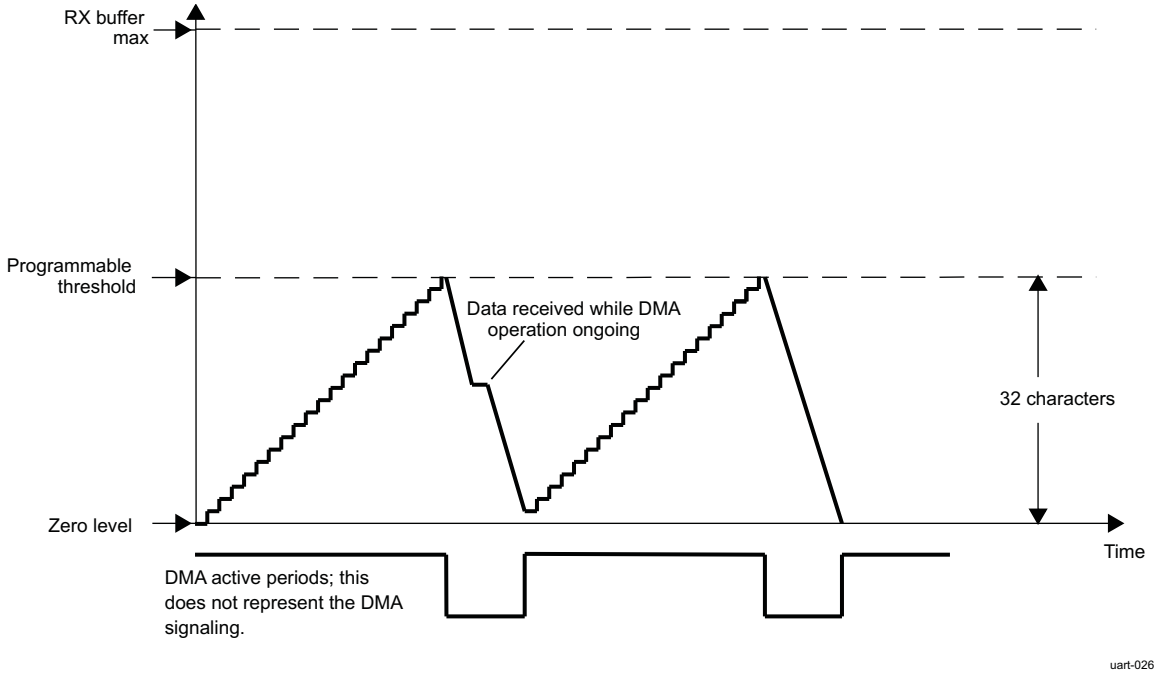
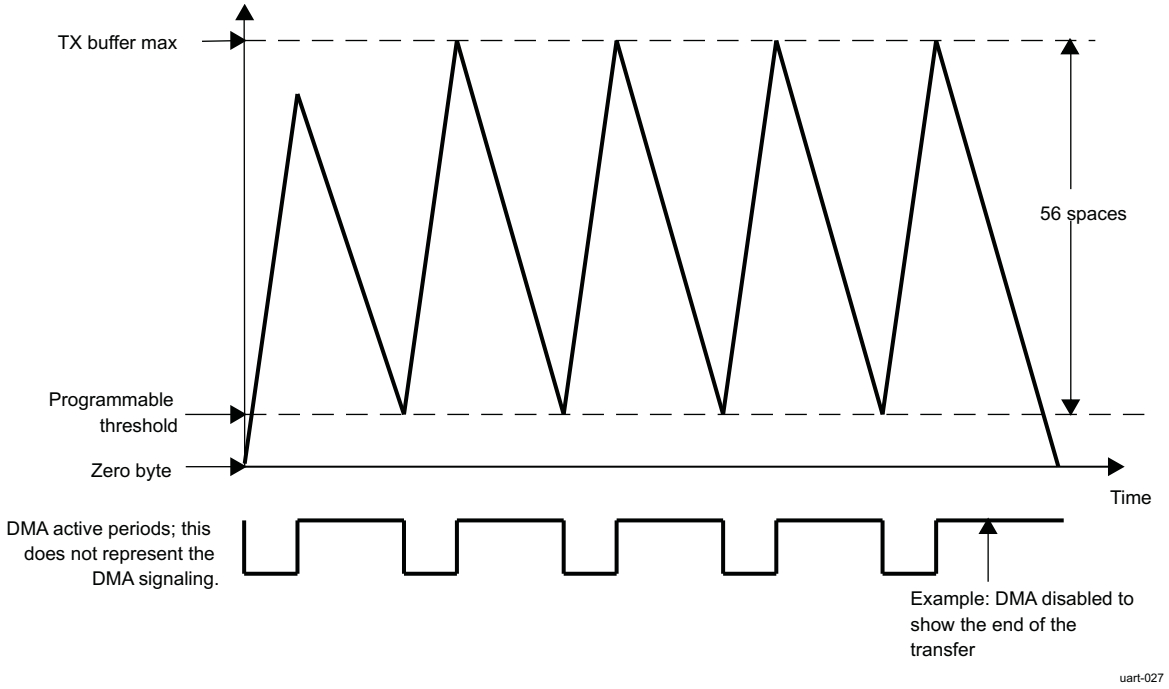



**Figure 24-22. Transmit FIFO DMA Request Generation (56 Spaces)**

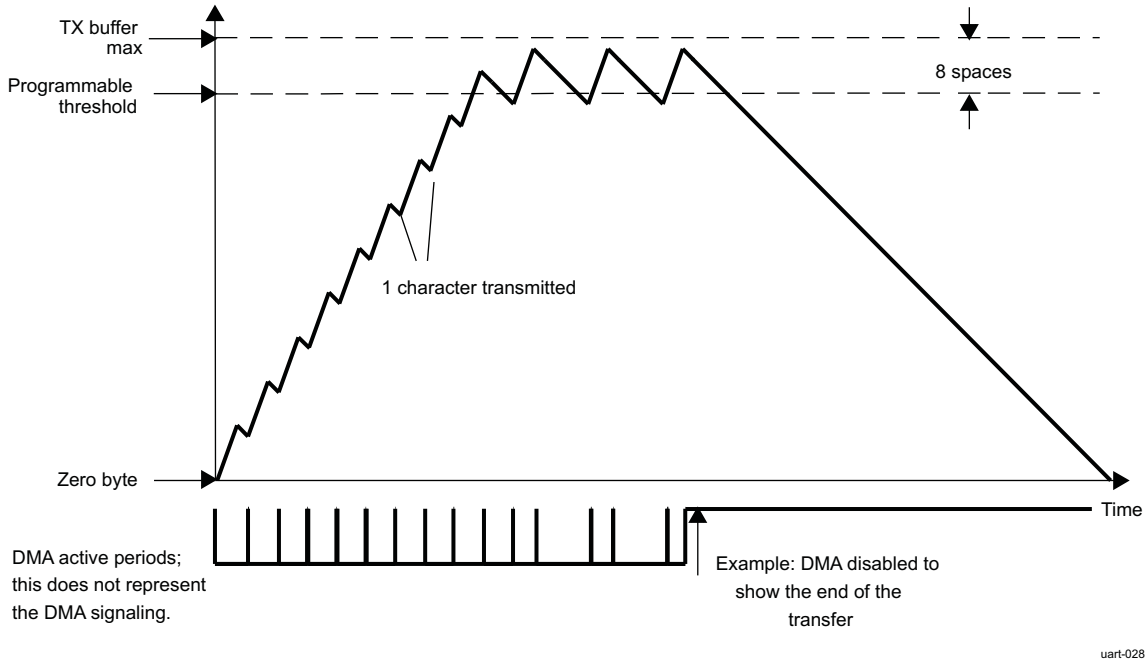




Copyright © 2011–2016, Texas Instruments Incorporated

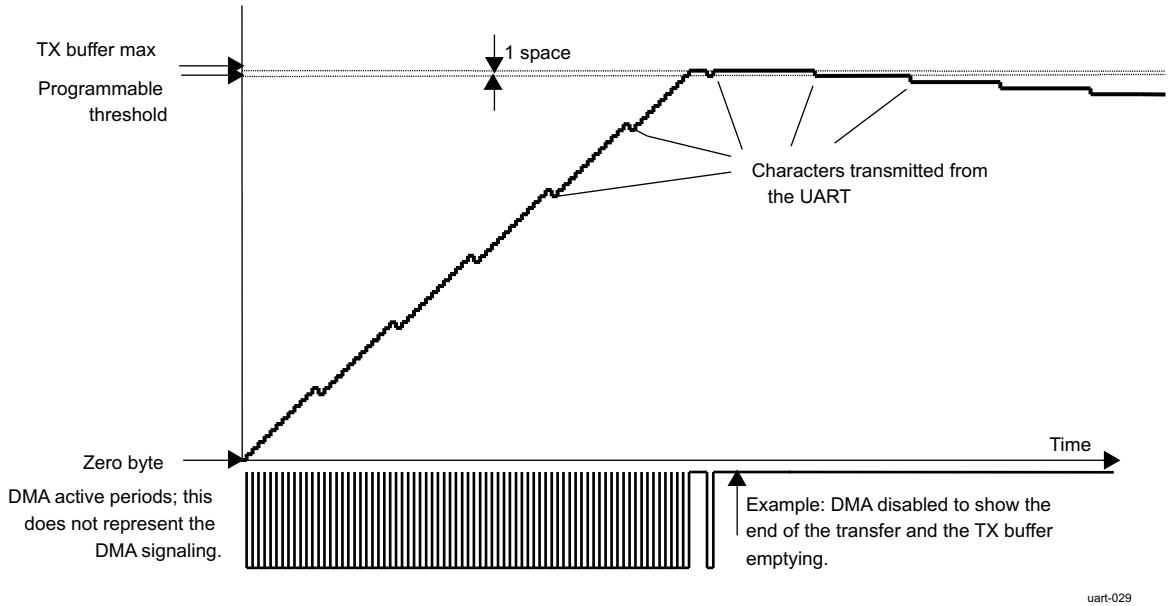
Figure 24-23 shows an example with 8 spaces to show the buffer level crossing the space threshold. Again, the Local Host DMA controller settings must correspond to that of the UART/IrDA/CIR module.

**Figure 24-23. Transmit FIFO DMA Request Generation (8 Spaces)**




The final example illustrates the setting of one space which uses the DMA for each transfer of one character to the transmit buffer (see Figure 24-24). The buffer is filled at a faster rate than the BAUD rate transmits data to the TX pin. Eventually, the buffer is completely full and the DMA operations stop transferring data to the transmit buffer. There are two occasions where the buffer holds the maximum amount of data words, shortly after this, the DMA is disabled to illustrate the slower transmission of the data words to the TX pin. Eventually, the buffer will be emptied at the rate specified by the BAUD Rate Settings of the DLL and DLH registers. Again, the DMA settings should correspond to the system's Local Host DMA controller settings in order to ensure the correct operation of this logic.

**Figure 24-24. Transmit FIFO DMA Request Generation (1 Space)**



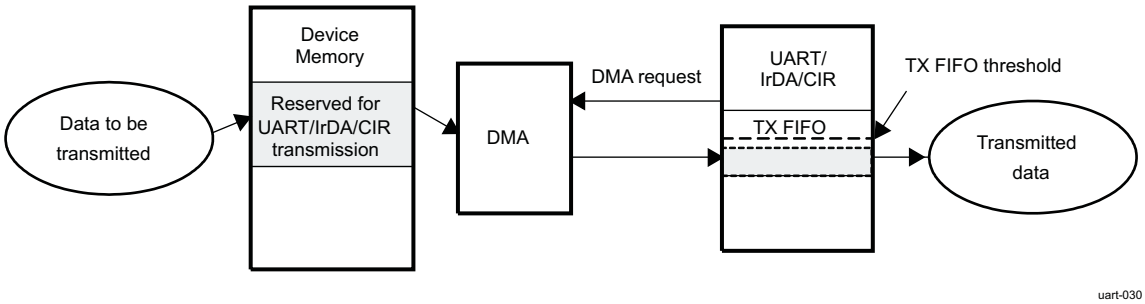


Copyright © 2011–2016, Texas Instruments Incorporated

#### 24.2.6.3.3 DMA Transmission

Figure 24-25 shows the DMA transmission process.

**Figure 24-25. Transmission Process**



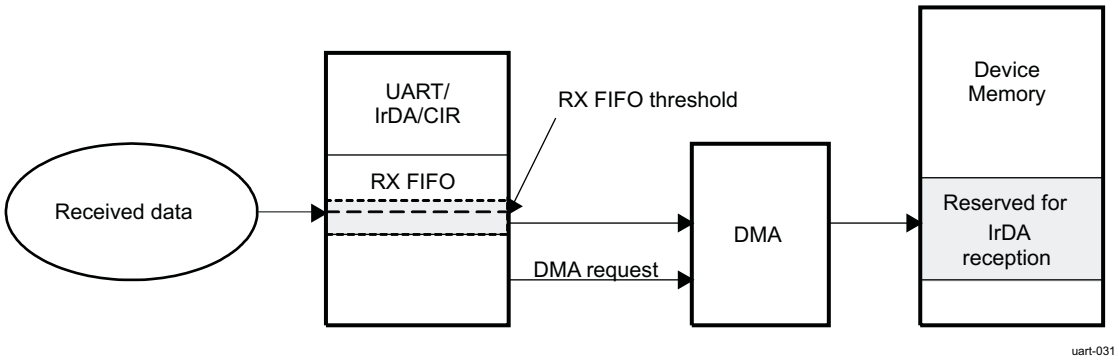

- Data to be transmitted are put in the memory reserved for UART/IrDA/CIR transmission by the DMA:
  - Until the TX FIFO trigger level is not reached, a DMA request is generated.
  - An element (1 byte) is transferred from the SDRAM to the TX FIFO at each DMA request (DMA element synchronization).
- Data in the TX FIFO are automatically transmitted.
- The end of the transmission is signaled by the UARTi.THR empty (TX FIFO empty).

**NOTE:** In IrDA mode, the transmission is not ended immediately after the TX FIFO empties, at which point there are still the last data byte, the CRC field, and the stop flag to be transmitted; thus, the end of transmission is a few milliseconds after the UARTi.THR register empties.

#### 24.2.6.3.4 DMA Reception

Figure 24-26 shows the DMA reception process.

**Figure 24-26. Reception Process**




1. Enable the reception.
2. Received data are put in the RX FIFO.
3. Data are transferred from the RX FIFO to the device memory by the DMA.
   - At each received byte, the RX FIFO trigger level (one character) is reached and a DMA request is generated.
   - An element (1 byte) is transferred from the RX FIFO to the SDRAM at each DMA request (DMA element synchronization).
4. The end of the reception is signaled by the EOF interrupt.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 24.2.6.4 IrDA Status FIFO

In IrDA modes, a status FIFO is used to record the received frame status. When a complete frame is received, the length of the frame and the error bits associated with the frame are written into the status FIFO. The frame length and error status can be read by reading SFREGL/H and SFLSR. Reading the SFLSR causes the read pointer to be incremented. The status FIFO is eight entries deep and therefore can hold the status of eight frames. The LH uses the frame-length information to locate the frame-boundary in the received frame data. The LH can screen bad frames using the error-status information and later request the sender to resend only the bad frames. This status FIFO can be used very effectively in DMA as the LH need not be interrupted every time a frame is received but only whenever the programmed status FIFO trigger level is reached.

#### 24.2.6.5 Frame Closing

There are two ways by which a transmission-frame can be properly terminated:

- Frame-length method: Frame-length method is selected when MDR1[7] = 0. The LH writes the frame-length value to TXFLH and TXFLL registers. The device automatically attaches end flags to the frame once the number of bytes transmitted becomes equal to the frame-length value.
- Set-EOT bit Method: Set-EOT bit method is selected when MDR1[7] = 1. The LH writes 1 to ACREG[0] (EOT bit) just before it writes the last byte to the TX FIFO. When the LH writes the last byte to the TX FIFO, the device internally sets the tag bit for that particular character in the TX FIFO. As the TX state machine reads data from the TX FIFO it uses this tag-bit information to attach end flags and properly terminate the frame.

#### 24.2.6.6 Store and Controlled Transmission (SCT)

In SCT the LH first starts writing data into the TX FIFO. Then, after it writes a part of a frame (for a bigger frame) or a whole frame (a small frame, that is, supervisory frame), it writes a 1 to ACREG[2] (deferred TX start) to start transmission. SCT is enabled when MDR1[5] = 1. This method of transmission is different from the normal mode, where transmission of data starts immediately after data is written to the TX FIFO. SCT is useful to send short frames without TX underrun.

#### 24.2.6.7 Underrun During Transmission

Underrun in transmission occurs when the TX FIFO becomes empty before the end of the frame is transmitted. When underrun occurs, the device closes the frame with end-flags but attaches an incorrect CRC value. The receiving device detects a CRC error and discards the frame; it can then ask for a re-transmission. Underrun also causes an internal flag to be set which disables further transmission. Before the next frame can be transmitted the system (LH) must:

- Reset the TX FIFO
- Read the RESUME register. This clears the internal flag.

This functionality can be disabled with ACREG[4], compensated by the extension of the stop bit in transmission in case the TX FIFO is empty.

#### 24.2.6.8 Overrun During Receive

Overrun occurs during receive if the RX state machine tries to write data into the RX FIFO when it is already full. When overrun occurs, the device interrupts the LH with IIR[3] and discards the remaining portion of the frame. Overrun also causes an internal flag to be set, which disables further reception. Before the next frame can be received the system (LH) must:

- Reset the RX FIFO
- Read the RESUME register. This clears the internal flag.


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.2.7 Interrupts

The UART/IrDA/CIR module generates interrupts. All interrupts can be enabled/disabled by writing to the appropriate bit in the interrupt enable register (IER). The interrupt status of the device can be checked at any time by reading the interrupt identification register (IIR). The UART, IrDA and CIR modes have different interrupts in the UART/IrDA/CIR module and therefore different IER and IIR mappings according to the selected mode.

#### 24.2.7.1 Trigger Levels

The UART provides programmable trigger levels for both receiver and transmitter DMA and Interrupt generation. After reset, both transmitter and receiver FIFOs are disabled; so, in effect, the trigger level is the default value of one byte. The programmable trigger levels are an enhanced feature available via the trigger level register (TLR).

#### 24.2.7.2 UART Mode Interrupts

In UART modes, there are seven possible interrupts. These interrupts are prioritized to six different levels. When an interrupt is generated, the interrupt identification register (IIR) indicates that an interrupt is pending by bringing IIR[0] to 0 and provides the type of interrupt through IIR[5-1]. Table 24-4 summarizes the interrupt control functions.

It is important to note that for the receiver line status interrupt, RX_FIFO_STS bit (LSR[7]) generates the interrupt. For the XOFF interrupt, if a XOFF flow character detection caused the interrupt, the interrupt is cleared by a XON flow character detection. If special character detection caused the interrupt, the interrupt is cleared by a read of the IIR.

**Table 24-4. UART Mode Interrupts**

| IIR[5:0] | Priority Level | Interrupt Type | Interrupt Source | Interrupt Reset Method |
|---|---|---|---|---|
| 00 0001 | None | None | None | None |
| 00 0110 | 1 | Receiver line status | OE, FE, PE, or BI errors occur in characters in the RX FIFO. | FE, PE, BI: Read RHR register. OE: Read LSR register. |
| 00 1100 | 2 | RX time-out | Stale data in RX FIFO | Read RHR register. |
| 00 0100 | 2 | RHR interrupt | DRDY (data ready) (FIFO disable)<br>RX FIFO above trigger level (FIFO enable) | Read RHR register until interrupt condition disappears. |
| 00 0010 | 3 | THR interrupt | TFE (THR empty) (FIFO disable)<br>TX FIFO below trigger level (FIFO enable) | Write to THR register until interrupt condition disappears. |
| 00 0000 | 4 | Modem status | See the MSR register. | Read MSR register. |
| 01 0000 | 5 | XOFF interrupt/special character interrupt | Receive XOFF characters/special character | Receive XON character(s), if XOFF interrupt. Read IIR register, if special character interrupt. |
| 10 0000 | 6 | CTS, RTS, DSR | $\overline{\text{RTS}}$ pin, $\overline{\text{CTS}}$ pin, or $\overline{\text{DSR}}$ pin changes state from active (low) to inactive (high). | Read IIR register. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.2.7.3 IrDA Mode Interrupts

In IrDA modes, there are eight possible interrupts. The interrupt line is activated when any of the eight interrupts is generated (there is no priority). For IIR[5], interrupt source 1 is used with interrupt reset method 1 and interrupt source 2 is used with interrupt reset method 2. Table 24-5 summarizes the interrupt control functions.

**Table 24-5. IrDA Mode Interrupts**

| IIR Bit | Interrupt Type | Interrupt Source | Interrupt Reset Method |
|---|---|---|---|
| 0 | RHR interrupt | DRDY (data ready) (FIFO disable)<br>RX FIFO above trigger level (FIFO enable) | Read RHR register until interrupt condition disappears. |
| 1 | THR interrupt | TFE (THR empty) (FIFO disable)<br>TX FIFO below trigger level (FIFO enable) | Write to THR register until interrupt condition disappears. |
| 2 | Last byte in RX FIFO | Last byte of frame in RX FIFO is available to be read at the RHR port. | Read RHR register. |
| 3 | RX overrun | Write to RHR register when RX FIFO full. | Read RESUME register. |
| 4 | Status FIFO interrupt | Status FIFO triggers level reached. | Read SFREGL/H, SFLSR. Status FIFO read pointer is incremented only when reading SFLSR. |
| 5 | TX status (indicated by MDR2[0] (IRTX_UNDERRUN) | 1. THR empty before EOF sent. Last bit of transmission of the IrDA frame occurred, but with an underrun error. OR<br>2. Transmission of the last bit of the IrDA frame completed successfully. | 1. Read RESUME register. OR<br>2. Read IIR register. |
| 6 | Receiver line status interrupt | CRC, ABORT, or frame-length error is written into STATUS FIFO. | Read STATUS FIFO (read until empty - maximum of eight reads required). |
| 7 | Received EOF | Received end-of-frame. | Read IIR register. |

### 24.2.7.4 CIR Mode Interrupts

The CIR mode uses a subset of the existing IrDA mode interrupts. Table 24-6 summarizes the interrupt modes that are to be maintained. In CIR mode, IIR bit 5 has a single purpose of indicating that the last bit of infrared data has been passed to the IR TX pin.

**Table 24-6. CIR Mode Interrupts**

| IIR Bit | Interrupt Type | Interrupt Source | Interrupt Reset Method |
|---|---|---|---|
| 0 | RHR interrupt | DRDY (data ready) (FIFO disable)<br>RX FIFO above trigger level (FIFO enable) | Read RHR register until interrupt condition disappears. |
| 1 | THR interrupt | TFE (THR empty) (FIFO disable)<br>TX FIFO below trigger level (FIFO enable) | Write to THR register until interrupt condition disappears. |
| 2 | RXSTOPIT interrupt | Receive stop interrupt (depending on value set in the BOF length register (EBLR)) | Read IIR register. |
| 3 | RXOEIT interrupt | Write to RHR register when RX FIFO full. | Read RESUME register. |
| 4 | N/A for CIR mode | | |
| 5 | TX status | Transmission of the last bit of the frame is completed successfully. | Read IIR register. |
| 6-7 | N/A for CIR mode | | |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 24.2.7.5 Wake-Up Interrupt

Wake-up interrupt is a special interrupt, which is not designed in the same way than previous ones. This interrupt is enabled when the RX_CTS_DSR_WAKE_UP_ENABLE bit of the supplementary control register (SCR[4]) is set to one. The IIR register is not modified when it occurs, SSR[1] must be checked to detect a wake-up-event. When wake-up interrupt occurs, the only way to clear it, is to reset SCR[4] to 0. Wake-up can also occur if the WER[7] TX_WAKEUP_EN is set to 1 and one of the followings occurred:

- THR interrupt occurred if it is enabled (omitted if TX DMA request is enabled)
- TX DMA request occurred if it is enabled
- TX_STATUS_IT occurred if it is enabled (only IrDA and CIR modes). Can not be used with THR interrupt

### *24.2.8 Sleep Modes*

#### 24.2.8.1 When in UART Mode

In UART modes, sleep mode is enabled by writing a 1 to IER[4] (when EFR[4] = 1). Sleep mode is entered when:

- The serial data input line, RX is idle
- The TX FIFO and TX shift register are empty
- The RX FIFO is empty
- There are no interrupts pending except THR interrupts

It should be noted that sleep mode is a good way to lower power consumption of UART but this state can be achieved only when the UART is set in modem mode. Therefore, even if the UART has no functional key role, it must be initialized in a functional mode to take advantage of sleep mode. In sleep mode the module clock and baud rate clock are stopped internally. Because most registers are clocked using these clocks the power consumption is greatly reduced. The module wakes up when any change is detected on the RX line, if data is written to the TX FIFO, when there is any change in the state of the modem input pins. Note that an interrupt can be generated on a wake up event by setting SCR[4] to 1. See interrupt section to know how to manage it.

**NOTE:** Writing to the divisor latches, DLL and DLH, to set the baud clock, BCLK, must not be done during sleep mode. Therefore, it is advisable to disable sleep mode using IER[4] before writing to DLL or DLH.

#### 24.2.8.2 When in IR-IrDA and CIR Modes

In IrDA/CIR modes, sleep mode is enabled by writing a 1 to MDR1[3]. Sleep mode is entered when:

- The serial data input line, RX is idle
- The TX FIFO and TX shift register are empty
- The RX FIFO is empty
- There are no interrupts pending except THR interrupts

The module wakes up when any change is detected on the RX line or if data is written to the TX FIFO.

### *24.2.9 Idle Modes*

Sleep and AutoIdle modes are embedded power-saving features. At the system level, power reduction techniques can be applied by shutting down certain internal clock and power domains of the device.

- The UART supports an idle req - idle ack handshaking protocol. This protocol is used at system level to shut down clocks of the UART in a clean and controlled manner and to switch the UART from the interrupt generation mode to a wakeup generation mode for unmasked events (Refer to SYSC[2] and WER).
- In the wakeup generation mode, interrupt request generation and DMA request generation are disabled.


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.2.10 Programmable Baud Rate Generator

The UART/rRDA/CIR module contains a programmable baud generator and a set of fixed dividers that takes the 48 MHz clock input and divides it down to the expected baud-rate. The baud rate generator and associated controls is shown in Figure 24-27.

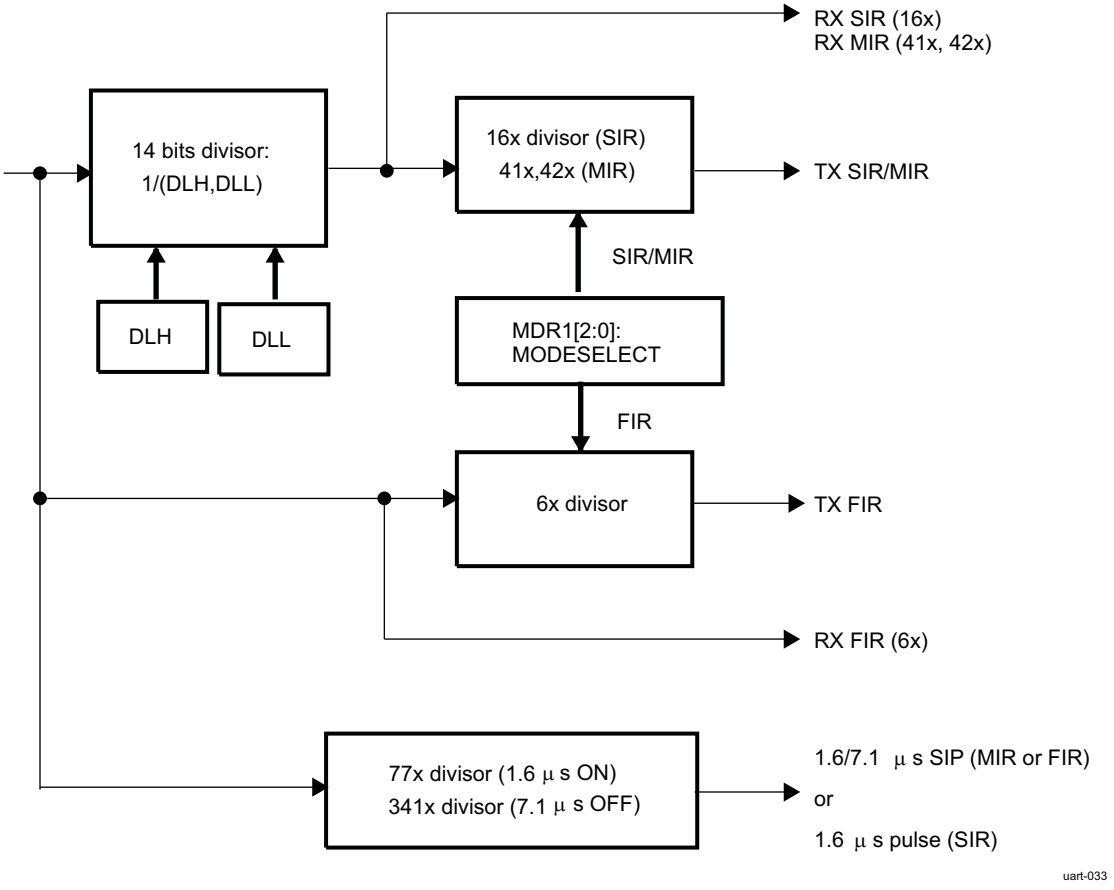
**CAUTION**

It is mandatory that the MODESELECT bit in MDR1[2:0] = 7h (disable) before initializing or modifying clock parameter controls (DLH and DLL). Failure to observe this rule can result in unpredictable module behavior.

Choosing the appropriate divisor value:

- UART 16× mode: Divisor value = Operating Frequency/(16× baud rate)
- UART 13× mode: Divisor value = Operating Frequency/(13× baud rate)
- SIR mode: Divisor value = Operating Frequency/(16× baud-rate)
- MIR mode: Divisor value = Operating Frequency/(41×/42× baud-rate)
- FIR mode: Divisor value = none

**Figure 24-27. BAUD Rate Generator**





Copyright © 2011–2016, Texas Instruments Incorporated

### 24.2.10.1 UART Baud Rates (48 MHz Clock)

Table 24-7 describes the UART baud rate settings.

**Table 24-7. UART Baud Rate Settings (48-MHz Clock)**

| Baud Rate | Baud Multiple | DLH, DLL (Decimal) | DLH, DLL (Hex) | Actual Baud Rate | Error (%) |
|---|---|---|---|---|---|
| 0.3 Kbps | 16x | 10000 | 27h, 10h | 0.3 Kbps | 0 |
| 0.6 Kbps | 16x | 5000 | 13h, 88h | 0.6 Kbps | 0 |
| 1.2 Kbps | 16x | 2500 | 09h, C4h | 1.2 Kbps | 0 |
| 2.4 Kbps | 16x | 1250 | 04h, E2h | 2.4 Kbps | 0 |
| 4.8 Kbps | 16x | 625 | 02h, 71h | 4.8 Kbps | 0 |
| 9.6 Kbps | 16x | 312 | 01h, 38h | 9.6153 Kbps | +0.16 |
| 14.4 Kbps | 16x | 208 | 00h, D0h | 14.423 Kbps | +0.16 |
| 19.2 Kbps | 16x | 156 | 00h, 9Ch | 19.231 Kbps | +0.16 |
| 28.8 Kbps | 16x | 104 | 00h, 68h | 28.846 Kbps | +0.16 |
| 38.4 Kbps | 16x | 78 | 00h, 4Eh | 38.462 Kbps | +0.16 |
| 57.6 Kbps | 16x | 52 | 00h, 34h | 57.692 Kbps | +0.16 |
| 115.2 Kbps | 16x | 26 | 00h, 1Ah | 115.38 Kbps | +0.16 |
| 230.4 Kbps | 16x | 13 | 00h, 0Dh | 230.77 Kbps | +0.16 |
| 460.8 Kbps | 13x | 8 | 00h, 08h | 461.54 Kbps | +0.16 |
| 921.6 Kbps | 13x | 4 | 00h, 04h | 923.08 Kbps | +0.16 |
| 1.843 Mbps | 13x | 2 | 00h, 02h | 1.846 Mbps | +0.16 |
| 3.6884 Mbps | 13× | 1 | 00h, 01h | 3.6923 Mbps | +0.16 |
| 3.0 Mbps | 16x | 1 | 00h, 01h | 3.0 Mbps | 0 |

### 24.2.10.2 IrDA Baud Rates (48 MHz Clock)

Table 24-8 lists the IrDA baud rate settings.

**Table 24-8. IrDA Baud Rates Settings**

| Baud Rate | IR Mode | Baud Multiple | Encoding | DLH, DLL (Decimal) | Actual Baud Rate | Error (%)(1) | Source Jitter (%) | Pulse Duration |
|---|---|---|---|---|---|---|---|---|
| 2.4 Kbps | SIR | 16× | 3/16 | 1250 | 2.4 Kbps | 0 | 0 | 78.1 µs |
| 9.6 Kbps | SIR | 16x | 3/16 | 312 | 9.6153 Kbps | +0.16 | 0 | 19.5 µs |
| 19.2 Kbps | SIR | 16x | 3/16 | 156 | 19.231 Kbps | +0.16 | 0 | 9.75 µs |
| 38.4 Kbps | SIR | 16x | 3/16 | 78 | 38.462 Kbps | +0.16 | 0 | 4.87 µs |
| 57.6 Kbps | SIR | 16x | 3/16 | 52 | 57.692 Kbps | +0.16 | 0 | 3.25 µs |
| 115.2 Kbps | SIR | 16x | 3/16 | 26 | 115.38 Kbps | +0.16 | 0 | 1.62 µs |
| 0.576 Mbps | MIR | 41x/42x | 1/4 | 2 | 0.5756 Mbps(2) | 0 | -2.0375 | 416 ns |
| 1.152 Mbps | MIR | 41x/42x | 1/4 | 1 | 1.1511 Mbps(2) | 0 | -2.0375 | 208 ns |
| 4 Mbps | FIR | 6x | 4PPM | - | 4 Mbps | 0 | 0 | 125 ns |

(1) Baud rate error and source jitter table values do not include 48 MHz reference clock error and jitter
(2) Average value


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.2.10.3 Autobauding Modes

In autobauding mode, the UART extracts transfer characteristics (speed, length, and parity) from an AT command. These characteristics are used to receive data after an “at” (AT) and to send data.

The valid AT commands follow:

| | | |
|---|---|---|
| AT | DATA | <CR> |
| at | DATA | <CR> |
| A/ | | |
| a/ | | |

A line break during the acquisition of the sequence AT is not recognized, and echo functionality is not implemented in hardware. A/ and a/ are not used to extract characteristics, but they must be recognized because of their special meaning. Either A/ or a/ is used to instruct the software to repeat the last received AT command; therefore, an a/ always follows an AT, and transfer characteristics are not expected to change between an AT and an a/.

When a valid AT is received, AT and all subsequent data, including the final <CR> (0Dh), are saved to RX FIFO. The autobaud state-machine waits for the next valid AT command. If an a/ (A/) is received, the a/ (A/) is saved into RX FIFO and the state-machine waits for the next valid AT command.

On the first successful detection of the baud rate, the UART activates an interrupt to signify that the AT (upper or lower case) sequence is detected. The UASR register reflects the correct settings for the baud rate detected. The interrupt activity continues in this way each time a subsequent character is received. Therefore, it is recommended that the software enables the RHR interrupt when using the autobaud mode.

The following settings are detected in autobaud mode with a module clock of 48 MHz:

- Speed: 115.2k baud, 57.6k baud, 38.4k baud, 28.8k baud, 19.2k baud, 14.4k baud, 9.6k baud, 4.8k baud, 2.4k baud, or 1.2k baud
- Length: 7 or 8 bits
- Parity: Odd, even, or space

**NOTE:** The combination of 7-bit character + space parity is not supported.

The method used to identify the speed is:

- Detect the transition 1->0 on the received data. This happens as soon as a stop to start bit transition occurs. The transition is valid after a majority vote on 3 sampling period.
- Sample the start bit duration with 115 200 × 16 Hz clock frequency as long as there is no rising edge. A transition 0->1 is considered as valid after a majority vote on 3 sampling period.
- Compare the sampled value with a table. If the sampled value is outside a valid range an error is reported (no speed identified). And the hardware goes back to the first state (1).
- Else store the first data bit in the received register (for serial to parallel conversion) and go to frame format identification.

The next received bits are sampled according to the programmed baud rate. However, after seven bits reception, the speed identification must be restarted since we may receive several “a” or “A” character before a valid “t” or “T” character.

Autobauding mode is selected when the MODESELECT field in MDR1[2:0] is set to 2h. In the UART autobauding mode, DLL, DLH, and LCR[5:0] settings are not used; instead, UASR is updated with the configuration detected by the autobauding logic.


Copyright © 2011–2016, Texas Instruments Incorporated

## 24.3 UART Registers

Each register is selected using a combination of address and, for some, LCR register bit settings as shown in Table 24-9. The following registers are accessible by the local host (LH) at address = module base address + address offset. The module base address is the module start address. Note that register address offsets depends upon the module address alignment at the system top level.

**Table 24-9. UART Registers**

| Address Offset | Registers | | | | | |
|---|---|---|---|---|---|---|
| | LCR[7] = 0 | | LCR[7:0] ≠ BFh | | LCR[7:0] = BFh | |
| | Read | Write | Read | Write | Read | Write |
| 0h | RHR | THR | DLL | DLL | DLL | DLL |
| 4h | IER(1) | IER(1) | DLH | DLH | DLH | DLH |
| 8h | IIR | FCR(2) | IIR | FCR(2) | EFR | EFR |
| Ch | LCR | LCR | LCR | LCR | LCR | LCR |
| 10h | MCR(2) | MCR(2) | MCR(2) | MCR(2) | XON1/ADDR1 | XON1/ADDR1 |
| 14h | LSR | - | LSR | - | XON2/ADDR2 | XON2/ADDR2 |
| 18h | MSR/TCR(3) | TCR(3) | MSR/TCR(3) | TCR(3) | XOFF1/TCR(3) | XOFF1/TCR(3) |
| 1Ch | SPR/TLR(3) | SPR/TLR(3) | SPR/TLR(3) | SPR/TLR(3) | XOFF2/TLR(3) | XOFF2/TLR(3) |
| 20h | MDR1 | MDR1 | MDR1 | MDR1 | MDR1 | MDR1 |
| 24h | MDR2 | MDR2 | MDR2 | MDR2 | MDR2 | MDR2 |
| 28h | SFLSR | TXFLL | SFLSR | TXFLL | SFLSR | TXFLL |
| 2Ch | RESUME | TXFLH | RESUME | TXFLH | RESUME | TXFLH |
| 30h | SFREGL | RXFLL | SFREGL | RXFLL | SFREGL | RXFLL |
| 34h | SFREGH | RXFLH | SFREGH | RXFLH | SFREGH | RXFLH |
| 38h | BLR | BLR | UASR | - | UASR | - |
| 3Ch | ACREG | ACREG | - | - | - | - |
| 40h | SCR | SCR | SCR | SCR | SCR | SCR |
| 44h | SSR | SSR[2] | SSR | SSR[2] | SSR | SSR[2] |
| 48h | EBLR | EBLR | - | - | - | - |
| 50h | MVR | - | MVR | - | MVR | - |
| 54h | SYSC | SYSC | SYSC | SYSC | SYSC | SYSC |
| 58h | SYSS | - | SYSS | - | SYSS | - |
| 5Ch | WER | WER | WER | WER | WER | WER |
| 60h | CFPS | CFPS | CFPS | CFPS | CFPS | CFPS |
| 80h | MDR3 | MDR3 | MDR3 | MDR3 | MDR3 | MDR3 |

(1) In UART modes, IER[7:4] can only be written when EFR[4] = 1; in IrDA/CIR modes, EFR[4] has no impact on the access to IER[7:4].
(2) MCR[7:5] and FCR[5:4] can only be written when EFR[4] = 1.
(3) Transmission control register (TCR) and trigger level register (TLR) are accessible only when EFR[4] = 1 and MCR[6] = 1.


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.1 Receiver Holding Register (RHR)

The receiver section consists of the receiver holding register and the receiver shift register. The RHR is actually a 64-byte FIFO. The receiver shift register receives serial data from RX input. The data is converted to parallel data and moved to the RHR. If the FIFO is disabled, location zero of the FIFO is used to store the single data character.

**NOTE:** If an overflow occurs, the data in the RHR is not overwritten.

The receiver holding register (RHR) register is shown in Figure 24-28 and described in Table 24-10.

**Figure 24-28. Receiver Holding Register (RHR)**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RHR | |
| R-0 | | R-unknown | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-10. Receiver Holding Register (RHR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | RHR | 0-FFh | Receive holding register. |

### 24.3.2 Transmit Holding Register (THR)

The transmitter section consists of the transmit holding register and the transmit shift register. The transmit holding register is a 64-byte FIFO. The MPU writes data to the THR. The data is placed in the transmit shift register where it is shifted out serially on the TX output. If the FIFO is disabled, location zero of the FIFO is used to store the data.

The transmit holding register (THR) is shown in Figure 24-29 and described in Table 24-11.

**Figure 24-29. Transmit Holding Register (THR)**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | THR | |
| R-0 | | W-unknown | |

LEGEND: R/W = Read/Write; W = Write only; -*n* = value after reset

**Table 24-11. Transmit Holding Register (THR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | THR | 0-FFh | Transmit holding register. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.3 Interrupt Enable Register (IER) - UART Mode

The interrupt enable register (IER) can be programmed to enable/disable any interrupt. There are seven types of interrupt in this mode: receiver error, RHR interrupt, THR interrupt, XOFF received and $\overline{CTS}$/$\overline{RTS}$ change of state from low to high. Each interrupt can be enabled/disabled individually. There is also a sleep mode enable bit. The UART interrupt enable register (IER) is shown in Figure 24-30 and described in Table 24-12.

**Figure 24-30. UART Interrupt Enable Register (IER)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CTSIT | RTSIT | XOFFIT | SLEEPMODE | MODEMSTSIT | LINESTSIT | THRIT | RHRIT |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-12. UART Interrupt Enable Register (IER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | CTSIT | | Can be written only when EFR[4] = 1. |
| | | 0 | Disables the $\overline{CTS}$ interrupt. |
| | | 1 | Enables the $\overline{CTS}$ interrupt. |
| 6 | RTSIT | | Can be written only when EFR[4] = 1. |
| | | 0 | Disables the $\overline{RTS}$ interrupt. |
| | | 1 | Enables the $\overline{RTS}$ interrupt. |
| 5 | XOFFIT | | Can be written only when EFR[4] = 1. |
| | | 0 | Disables the XOFF interrupt. |
| | | 1 | Enables the XOFF interrupt. |
| 4 | SLEEPMODE | | Can be only written when EFR[4] = 1. |
| | | 0 | Disables sleep mode. |
| | | 1 | Enables sleep mode (stop baud rate clock when the module is inactive). |
| 3 | MODEMSTSIT | 0 | Disables the modem status register interrupt. |
| | | 1 | Enables the modem status register interrupt |
| 2 | LINESTSIT | 0 | Disables the receiver line status interrupt. |
| | | 1 | Enables the receiver line status interrupt. |
| 1 | THRIT | 0 | Disables the THR interrupt. |
| | | 1 | Enables the THR interrupt. |
| 0 | RHRIT | 0 | Disables the RHR interrupt and time out interrupt. |
| | | 1 | Enables the RHR interrupt and time out interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.4 Interrupt Enable Register (IER) - IrDA Mode

The IrDA interrupt enable register (IER) can be programmed to enable/disable any interrupt. There are 8 types of interrupt in these modes, received EOF, LSR interrupt, TX status, status FIFO interrupt, RX overrun, last byte in RX FIFO, THR interrupt, and RHR interrupt. Each interrupt can be enabled/disabled individually. The IrDA interrupt enable register (IER) is shown in Figure 24-31 and described in Table 24-13.

**NOTE:** The TXSTATUSIT interrupt reflects two possible conditions. The MDR2[0] bit should be read to determine the status in the event of this interrupt.

**Figure 24-31. IrDA Interrupt Enable Register (IER)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EOFIT | LINESTSIT | TXSTATUSIT | STSFIFOTRIGIT | RXOVERRUNIT | LASTRXBYTEIT | THRIT | RHRIT |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 24-13. IrDA Interrupt Enable Register (IER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | EOFIT | 0 | Disables the received EOF interrupt. |
| | | 1 | Enables the received EOF interrupt. |
| 6 | LINESTSIT | 0 | Disables the receiver line status interrupt. |
| | | 1 | Enables the receiver line status interrupt. |
| 5 | TXSTATUSIT | 0 | Disables the TX status interrupt. |
| | | 1 | Enables the TX status interrupt. |
| 4 | STSFIFOTRIGIT | 0 | Disables status FIFO trigger level interrupt. |
| | | 1 | Enables status FIFO trigger level interrupt. |
| 3 | RXOVERRUNIT | 0 | Disables the RX overrun interrupt. |
| | | 1 | Enables the RX overrun interrupt. |
| 2 | LASTRXBYTEIT | 0 | Disables the last byte of frame in RX FIFO interrupt. |
| | | 1 | Enables the last byte of frame in RX FIFO interrupt. |
| 1 | THRIT | 0 | Disables the THR interrupt. |
| | | 1 | Enables the THR interrupt. |
| 0 | RHRIT | 0 | Disables the RHR interrupt. |
| | | 1 | Enables the RHR interrupt. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.5 Interrupt Enable Register (IER) - CIR Mode

The CIR interrupt enable register (IER) can be programmed to enable/disable any interrupt. There are 5 types of interrupt in these modes, TX status, RX overrun, RX stop interrupt, THR interrupt, and RHR interrupt. Each interrupt can be enabled/disabled individually. The CIR interrupt enable register (IER) is shown in Figure 24-32 and described in Table 24-14.

**NOTE:** In CIR mode, the TXSTATUSIT bit has only one meaning corresponding to the case MDR2[0] = 0.

The RXSTOPIT interrupt is generated based on the value set in the BOF Length register (EBLR).

**Figure 24-32. CIR Interrupt Enable Register (IER)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | TXSTATUSIT | Reserved | RXOVERRUNIT | RXSTOPIT | THRIT | RHRIT |
| R-0 | | R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-14. CIR Interrupt Enable Register (IER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-6 | Reserved | 0 | Reserved. |
| 5 | TXSTATUSIT | 0 | Disables the TX status interrupt. |
| | | 1 | Enables the TX status interrupt. |
| 4 | Reserved | 0 | Reserved. |
| 3 | RXOVERRUNIT | 0 | Disables the RX overrun interrupt. |
| | | 1 | Enables the RX overrun interrupt. |
| 2 | RXSTOPIT | 0 | Disables the RX stop interrupt. |
| | | 1 | Disables the RX stop interrupt. |
| 1 | THRIT | 0 | Disables the THR interrupt. |
| | | 1 | Enables the THR interrupt. |
| 0 | RHRIT | 0 | Disables the RHR interrupt. |
| | | 1 | Enables the RHR interrupt. |

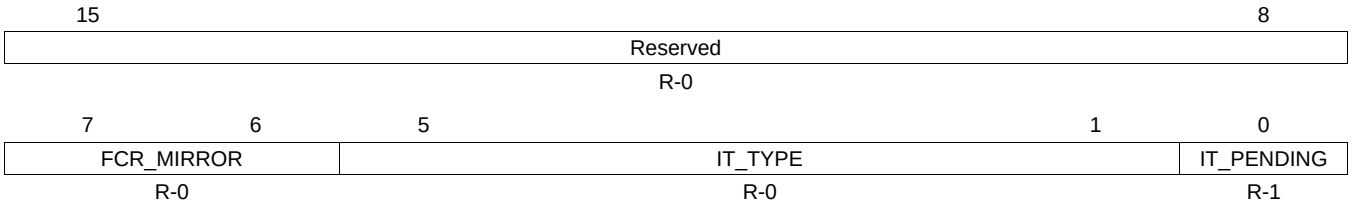

Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.6 Interrupt Identification Register (IIR) - UART Mode

The UART interrupt identification register (IIR) is a read-only register that provides the source of the interrupt. The UART interrupt identification register (IIR) is shown in Figure 24-33 and described in Table 24-15.

**NOTE:** An interrupt source can be flagged only if enabled in the IER register.

**Figure 24-33. UART Interrupt Identification Register (IIR)**




LEGEND: R = Read only; -*n* = value after reset

**Table 24-15. UART Interrupt Identification Register (IIR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-6 | FCR_MIRROR | 0-3h | Mirror the contents of FCR[0] on both bits. |
| 5-1 | IT_TYPE | 0-1Fh | Seven possible interrupts in UART mode; other combinations never occur: |
| | | 0 | Modem interrupt. Priority = 4. |
| | | 1h | THR interrupt. Priority = 3. |
| | | 2h | RHR interrupt. Priority = 2. |
| | | 3h | Receiver line status error. Priority = 1. |
| | | 4h-5h | Reserved |
| | | 6h | Rx timeout. Priority = 2. |
| | | 7h | Reserved |
| | | 8h | Xoff/special character. Priority = 5. |
| | | 9h-Fh | Reserved |
| | | 10h | $\overline{CTS}$, $\overline{RTS}$, $\overline{DSR}$ change state from active (low) to inactive (high). Priority = 6. |
| | | 11h-1Fh | Reserved |
| 0 | IT_PENDING | | Interrupt pending. |
| | | 0 | An interrupt is pending. |
| | | 1 | No interrupt is pending. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.7 Interrupt Identification Register (IIR) - IrDA Mode

The IrDA interrupt identification register (IIR) is a read-only register that provides the source of the interrupt. The IrDA interrupt identification register (IIR) is shown in Figure 24-34 and described in Table 24-16.

**NOTE:** An interrupt source can be flagged only if enabled in the IER register.

**Figure 24-34. IrDA Interrupt Identification Register (IIR)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EOF_IT | LINE_STS_IT | TX_STATUS_IT | STS_FIFO_IT | RX_OE_IT | RX_FIFO_LAST_BYTE_IT | THR_IT | RHR_IT |
| R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-16. IrDA Interrupt Identification Register (IIR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | EOF_IT | 0 | Received EOF interrupt inactive. |
| | | 1 | Received EOF interrupt active. |
| 6 | LINE_STS_IT | 0 | Receiver line status interrupt inactive. |
| | | 1 | Receiver line status interrupt active. |
| 5 | TX_STATUS_IT | 0 | TX status interrupt inactive. |
| | | 1 | TX status interrupt active. |
| 4 | STS_FIFO_IT | 0 | Status FIFO trigger level interrupt inactive. |
| | | 1 | Status FIFO trigger level interrupt active. |
| 3 | RX_OE_IT | 0 | RX overrun interrupt inactive. |
| | | 1 | RX overrun interrupt active. |
| 2 | RX_FIFO_LAST_BYTE_IT | 0 | Last byte of frame in RX FIFO interrupt inactive. |
| | | 1 | Last byte of frame in RX FIFO interrupt active. |
| 1 | THR_IT | 0 | THR interrupt inactive. |
| | | 1 | THR interrupt active. |
| 0 | RHR_IT | 0 | RHR interrupt inactive. |
| | | 1 | RHR interrupt active. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.8 Interrupt Identification Register (IIR) - CIR Mode

The CIR interrupt identification register (IIR) is a read-only register that provides the source of the interrupt. The CIR interrupt identification register (IIR) is shown in Figure 24-35 and described in Table 24-17.

**NOTE:** An interrupt source can be flagged only if enabled in the IER register.

**Figure 24-35. CIR Interrupt Identification Register (IIR)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | TXSTATUSIT | Reserved | RXOEIT | RXSTOPIT | THRIT | RHRIT |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-17. CIR Interrupt Identification Register (IIR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-6 | Reserved | 0 | Reserved. |
| 5 | TXSTATUSIT | 0 | TX status interrupt inactive |
| | | 1 | TX status interrupt active |
| 4 | Reserved | 0 | Reserved. |
| 3 | RXOEIT | 0 | RX overrun interrupt inactive |
| | | 1 | RX overrun interrupt active |
| 2 | RXSTOPIT | 0 | Receive stop interrupt is inactive |
| | | 1 | Receive stop interrupt is active |
| 1 | THRIT | 0 | THR interrupt inactive |
| | | 1 | THR interrupt active |
| 0 | RHRIT | 0 | RHR interrupt inactive |
| | | 1 | RHR interrupt active |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.9 FIFO Control Register (FCR)

The FIFO Control Register (FCR) is shown in Figure 24-36 and described in Table 24-18.

**Figure 24-36. FIFO Control Register (FCR)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RX_FIFO_TRIG | | TX_FIFO_TRIG | | DMA_MODE | TX_FIFO_CLEAR | RX_FIFO_CLEAR | FIFO_EN |
| W-0 | | W-0 | | W-0 | W-0 | W-0 | W-0 |

LEGEND: R = Read only; W = Write only; -*n* = value after reset

**Table 24-18. FIFO Control Register (FCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-6 | RX_FIFO_TRIG | 0-3h | Sets the trigger level for the RX FIFO:<br>If SCR[7] = 0 and TLR[7:4] ≠ 0000, RX_FIFO_TRIG is not considered.<br>If SCR[7] = 1, RX_FIFO_TRIG is 2 LSB of the trigger level (1 to 63 on 6 bits) with the granularity 1.<br>If SCR[7] = 0 and TLR[7:4] = 0000: |
| | | 0 | 8 characters |
| | | 1h | 16 characters |
| | | 2h | 56 characters |
| | | 3h | 60 characters |
| 5-4 | TX_FIFO_TRIG | 0-3h | Can be written only if EFR[4] = 1.<br>Sets the trigger level for the TX FIFO:<br>If SCR[6] = 0 and TLR[3:0] ≠ 0000, TX_FIFO_TRIG is not considered.<br>If SCR[6] = 1, TX_FIFO_TRIG is 2 LSB of the trigger level (1 to 63 on 6 bits) with a granularity of 1.<br>If SCR[6] = 0 and TLR[3:0] = 0000: |
| | | 0 | 8 characters |
| | | 1h | 16 characters |
| | | 2h | 32 characters |
| | | 3h | 56 characters |
| 3 | DMA_MODE | | Can be changed only when the baud clock is not running (DLL and DLH cleared to 0).<br>If SCR[0] = 0, this register is considered. |
| | | 0 | DMA_MODE 0 (No DMA). |
| | | 1 | DMA_MODE 1 (UART_NDMA_REQ[0] in TX, UART_NDMA_REQ[1] in RX). |
| 2 | TX_FIFO_CLEAR | 0 | No change. |
| | | 1 | Clears the transmit FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO. |
| 1 | RX_FIFO_CLEAR | 0 | No change. |
| | | 1 | Clears the receive FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO. |
| 0 | FIFO_EN | | Can be changed only when the baud clock is not running (DLL and DLH cleared to 0). |
| | | 0 | Disables the transmit and receive FIFOs. The transmit and receive holding registers are 1-byte FIFOs. |
| | | 1 | Enables the transmit and receive FIFOs. The transmit and receive holding registers are 64-byte FIFOs. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.10 Line Control Register (LCR)

The line control register (LCR) is shown in Figure 24-37 and described in Table 24-19.

**NOTE:** As soon as LCR[6] is set to 1, the TX line is forced to 0 and remains in this state as long as LCR[6] = 1.

**Figure 24-37. Line Control Register (LCR)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIV_EN | BREAK_EN | PARITY_TYPE2 | PARITY_TYPE1 | PARITY_EN | NB_STOP | CHAR_LENGTH | |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-19. Line Control Register (LCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | DIV_EN | | Divisor latch enable. |
| | | 0 | Normal operating condition. |
| | | 1 | Divisor latch enable. Allows access to DLL and DLH. |
| 6 | BREAK_EN | | Break control bit. |
| | | | **Note:** When LCR[6] is set to 1, the TX line is forced to 0 and remains in this state as long as LCR[6] = 1. |
| | | 0 | Normal operating condition. |
| | | 1 | Forces the transmitter output to go low to alert the communication terminal. |
| 5 | PARITY_TYPE2 | | If LCR[3] = 1: |
| | | 0 | If LCR[5] = 0, LCR[4] selects the forced parity format. |
| | | 1 | If LCR[5] = 1 and LCR[4] = 0, the parity bit is forced to 1 in the transmitted and received data. |
| | | 1 | If LCR[5] = 1 and LCR[4] = 1, the parity bit is forced to 0 in the transmitted and received data. |
| 4 | PARITY_TYPE1 | | If LCR[3] = 1: |
| | | 0 | Odd parity is generated. |
| | | 1 | Even parity is generated. |
| 3 | PARITY_EN | | Parity bit. |
| | | 0 | No parity. |
| | | 1 | A parity bit is generated during transmission, and the receiver checks for received parity. |
| 2 | NB_STOP | | Specifies the number of stop bits. |
| | | 0 | 1 stop bit (word length = 5, 6, 7, 8). |
| | | 1 | 1.5 stop bits (word length = 5) or 2 stop bits (word length = 6, 7, 8). |
| 1-0 | CHAR_LENGTH | 0-3h | Specifies the word length to be transmitted or received. |
| | | 0 | 5 bits |
| | | 1h | 6 bits |
| | | 2h | 7 bits |
| | | 3h | 8 bit |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.11 Modem Control Register (MCR)

Bits 3-0 control the interface with the modem, data set, or peripheral device that is emulating the modem. The modem control register (MCR) is shown in Figure 24-38 and described in Table 24-20.

**Figure 24-38. Modem Control Register (MCR)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | TCRTLR | XONEN | LOOPBACKEN | CDSTSCH | RISTSCH | RTS | DTR |
| R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 24-20. Modem Control Register (MCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-7 | Reserved | 0 | Reserved. |
| 6 | TCRTLR | | Can be written only when EFR[4] = 1. |
| | | 0 | No action. |
| | | 1 | Enables access to the TCR and TLR registers. |
| 5 | XONEN | | Can be written only when EFR[4] = 1. |
| | | 0 | Disable XON any function. |
| | | 1 | Enable XON any function. |
| 4 | LOOPBACKEN | | Loopback mode enable. |
| | | 0 | Normal operating mode. |
| | | 1 | Enable local loopback mode (internal). In this mode, the MCR[3:0] signals are looped back into MSR[7:4]. The transmit output is looped back to the receive input internally. |
| 3 | CDSTSCH | 0 | In loopback mode, forces $\overline{\text{DCD}}$ input high and IRQ outputs to INACTIVE state. |
| | | 1 | In loopback mode, forces $\overline{\text{DCD}}$ input low and IRQ outputs to INACTIVE state. |
| 2 | RISTSCH | 0 | In loopback mode, forces $\overline{\text{RI}}$ input inactive (high). |
| | | 1 | In loopback mode, forces $\overline{\text{RI}}$ input active (low). |
| 1 | RTS | | In loopback mode, controls MSR[4]. If auto-RTS is enabled, the $\overline{\text{RTS}}$ output is controlled by hardware flow control. |
| | | 0 | Force $\overline{\text{RTS}}$ output to inactive (high). |
| | | 1 | Force $\overline{\text{RTS}}$ output to active (low). |
| 0 | DTR | 0 | Force $\overline{\text{DTR}}$ output (used in loopback mode) to inactive (high). |
| | | 1 | Force $\overline{\text{DTR}}$ output (used in loopback mode) to active (low). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.12 Line Status Register (LSR) - UART Mode

When the UART line status register (LSR) is read, LSR[4:2] reflect the error bits (BI, FE, PE) of the character at the top of the RX FIFO (next character to be read). Therefore, reading the LSR and then reading the RHR identifies errors in a character. Reading RHR updates BI, FE, and PE. LSR [7] is set when there is an error anywhere in the RX FIFO and is cleared only when there are no more errors remaining in the RX FIFO. The UART line status register (LSR) is shown in Figure 24-39 and described in Table 24-21.

**NOTE:** Reading the LSR does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the RHR. Reading LSR clears OE if set.

**Figure 24-39. UART Line Status Register (LSR)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXFIFOSTS | TXSRE | TXFIFOE | RXBI | RXFE | RXPE | RXOE | RXFIFOE |
| R-0 | R-1 | R-1 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-21. UART Line Status Register (LSR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | RXFIFOSTS | 0 | Normal operation. |
| | | 1 | At least one parity error, framing error, or break indication in the RX FIFO. Bit 7 is cleared when no errors are present in the RX FIFO. |
| 6 | TXSRE | 0 | Transmitter hold (TX FIFO) and shift registers are not empty. |
| | | 1 | Transmitter hold (TX FIFO) and shift registers are empty. |
| 5 | TXFIFOE | 0 | Transmit hold register (TX FIFO) is not empty. |
| | | 1 | Transmit hold register (TX FIFO) is empty. The transmission is not necessarily completed. |
| 4 | RXBI | 0 | No break condition. |
| | | 1 | A break was detected while the data being read from the RX FIFO was being received (RX input was low for one character + 1 bit time frame). |
| 3 | RXFE | 0 | No framing error in data being read from RX FIFO. |
| | | 1 | Framing error occurred in data being read from RX FIFO (received data did not have a valid stop bit). |
| 2 | RXPE | 0 | No parity error in data being read from RX FIFO. |
| | | 1 | Parity error in data being read from RX FIFO. |
| 1 | RXOE | 0 | No overrun error. |
| | | 1 | Overrun error occurred. Set when the character held in the receive shift register is not transferred to the RX FIFO. This case occurs only when receive FIFO is full. |
| 0 | RXFIFOE | 0 | No data in the receive FIFO. |
| | | 1 | At least one data character in the RX FIFO. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.13 Line Status Register (LSR) - IrDA Mode

When the IrDA line status register (LSR) is read, LSR[4:2] reflect the error bits (FL, CRC, ABORT) of the frame at the top of the status FIFO (next frame status to be read). The IrDA line status register (LSR) is shown in Figure 24-40 and described in Table 24-22.

**Figure 24-40. IrDA Line Status Register (LSR)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| THR_EMPTY | STS_FIFO_FULL | RX_LAST_BYTE | FRAME_TOO_LONG | ABORT | CRC | STS_FIFO_E | RX_FIFO_E |
| R-1 | R-0 | R-0 | R-0 | R-0 | R-0 | R-1 | R-1 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 24-22. IrDA Line Status Register (LSR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | THR_EMPTY | 0 | Transmit holding register (TX FIFO) is not empty. |
| | | 1 | Transmit hold register (TX FIFO) is empty. The transmission is not necessarily completed. |
| 6 | STS_FIFO_FULL | 0 | Status FIFO is not full. |
| | | 1 | Status FIFO is full. |
| 5 | RX_LAST_BYTE | 0 | The RX FIFO (RHR) does not contain the last byte of the frame to be read. |
| | | 1 | The RX FIFO (RHR) contains the last byte of the frame to be read. This bit is set to 1 only when the last byte of a frame is available to be read. It is used to determine the frame boundary. It is cleared on a single read of the LSR register. |
| 4 | FRAME_TOO_LONG | 0 | No frame-too-long error in frame. |
| | | 1 | Frame-too-long error in the frame at the top of the status FIFO (next character to be read). This bit is set to 1 when a frame exceeding the maximum length (set by RXFLH and RXFLL registers) is received. When this error is detected, current frame reception is terminated. Reception is stopped until the next START flag is detected. |
| 3 | ABORT | 0 | No abort pattern error in frame. |
| | | 1 | Abort pattern received. SIR and MIR: abort pattern. FIR: Illegal symbol. |
| 2 | CRC | 0 | No CRC error in frame. |
| | | 1 | CRC error in the frame at the top of the status FIFO (next character to be read). |
| 1 | STS_FIFO_E | 0 | Status FIFO is not empty. |
| | | 1 | Status FIFO is empty. |
| 0 | RX_FIFO_E | 0 | At least one data character in the RX FIFO. |
| | | 1 | No data in the receive FIFO. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.14 Line Status Register (LSR) - CIR Mode

The CIR line status register (LSR) is shown in Figure 24-41 and described in Table 24-23.

**Figure 24-41. CIR Line Status Register (LSR)**

| 15 | | | | | 8 |
|---|---|---|---|---|---|
| Reserved | | | | | |
| R-0 | | | | | |

| 7 | 6 | 5 | 4 | 1 | 0 |
|---|---|---|---|---|---|
| THREMPTY | Reserved | RXSTOP | Reserved | | RXFIFOE |
| R-1 | R-0 | R-0 | R-0 | | R-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-23. CIR Line Status Register (LSR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | THREMPTY | 0 | Transmit holding register (TX FIFO) is not empty. |
| | | 1 | Transmit hold register (TX FIFO) is empty. The transmission is not necessarily completed. |
| 6 | Reserved | 0 | Reserved. |
| 5 | RXSTOP | | The RXSTOP is generated based on the value set in the BOF Length register (EBLR). |
| | | 0 | Reception is on going or waiting for a new frame. |
| | | 1 | Reception is completed. It is cleared on a single read of the LSR register. |
| 4-1 | Reserved | 0 | Reserved |
| 0 | RXFIFOE | 0 | At least one data character in the RX FIFO. |
| | | 1 | No data in the receive FIFO. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.15 Modem Status Register (MSR)

The modem status register (MSR) provides information about the current state of the control lines from the modem, data set, or peripheral device to the Local Host. It also indicates when a control input from the modem changes state. The modem status register (MSR) is shown in Figure 24-42 and described in Table 24-24.

**Figure 24-42. Modem Status Register (MSR)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NCD_STS | NRI_STS | NDSR_STS | NCTS_STS | DCD_STS | RI_STS | DSR_STS | CTS_STS |
| R-unknown | R-unknown | R-unknown | R-unknown | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-24. Modem Status Register (MSR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | NCD_STS | | This bit is the complement of the $\overline{DCD}$ input. In loopback mode, it is equivalent to MCR[3]. |
| 6 | NRI_STS | | This bit is the complement of the $\overline{RI}$ input. In loopback mode, it is equivalent to MCR[2]. |
| 5 | NDSR_STS | | This bit is the complement of the $\overline{DSR}$ input. In loopback mode, it is equivalent to MCR[0]. |
| 4 | NCTS_STS | | This bit is the complement of the $\overline{CTS}$ input. In loopback mode, it is equivalent to MCR[1]. |
| 3 | DCD_STS | 0 | No change. |
| | | 1 | Indicates that $\overline{DCD}$ input (or MCR[3] in loopback mode) has changed. Cleared on a read. |
| 2 | RI_STS | 0 | No change. |
| | | 1 | Indicates that $\overline{RI}$ input (or MCR[2] in loopback mode) changed state from low to high. Cleared on a read. |
| 1 | DSR_STS | 0 | No change. |
| | | 1 | Indicates that $\overline{DSR}$ input (or MCR[0] in loopback mode) changed state. Cleared on a read. |
| 0 | CTS_STS | 0 | No change. |
| | | 1 | Indicates that $\overline{CTS}$ input (or MCR[1] in loopback mode) changed state. Cleared on a read. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.16 Transmission Control Register (TCR)

The transmission control register (TCR) stores the receive FIFO threshold levels to start/stop transmission during hardware flow control. The transmission control register (TCR) is shown in Figure 24-43 and described in Table 24-25.

**NOTE:**

- Trigger levels from 0-60 bytes are available with a granularity of 4.
- Trigger level = 4 × [4-bit register value]
- You must ensure that TCR[3:0] > TCR[7:4], whenever auto-RTS or software flow control is enabled to avoid a misoperation of the device. In FIFO interrupt mode with flow control, you have to also ensure that the trigger level to HALT transmission is greater or equal to receive FIFO trigger level (either TLR[7:4] or FCR[7:6]); otherwise, FIFO operation stalls.
- In FIFO DMA mode with flow control, this concept does not exist because the DMA request is sent each time a byte is received.

**Figure 24-43. Transmission Control Register (TCR)**

| 15 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | RXFIFOTRIGSTART | | RXFIFOTRIGHALT | |
| R-0 | | R/W-0 | | R/W-Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-25. Transmission Control Register (TCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-4 | RXFIFOTRIGSTART | 0-Fh | RX FIFO trigger level to RESTORE transmission (0 to 60). |
| 3-0 | RXFIFOTRIGHALT | 0-Fh | RX FIFO trigger level to HALT transmission (0 to 60). |

### 24.3.17 Scratchpad Register (SPR)

The scratchpad register (SPR) is a read/write register that does not control the module. It is a scratchpad register used to hold temporary data. The scratchpad register (SPR) is shown in Figure 24-44 and described in Table 24-26.

**Figure 24-44. Scratchpad Register (SPR)**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | SPR_WORD | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-26. Scratchpad Register (SPR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | SPR_WORD | 0-FFh | Scratchpad register. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.18 Trigger Level Register (TLR)

This register stores the programmable transmit and receive FIFO trigger levels used for DMA and IRQ generation. The trigger level register (TLR) is shown in Figure 24-45 and described in Table 24-27.

**Figure 24-45. Trigger Level Register (TLR)**

| 15 8 | 7 4 | 3 0 |
|---|---|---|
| Reserved | RX_FIFO_TRIG_ DMA | TX_FIFO_TRIG_ DMA |
| R-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-27. Trigger Level Register (TLR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-4 | RX_FIFO_TRIG_ DMA | 0-Fh | Receive FIFO trigger level. See Table 24-28. |
| 3-0 | TX_FIFO_TRIG_ DMA | 0-Fh | Transmit FIFO trigger level. See Table 24-29. |

**Table 24-28. RX FIFO Trigger Level Setting Summary**

| SCR[7] | TLR[7:4] | Description |
|---|---|---|
| 0 | 0 | Defined by FCR[7:6] (either 8, 16, 56, 60 characters). |
| 0 | ≠ 0000 | Defined by TLR[7:4] (from 4 to 60 characters with a granularity of 4 characters). |
| 1 | any value | Defined by the concatenated value of TLR[7:4] and FCR[7:6] (from 1 to 63 characters with a granularity of 1 character). **Note:** the combination of TLR[7:4] = 0000 and FCR[7:6] = 00 (all zeros) is not supported (minimum of 1 character is required). All zeros results in unpredictable behavior. |

**Table 24-29. TX FIFO Trigger Level Setting Summary**

| SCR[6] | TLR[3:0] | Description |
|---|---|---|
| 0 | 0 | Defined by FCR[5:4] (either 8, 16, 32, 56 characters). |
| 0 | ≠ 0000 | Defined by TLR[3:0] (from 4 to 60 characters with a granularity of 4 characters). |
| 1 | any value | Defined by the concatenated value of TLR[3:0] and FCR[5:4] (from 1 to 63 characters with a granularity of 1 character). **Note:** the combination of TLR[3:0] = 0000 and FCR[5:4] = 00 (all zeros) is not supported (minimum of 1 character is required). All zeros results in unpredictable behavior. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.19 Mode Definition Register 1 (MDR1)

The mode of operation is programmed by writing to MDR1[2:0]; therefore, the mode definition register 1 (MDR1) must be programmed on startup after configuration of the configuration registers (DLL, DLH, and LCR). The value of MDR1[2:0] must not be changed again during normal operation. The mode definition register 1 (MDR1) is shown in Figure 24-46 and described in Table 24-30.

**NOTE:** If the module is disabled by setting the MODESELECT field to 7h, interrupt requests can still be generated unless disabled through the interrupt enable register (IER). In this case, UART mode interrupts are visible. Reading the interrupt identification register (IIR) shows the UART mode interrupt flags.

**Figure 24-46. Mode Definition Register 1 (MDR1)**

| 15 | | | | | | 8 |
|---|---|---|---|---|---|---|
| Reserved | | | | | | |
| R-0 | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 0 |
| FRAMEENDMODE | SIPMODE | SCT | SETTXIR | IRSLEEP | MODESELECT | |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-7h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-30. Mode Definition Register 1 (MDR1) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | FRAMEENDMODE | | IrDA mode only. |
| | | 0 | Frame-length method. |
| | | 1 | Set EOT bit method. |
| 6 | SIPMODE | | MIR/FIR modes only. |
| | | 0 | Manual SIP mode: SIP is generated with the control of ACREG[3]. |
| | | 1 | Automatic SIP mode: SIP is generated after each transmission. |
| 5 | SCT | | Store and control the transmission. |
| | | 0 | Starts the infrared transmission when a value is written to the THR register. |
| | | 1 | Starts the infrared transmission with the control of ACREG[2]. **Note:** Before starting any transmission, there must be no reception ongoing. |
| 4 | SETTXIR | | Used to configure the infrared transceiver. |
| | | 0 | If MDR2[7] = 0, no action; if MDR2[7] = 1, TXIR pin output is forced low. |
| | | 1 | TXIR pin output is forced high (not dependant of MDR2[7] value). |
| 3 | IRSLEEP | | IrDA/CIR sleep mode. |
| | | 0 | IrDA/CIR sleep mode disabled. |
| | | 1 | IrDA/CIR sleep mode enabled. |
| 2-0 | MODESELECT | 0-7h | UART/IrDA/CIR mode selection. |
| | | 0 | UART 16× mode. |
| | | 1 | SIR mode. |
| | | 2h | UART 16× auto-baud. |
| | | 3h | UART 13× mode. |
| | | 4h | MIR mode. |
| | | 5h | FIR mode. |
| | | 6h | CIR mode. |
| | | 7h | Disable (default state). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.20 Mode Definition Register 2 (MDR2)

The MDR2[0] bit describes the status of the TX status interrupt in IIR[5]. The IRTXUNDERRUN bit must be read after a TX status interrupt occurs. The MDR2[2:1] bits set the trigger level for the frame status FIFO (8 entries) and must be programmed before the mode is programmed in MDR1[2:0]. The mode definition register 2 (MDR2) is shown in Figure 24-47 and described in Table 24-31.

**NOTE:** The MDR2[6] bit gives the flexibility to invert the RX pin inside the UART module to ensure that the protocol at the input of the transceiver module has the same polarity at module level. By default, the RX pin is inverted because most of transceiver invert the IR receive pin.

**Figure 24-47. Mode Definition Register 2 (MDR2)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SETTXIRALT | IRRXINVERT | CIRPULSEMODE | | UARTPULSE | STSFIFOTRIG | | IRTXUNDERRUN |
| R/W-0 | R/W-0 | R/W-0 | | R/W-0 | R/W-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 24-31. Mode Definition Register 2 (MDR2) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | SETTXIRALT | | Provides alternate functionality for MDR1[4]. |
| | | 0 | Normal mode |
| | | 1 | Alternate mode for SETTXIR |
| 6 | IRRXINVERT | | Only for IR mode (IrDA and CIR). Invert RX pin in the module before the voting or sampling system logic of the infrared block. This does not affect the RX path in UART modem modes. |
| | | 0 | Inversion is performed. |
| | | 1 | No inversion is performed. |
| 5-4 | CIRPULSEMODE | 0-3h | CIR pulse modulation definition. Defines high level of the pulse width associated with a digit: |
| | | 0 | Pulse width of 3 from 12 cycles. |
| | | 1h | Pulse width of 4 from 12 cycles. |
| | | 2h | Pulse width of 5 from 12 cycles. |
| | | 3h | Pulse width of 6 from 12 cycles. |
| 3 | UARTPULSE | | UART mode only. Used to allow pulse shaping in UART mode. |
| | | 0 | Normal UART mode. |
| | | 1 | UART mode with pulse shaping. |
| 2-1 | STSFIFOTRIG | 0-3h | Only for IrDA mode. Frame status FIFO threshold select: |
| | | 0 | 1 entry |
| | | 1h | 4 entries |
| | | 2h | 7 entries |
| | | 3h | 8 entries |
| 0 | IRTXUNDERRUN | | IrDA transmission status interrupt. When the TX status interrupt (IIR[5]) occurs, the meaning of the interrupt is: |
| | | 0 | The last bit of the frame was transmitted successfully without error. |
| | | 1 | An underrun occurred. The last bit of the frame was transmitted but with an underrun error. The bit is reset to 0 when the RESUME register is read. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.21 Mode Definition Register 3 (MDR3)

The MDR3[0] DISABLE_CIR_RX_DEMOD bit will force the CIR receiver to bypass demodulation of received data if set.

The MDR3[1] NONDEFAULT_FREQ bit allows the user to set sample per bi. Set this bit if non default (48MHz) clock frequency is used to achieve less than 2 % error rate. Changing this bit automatically disables the device by setting MDR1[2:0] to 3h.

MDR3[2] SET_DMA_THRESHOLD bit enables the usage of different TX DMA threshold then 64-trigger, when set to 1h.

The mode definition register 3 (MDR3) is shown in Figure 24-48 and described in Table 24-32.

**Figure 24-48. Mode Definition Register 3 (MDR3)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | SET_DMA_TX_THRESHOLD | NONDEFAULT_FREQ | DISABLE_CIR_RX_DEMOD |
| R-0 | | | | | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-32. Mode Definition Register 3 (MDR3) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 31-3 | Reserved | 0 | Reserved. |
| 2 | SET_DMA_THRESHOLD | 0 | Use the value of 64-tx trigger. |
| | | 1h | Sets different TX DMA threshold then 64-trigger |
| 1 | NONDEFAULT_FREQ | 0 | Default fclk frequencies |
| | | 1h | Nondefault fclk frequencies |
| 0 | DISABLE_CIR_RX_DEMOD | 0 | Enalbes CIR RX demodulation |
| | | 1h | Disables CIR RX demodulation |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.22 Status FIFO Line Status Register (SFLSR)

Reading the status FIFO line status register (SFLSR) effectively reads frame status information from the status FIFO. This register does not physically exist. Reading this register increments the status FIFO read pointer (SFREGL and SFREGH must be read first). The status FIFO line status register (SFLSR) is shown in Figure 24-49 and described in Table 24-33.

**NOTE:** Top of RX FIFO = Next frame to be read from RX FIFO.

**Figure 24-49. Status FIFO Line Status Register (SFLSR)**

| 15 | 8 |
|---|---|
| Reserved | |
| R-0 | |

| 7 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | OE_ERROR | FRAME_TOO_LONG_ERROR | ABORT_DETECT | CRC_ERROR | Reserved |
| R-0 | | R-unknown | R-unknown | R-unknown | R-unknown | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-33. Status FIFO Line Status Register (SFLSR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-5 | Reserved | 0 | Reserved. |
| 4 | OE_ERROR | 0 | No error |
| | | 1 | Overrun error in RX FIFO when frame at top of RX FIFO was received. |
| 3 | FRAME_TOO_LONG_ERROR | 0 | No error |
| | | 1 | Frame-length too long error in frame at top of RX FIFO. |
| 2 | ABORT_DETECT | 0 | No error |
| | | 1 | Abort pattern detected in frame at top of RX FIFO. |
| 1 | CRC_ERROR | 0 | No error |
| | | 1 | CRC error in frame at top of RX FIFO. |
| 0 | Reserved | 0 | Reserved. |

### 24.3.23 RESUME Register

The RESUME register is used to clear internal flags, which halt transmission/reception when an underrun/overrun error occurs. Reading this register resumes the halted operation. This register does not physically exist and always reads as 00. The RESUME register is shown in Figure 24-50 and described in Table 24-34.

**Figure 24-50. RESUME Register**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RESUME | |
| R-0 | | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-34. RESUME Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | RESUME | 0-FFh | Dummy read to restart the TX or RX. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.24 Status FIFO Register Low (SFREGL)

The frame lengths of received frames are written into the status FIFO. This information can be read by reading the status FIFO register low (SFREGL) and the status FIFO register high (SFREGH). These registers do not physically exist. The LSBs are read from SFREGL and the MSBs are read from SFREGH. Reading these registers does not alter the status FIFO read pointer. These registers must be read before the pointer is incremented by reading the SFLSR. The status FIFO register low (SFREGL) is shown in Figure 24-51 and described in Table 24-35.

**Figure 24-51. Status FIFO Register Low (SFREGL)**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | SFREGL | |
| R-0 | | R-unknown | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-35. Status FIFO Register Low (SFREGL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | SFREGL | 0-FFh | LSB part of the frame length. |

### 24.3.25 Status FIFO Register High (SFREGH)

The frame lengths of received frames are written into the status FIFO. This information can be read by reading the status FIFO register low (SFREGL) and the status FIFO register high (SFREGH). These registers do not physically exist. The LSBs are read from SFREGL and the MSBs are read from SFREGH. Reading these registers does not alter the status FIFO read pointer. These registers must be read before the pointer is incremented by reading the SFLSR. The status FIFO register high (SFREGH) is shown in Figure 24-52 and described in Table 24-36.

**Figure 24-52. Status FIFO Register High (SFREGH)**

| 15 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | SFREGH | |
| R-0 | | R-unknown | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-36. Status FIFO Register High (SFREGH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-4 | Reserved | 0 | Reserved. |
| 3-0 | SFREGH | 0-Fh | MSB part of the frame length. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.26 BOF Control Register (BLR)

The BLR[6] bit is used to select whether C0h or FFh start patterns are to be used, when multiple start flags are required in SIR mode. If only one start flag is required, this is always C0h. If *n* start flags are required, either (n – 1) C0h or (n –1) FFh flags are sent, followed by a single C0h flag (immediately preceding the first data byte). The BOF control register (BLR) is shown in Figure 24-53 and described in Table 24-37.

**Figure 24-53. BOF Control Register (BLR)**

| 15 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|
| Reserved | | STSFIFORESET | XBOFTYPE | Reserved | |
| R-0 | | R/W-0 | R/W-1 | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-37. BOF Control Register (BLR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | STSFIFORESET | | Status FIFO reset. This bit is self-clearing.. |
| 6 | XBOFTYPE | | SIR xBOF select. |
| | | 0 | FFh start pattern is used. |
| | | 1 | C0h start pattern is used. |
| 0 | Reserved | 0 | Reserved. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.27 Auxiliary Control Register (ACREG)

The auxiliary control register (ACREG) is shown in Figure 24-54 and described in Table 24-38.

**NOTE:**

- If transmit FIFO is not empty and MDR1[5] = 1, IrDA starts a new transfer with data of previous frame as soon as abort frame has been sent. Therefore, TX FIFO must be reset before sending an abort frame.
- It is recommended to disable TX FIFO underrun capability by masking corresponding underrun interrupt. When disabling underrun by setting ACREG[4] = 1, unknown data is sent over TX line.

**Figure 24-54. Auxiliary Control Register (ACREG)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PULSETYPE | SDMOD | DISIRRX | DISTXUNDERRUN | SENDSIP | SCTXEN | ABORTEN | EOTEN |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-38. Auxiliary Control Register (ACREG) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | PULSETYPE | | SIR pulse-width select: |
| | | 0 | 3/16 of baud-rate pulse width |
| | | 1 | 1.6 μs |
| 6 | SDMOD | | Primary output used to configure transceivers. Connected to the SD/MODE input pin of IrDA transceivers. |
| | | 0 | SD pin is set to high. |
| | | 1 | SD pin is set to low. |
| 5 | DISIRRX | | Disable RX input. |
| | | 0 | Normal operation (RX input automatically disabled during transmit, but enabled outside of transmit operation). |
| | | 1 | Disables RX input (permanent state; independent of transmit). |
| 4 | DISTXUNDERRUN | | Disable TX underrun. |
| | | 0 | Long stop bits cannot be transmitted. TX underrun is enabled. |
| | | 1 | Long stop bits can be transmitted. |
| 3 | SENDSIP | | MIR/FIR modes only. Send serial infrared interaction pulse (SIP). |
| | | | If this bit is set during an MIR/FIR transmission, the SIP is sent at the end of it. This bit is automatically cleared at the end of the SIP transmission. |
| | | 0 | No action. |
| | | 1 | Send SIP pulse. |
| 2 | SCTXEN | | Store and control TX start. When MDR1[5] = 1 and the LH writes 1 to this bit, the TX state-machine starts frame transmission. This bit is self-clearing. |
| 1 | ABORTEN | | Frame abort. The LH can intentionally abort transmission of a frame by writing 1 to this bit. Neither the end flag nor the CRC bits are appended to the frame. |
| 0 | EOTEN | | EOT (end-of-transmission) bit. The LH writes 1 to this bit just before it writes the last byte to the TX FIFO in the set-EOT bit frame-closing method. This bit is automatically cleared when the LH writes to the THR (TX FIFO). |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.28 Supplementary Control Register (SCR)

The supplementary control register (SCR) is shown in Figure 24-55 and described in Table 24-39.

**NOTE:** Bit 4 enables the wake-up interrupt, but this interrupt is not mapped into the IIR register. Therefore, when an interrupt occurs and there is no interrupt pending in the IIR register, the SSR[1] bit must be checked. To clear the wake-up interrupt, bit SCR[4] must be reset to 0.

**Figure 24-55. Supplementary Control Register (SCR)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| RXTRIGGRANU1 | TXTRIGGRANU1 | DSRIT | RXCTSDSRWAKEUPENABLE | TXEMPTYCTLIT | DMAMODE2 | | DMAMODECTL |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-39. Supplementary Control Register (SCR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | RXTRIGGRANU1 | 0 | Disables the granularity of 1 for trigger RX level. |
| | | 1 | Enables the granularity of 1 for trigger RX level. |
| 6 | TXTRIGGRANU1 | 0 | Disables the granularity of 1 for trigger TX level. |
| | | 1 | Enables the granularity of 1 for trigger TX level. |
| 5 | DSRIT | 0 | Disables $\overline{\text{DSR}}$ interrupt. |
| | | 1 | Enables $\overline{\text{DSR}}$ interrupt. |
| 4 | RXCTSDSRWAKEUPENABLE | | RX CTS wake-up enable. |
| | | 0 | Disables the WAKE UP interrupt and clears SSR[1]. |
| | | 1 | Waits for a falling edge of RX, $\overline{\text{CTS}}$, or $\overline{\text{DSR}}$ pins to generate an interrupt. |
| 3 | TXEMPTYCTLIT | 0 | Normal mode for THR interrupt. |
| | | 1 | THR interrupt is generated when TX FIFO and TX shift register are empty. |
| 2-1 | DMAMODE2 | 0-3h | Specifies the DMA mode valid if SCR[0] = 1: |
| | | 0 | DMA mode 0 (no DMA). |
| | | 1h | DMA mode 1 (UARTnDMAREQ[0] in TX, UARTnDMAREQ[1] in RX) |
| | | 2h | DMA mode 2 (UARTnDMAREQ[0] in RX) |
| | | 3h | DMA mode 3 (UARTnDMAREQ[0] in TX) |
| 0 | DMAMODECTL | 0 | The DMAMODE is set with FCR[3]. |
| | | 1 | The DMAMODE is set with SCR[2:1]. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.29 Supplementary Status Register (SSR)

The supplementary status register (SSR) is shown in Figure 24-56 and described in Table 24-40.

**NOTE:** Bit 1 is reset only when SCR[4] is reset to 0.

**Figure 24-56. Supplementary Status Register (SSR)**

| 15 | | | | 8 |
|---|---|---|---|---|
| Reserved | | | | |
| R-0 | | | | |
| **7** | **3** | **2** | **1** | **0** |
| Reserved | | DMACOUNTERRST | RXCTSDSRWAKEUPSTS | TXFIFOFULL |
| R-0 | | R/W-1 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-40. Supplementary Status Register (SSR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-3 | Reserved | 0 | Reserved. |
| 2 | DMACOUNTERRST | 0 | The DMA counter will not be reset, if the corresponding FIFO is reset (via FCR[1] or FCR[2]). |
| | | 1 | The DMA counter will be reset, if the corresponding FIFO is reset (via FCR[1] or FCR[2]). |
| 1 | RXCTSDSRWAKEUPSTS | | Pin falling edge detection: Reset only when SCR[4] is reset to 0. |
| | | 0 | No falling-edge event on RX, $\overline{CTS}$, and $\overline{DSR}$. |
| | | 1 | A falling edge occurred on RX, $\overline{CTS}$, or $\overline{DSR}$. |
| 0 | TXFIFOFULL | 0 | TX FIFO is not full. |
| | | 1 | TX FIFO is full. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.30 BOF Length Register (EBLR)

In IrDA SIR operation, the BOF length register (EBLR) specifies the number of BOF + xBOFs to transmit. The value set into this register must consider the BOF character; therefore, to send only one BOF with no XBOF, this register must be set to 1. To send one BOF with *n* XBOFs, this register must be set to *n* + 1. Furthermore, the value 0 sends 1 BOF plus 255 XBOFs.

In IrDA MIR mode, the BOF length register (EBLR) specifies the number of additional start flags (MIR protocol mandates a minimum of 2 start flags).

In CIR mode, the BOF length register (EBLR) specifies the number of consecutive zeros to be received before generating the RXSTOP interrupt (IIR[2]). All the received zeros are stored in the RX FIFO. When the register is cleared to 0, this feature is deactivated and always in reception state, which is disabled by setting the ACREG[5] bit to 1.

The BOF length register (EBLR) is shown in Figure 24-57 and described in Table 24-41.

**NOTE:** If the RX_STOP interrupt occurs before a byte boundary, the remaining bits of the last byte are filled with zeros and then passed into the RX FIFO.

**Figure 24-57. BOF Length Register (EBLR)**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | EBLR | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-41. BOF Length Register (EBLR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | EBLR | | IrDA mode: This register allows definition of up to 176 xBOFs, the maximum required by IrDA specification.<br>CIR mode: This register specifies the number of consecutive zeros to be received before generating the RXSTOP interrupt (IIR[2]). |
| | | 0 | Feature disabled. |
| | | 1h | Generate RXSTOP interrupt after receiving 1 zero bit. |
| | | ... | |
| | | FFh | Generate RXSTOP interrupt after receiving 255 zero bits. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.31 Module Version Register (MVR)

The reset value is fixed by hardware and corresponds to the RTL revision of this module. A reset has no effect on the value returned. The module version register (MVR) is shown in Figure 24-58 and described in Table 24-42.

**Figure 24-58. Module Version Register (MVR)**

| 15 | 8 | 7 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | MAJORREV | | MINORREV | |
| R-0 | | R-unknown | | R-unknown | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-42. Module Version Register (MVR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-4 | MAJORREV | 0-Fh | Major revision number of the module. |
| 3-0 | MINORREV | 0-Fh | Minor revision number of the module. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.32 System Configuration Register (SYSC)

The AUTOIDLE bit controls a power-saving technique to reduce the logic power consumption of the module interface; that is, when the feature is enabled, the interface clock is gated off until the module interface is accessed. When the SOFTRESET bit is set high, it causes a full device reset. The system configuration register (SYSC) is shown in Figure 24-59 and described in Table 24-43.

**Figure 24-59. System Configuration Register (SYSC)**

| 15 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | IDLEMODE | | ENAWAKEUP | SOFTRESET | AUTOIDLE |
| R-0 | | R/W-0 | | R/W-0 | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-43. System Configuration Register (SYSC) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-5 | Reserved | 0 | Reserved. |
| 4-3 | IDLEMODE | 0-3h | Power management req/ack control. |
| | | 0 | Force idle: Idle request is acknowledged unconditionally. |
| | | 1h | No-idle: Idle request is never acknowledged. |
| | | 2h | Smart idle: Acknowledgement to an idle request is given based in the internal activity of the module. |
| | | 3h | Smart idle Wakeup: Acknowledgement to an idle request is given based in the internal activity of the module. The module is allowed to generate wakeup request. |
| 2 | ENAWAKEUP | | Wakeup control. |
| | | 0 | Wakeup is disabled. |
| | | 1 | Wakeup capability is enabled. |
| 1 | SOFTRESET | | Software reset. Set this bit to 1 to trigger a module reset. This bit is automatically reset by the hardware. Read returns 0. |
| | | 0 | Normal mode. |
| | | 1 | Module is reset. |
| 0 | AUTOIDLE | | Internal interface clock-gating strategy. |
| | | 0 | Clock is running. |
| | | 1 | Automatic interface clock-gating strategy is applied based on interface activity. |

### 24.3.33 System Status Register (SYSS)

The system status register (SYSS) is shown in Figure 24-60 and described in Table 24-44.

**Figure 24-60. System Status Register (SYSS)**

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | RESETDONE |
| R-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-44. System Status Register (SYSS) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-1 | Reserved | 0 | Reserved. |
| 0 | RESETDONE | | Internal reset monitoring. |
| | | 0 | Internal module reset is ongoing. |
| | | 1 | Reset complete. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.34 Wake-Up Enable Register (WER)

The wake-up enable register (WER) is used to mask and unmask a UART event that subsequently notifies the system. An event is any activity in the logic that can cause an interrupt and/or an activity that requires the system to wake up. Even if wakeup is disabled for certain events, if these events are also an interrupt to the UART, the UART still registers the interrupt as such. The wake-up enable register (WER) is shown in Figure 24-61 and described in Table 24-45.

**Figure 24-61. Wake-Up Enable Register (WER)**

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXWAKEUPEN | RLS_INTERRUPT | RHR_INTERRUPT | RX_ ACTIVITY | DCD_ACTIVITY | RI_ ACTIVITY | DSR_ACTIVITY | CTS_ACTIVITY |
| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-45. Wake-Up Enable Register (WER) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | TXWAKEUPEN | | Wake-up interrupt. |
| | | 0 | Event is not allowed to wake up the system. |
| | | 1 | Event can wake up the system: Event can be: THRIT or TXDMA request and/or TXSATUSIT. |
| 6 | RLS_ INTERRUPT | | Receiver line status interrupt. |
| | | 0 | Event is not allowed to wake up the system. |
| | | 1 | Event can wake up the system. |
| 5 | RHR_ INTERRUPT | | RHR interrupt. |
| | | 0 | Event is not allowed to wake up the system. |
| | | 1 | Event can wake up the system. |
| 4 | RX_ ACTIVITY | 0 | Event is not allowed to wake up the system. |
| | | 1 | Event can wake up the system. |
| 3 | DCD_ACTIVITY | 0 | Event is not allowed to wake up the system. |
| | | 1 | Event can wake up the system. |
| 2 | RI_ ACTIVITY | 0 | Event is not allowed to wake up the system. |
| | | 1 | Event can wake up the system. |
| 1 | DSR_ACTIVITY | 0 | Event is not allowed to wake up the system. |
| | | 1 | Event can wake up the system. |
| 0 | CTS_ ACTIVITY | 0 | Event is not allowed to wake up the system. |
| | | 1 | Event can wake up the system. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.35 Carrier Frequency Prescaler Register (CFPS)

Since the consumer IR (CIR) works at modulation rates of 30–56.8 kHz, the 48 MHz clock must be prescaled before the clock can drive the IR logic. The carrier frequency prescaler register (CFPS) sets the divisor rate to give a range to accommodate the remote control requirements in BAUD multiples of 12×. The value of the CFPS at reset is 105 decimal (69h), which equates to a 38.1 kHz output from starting conditions. The 48 MHz carrier is prescaled by the CFPS that is then divided by the 12× BAUD multiple. The carrier frequency prescaler register (CFPS) is shown in Figure 24-62 and described in Table 24-46.

**Figure 24-62. Carrier Frequency Prescaler Register (CFPS)**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CFPS | |
| R-0 | | R/W-69h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-46. Carrier Frequency Prescaler Register (CFPS) Field Descriptions**

| Bit | Field | Value | Description | | |
|---|---|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. | | |
| 7-0 | CFPS | 0-FFh | System clock frequency prescaler at (12× multiple). CFPS = 0 is not supported. Examples for CFPS values: | | |
| | | | **Target Frequency (kHz)** | **CFPS (decimal)** | **Actual Frequency (kHz)** |
| | | | 30 | 133 | 30.08 |
| | | | 32.75 | 122 | 32.79 |
| | | | 36 | 111 | 36.04 |
| | | | 36.7 | 109 | 36.69 |
| | | | 38 | 105 | 38.1 |
| | | | 40 | 100 | 40 |
| | | | 56.8 | 70 | 57.14 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.36 Divisor Latches Low Register (DLL)

The divisor latches low register (DLL) with the DLH register stores the 14-bit divisor for generation of the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor, DLL stores the least-significant part of the divisor. The DLL register is shown in Figure 24-63 and described in Table 24-47.

**NOTE:** DLL and DLH can be written to only before sleep mode is enabled (before IER[4] is set).

**Figure 24-63. Divisor Latches Low Register (DLL)**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | CLOCK_LSB | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-47. Divisor Latches Low Register (DLL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | CLOCK_LSB | 0-FFh | Divisor latches low. Stores the 8 LSB divisor value. |

### 24.3.37 Divisor Latches High Register (DLH)

The divisor latches high register (DLH) with the DLL register stores the 14-bit divisor for generation of the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor, DLL stores the least-significant part of the divisor. The DLH register is shown in Figure 24-64 and described in Table 24-48.

**NOTE:** DLL and DLH can be written to only before sleep mode is enabled (before IER[4] is set).

**Figure 24-64. Divisor Latches High Register (DLH)**

| 15 | 6 | 5 | 0 |
|---|---|---|---|
| Reserved | | CLOCK_MSB | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-48. Divisor Latches High Register (DLH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-6 | Reserved | 0 | Reserved. |
| 5-0 | CLOCK_MSB | 0-3Fh | Divisor latches high. Stores the 6 MSB divisor value. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.38 Enhanced Feature Register (EFR)

The enhanced feature register (EFR) enables or disables enhanced features. Most enhanced functions apply only to UART modes, but EFR[4] enables write accesses to FCR[5:4], the TX trigger level, which is also used in IrDA modes. The enhanced feature register (EFR) is shown in Figure 24-65 and described in Table 24-49.

**Figure 24-65. Enhanced Feature Register (EFR)**

| 15 | | | | | 8 |
|---|---|---|---|---|---|
| Reserved | | | | | |
| R-0 | | | | | |
| 7 | 6 | 5 | 4 | 3 | 0 |
| AUTOCTSEN | AUTORTSEN | SPECIALCHARDETECT | ENHANCEDEN | SWFLOWCONTROL | |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-49. Enhanced Feature Register (EFR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7 | AUTOCTSEN | | Auto-CTS enable bit (UART mode only). |
| | | 0 | Normal operation. |
| | | 1 | Auto-CTS flow control is enabled; transmission is halted when the $\overline{CTS}$ pin is high (inactive). |
| 6 | AUTORTSEN | | Auto-RTS enable bit (UART mode only). |
| | | 0 | Normal operation. |
| | | 1 | Auto-RTS flow control is enabled; $\overline{RTS}$ pin goes high (inactive) when the receiver FIFO HALT trigger level, TCR[3:0], is reached and goes low (active) when the receiver FIFO RESTORE transmission trigger level is reached. |
| 5 | SPECIALCHARDETECT | | Special character detect (UART mode only). |
| | | 0 | Normal operation. |
| | | 1 | Special character detect enable. Received data is compared with XOFF2 data. If a match occurs, the received data is transferred to RX FIFO and the IIR[4] bit is set to 1 to indicate that a special character was detected. |
| 4 | ENHANCEDEN | | Enhanced functions write enable bit. |
| | | 0 | Disables writing to IER[7:4], FCR[5:4], and MCR[7:5]. |
| | | 1 | Enables writing to IER[7:4], FCR[5:4], and MCR[7:5]. |
| 3-0 | SWFLOWCONTROL | 0-Fh | Combinations of software flow control can be selected by programming this bit. XON1 and XON2 should be set to different values if the software flow control is enabled. See Table 24-50. In IrDA mode, EFR[1:0] selects the IR address to check. See *IR Address Checking*. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 24-50. EFR[3:0] Software Flow Control Options**

| EFR[3:0] | | | | |
|---|---|---|---|---|
| Bit 3 | Bit 2 | Bit 1 | Bit 0 | TX/RX Software Flow Control |
| 0 | 0 | X | X | No transmit flow control |
| 1 | 0 | X | X | Transmit XON1, XOFF1 |
| 0 | 1 | X | X | Transmit XON2, XOFF2 |
| 1 | 1 | X | X | Transmit XON1, XON2: XOFF1, XOFF2(1) |
| X | X | 0 | 0 | No receive flow control |
| X | X | 1 | 0 | Receiver compares XON1, XOFF1 |
| X | X | 0 | 1 | Receiver compares XON2, XOFF2 |
| X | X | 1 | 1 | Receiver compares XON1, XON2: XOFF1, XOFF2(1) |

(1) The XON1 and XON2 characters or the XOFF1 and XOFF2 characters must be transmitted/received sequentially with XON1/XOFF1 followed by XON2/XOFF2.

### 24.3.39 XON1/ADDR1 Register

In UART mode, XON1 character; in IrDA mode, ADDR1 address 1. The XON1/ADDR1 register is shown in Figure 24-66 and described in Table 24-51.

**Figure 24-66. XON1/ADDR1 Register**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | XONWORD1 | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-51. XON1/ADDR1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | XONWORD1 | 0-FFh | Stores the 8-bit XON1 character in UART modes and ADDR1 address 1 in IrDA modes. |

### 24.3.40 XON2/ADDR2 Register

In UART mode, XON2 character; in IrDA mode, ADDR2 address 2. The XON2/ADDR2 register is shown in Figure 24-67 and described in Table 24-52.

**Figure 24-67. XON2/ADDR2 Register**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | XONWORD2 | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-52. XON2/ADDR2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 5 | XONWORD2 | 0-FFh | Stores the 8-bit XON2 character in UART modes and ADDR2 address 2 in IrDA modes. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.41 XOFF1 Register

In UART mode, XOFF1 character. The XOFF1 register is shown in Figure 24-68 and described in Table 24-53.

**Figure 24-68. XOFF1 Register**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | XOFFWORD1 | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-53. XOFF1 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | XOFFWORD1 | 0 | Stores the 8-bit XOFF1 character in UART modes. |

### 24.3.42 XOFF2 Register

In UART mode, XOFF2 character. The XOFF2 register is shown in Figure 24-69 and described in Table 24-54.

**Figure 24-69. XOFF2 Register**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | XOFFWORD2 | |
| R-0 | | R/W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-54. XOFF2 Register Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | XOFFWORD2 | 0-FFh | Stores the 8-bit XOFF2 character in UART modes. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.43 Transmit Frame Length Low Register (TXFLL)

The transmit frame length low register (TXFLL) and the TXFLH register hold the 13-bit transmit frame length (expressed in bytes). TXFLL holds the LSBs and TXFLH holds the MSBs. The frame length value is used if the frame length method of frame closing is used. The transmit frame length low register (TXFLL) is shown in Figure 24-70 and described in Table 24-55.

**Figure 24-70. Transmit Frame Length Low Register (TXFLL)**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | TXFLL | |
| R-0 | | W-0 | |

LEGEND: R/W = Read/Write; W = Write only; -*n* = value after reset

**Table 24-55. Transmit Frame Length Low Register (TXFLL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | TXFLL | 0-FFh | LSB register used to specify the frame length. |

### 24.3.44 Transmit Frame Length High Register (TXFLH)

The transmit frame length high register (TXFLH) and the TXFLL register hold the 13-bit transmit frame length (expressed in bytes). TXFLL holds the LSBs and TXFLH holds the MSBs. The frame length value is used if the frame length method of frame closing is used. The transmit frame length high register (TXFLH) is shown in Figure 24-71 and described in Table 24-56.

**Figure 24-71. Transmit Frame Length High Register (TXFLH)**

| 15 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | TXFLH | |
| R-0 | | W-0 | |

LEGEND: R/W = Read/Write; W = Write only; -*n* = value after reset

**Table 24-56. Transmit Frame Length High Register (TXFLH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-5 | Reserved | 0 | Reserved. |
| 4-0 | TXFLH | 0-1Fh | MSB register used to specify the frame length. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.45 Received Frame Length Low Register (RXFLL)

The received frame length low register (RXFLL) and the RXFLH register hold the 12-bit receive maximum frame length. RXFLL holds the LSBs and RXFLH holds the MSBs. If the intended maximum receive frame length is *n* bytes, program RXFLL and RXFLH to be *n* + 3 in SIR or MIR modes and *n* + 6 in FIR mode (+3 and +6 are the result of frame format with CRC and stop flag; two bytes are associated with the FIR stop flag). The received frame length low register (RXFLL) is shown in Figure 24-72 and described in Table 24-57.

**Figure 24-72. Received Frame Length Low Register (RXFLL)**

| 15 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | RXFLL | |
| R-0 | | W-0 | |

LEGEND: R/W = Read/Write; W = Write only; -*n* = value after reset

**Table 24-57. Received Frame Length Low Register (RXFLL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-0 | RXFLL | 0-FFh | LSB register used to specify the frame length in reception. |

### 24.3.46 Received Frame Length High Register (RXFLH)

The received frame length high register (RXFLH) and the RXFLL register hold the 12-bit receive maximum frame length. RXFLL holds the LSBs and RXFLH holds the MSBs. If the intended maximum receive frame length is *n* bytes, program RXFLL and RXFLH to be *n* + 3 in SIR or MIR modes and *n* + 6 in FIR mode (+3 and +6 are the result of frame format with CRC and stop flag; two bytes are associated with the FIR stop flag). The received frame length high register (RXFLH) is shown in Figure 24-73 and described in Table 24-58.

**Figure 24-73. Received Frame Length High Register (RXFLH)**

| 15 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | RXFLH | |
| R-0 | | W-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-58. Received Frame Length High Register (RXFLH) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-4 | Reserved | 0 | Reserved. |
| 3-0 | RXFLH | 0-Fh | MSB register used to specify the frame length in reception. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 24.3.47 UART Autobauding Status Register (UASR)

The UART autobauding status register (UASR) returns the speed, the number of bits by characters, and the type of parity in UART autobauding mode. In autobauding mode, the input frequency of the UART modem must be fixed to 48 MHz. Any other module clock frequency results in incorrect baud rate recognition. The UART autobauding status register (UASR) is shown in Figure 24-74 and described in Table 24-59.

**NOTE:**

- This register is used to set up transmission according to characteristics of previous reception, instead of LCR, DLL, and DLH registers when UART is in autobauding mode.
- To reset the autobauding hardware (to start a new “AT” detection) or to set the UART in standard mode (no autobaud), MDR1[2:0] must be set to 7h (reset state), then set to 2h (UART in autobaud mode) or cleared to 0 (UART in standard mode).

Usage limitation:

- Only 7 and 8 bits character (5 and 6 bits not supported).
- 7 bits character with space parity not supported.
- Baud rate between 1200 and 115 200 bp/s (10 possibilities).

**Figure 24-74. UART Autobauding Status Register (UASR)**

| 15 | 8 | 7 | 6 | 5 | 4 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | PARITYTYPE | | BITBYCHAR | SPEED | |
| R-0 | | R-0 | | R-0 | R-0 | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 24-59. UART Autobauding Status Register (UASR) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 15-8 | Reserved | 0 | Reserved. |
| 7-6 | PARITYTYPE | 0-3h | Type of the parity in UART autobauding mode. |
| | | 0 | No parity identified |
| | | 1h | Parity space |
| | | 2h | Even parity |
| | | 3h | Odd parity |
| 5 | BITBYCHAR | | Number of bits by characters. |
| | | 0 | 7-bit character identified. |
| | | 1 | 8-bit character identified. |
| 4-0 | SPEED | 0-1Fh | Speed |
| | | 0 | No speed identified. |
| | | 1h | 115 200 baud |
| | | 2h | 57 600 baud |
| | | 3h | 38 400 baud |
| | | 4h | 28 800 baud |
| | | 5h | 19 200 baud |
| | | 6h | 14 400 baud |
| | | 7h | 9600 baud |
| | | 8h | 4800 baud |
| | | 9h | 2400 baud |
| | | Ah | 1200 baud |
| | | Bh-1Fh | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

# Universal Serial Bus Subsystem (USBSS)

The Universal Serial Bus Subsystem (USBSS) contains dual USB 2.0 compliant modules. Both modules are configurable as a USB Host, a USB Device or a dual role OTG Host/Device via the On-The-Go (OTG) supplement of the USB 2.0 Specification. Both USB modules operate independently of each other.

**Topic** **Page**

**25.1 Introduction** ..... **3653**
**25.2 Features Supported** ..... **3653**
**25.3 Functional Block Diagram** ..... **3654**
**25.4 Industry Standard(s) Compliance** ..... **3655**
**25.5 Architecture** ..... **3656**
**25.6 Error Handling** ..... **3666**
**25.7 Protocol Description(s)** ..... **3667**
**25.8 DMA** ..... **3699**
**25.9 Registers** ..... **3726**


Copyright © 2011–2016, Texas Instruments Incorporated

## 25.1 Introduction

Each USB module is built around the a USB OTG controller and PHY. Each USB controller has 32K Bytes of Endpoint FIFO RAM and support for up to 15 Endpoints in addition to Endpoint 0. The USB subsystem makes use of a Direct Memory Access (DMA) Controller for accelerating endpoint data movement via dedicated DMA hardware with minimal intervention by the CPU.

The two USB modules share a single DMA controller and accompanying Queue Manager, Interrupt Pacer, Power Management module, and PHY clock.

For the remainder of this chapter the term USB controller or USB PHY refers to any of the two USB controllers or PHYs existing within the USB subsystem. The term USB module is used to mean/refer to any of the two USB modules. USB0 is used to refer to the first instantiation of a USB modules and USB1 is used to refer to the other USB module instantiation.

## 25.2 Features Supported

The main features of the USB Subsystem are:

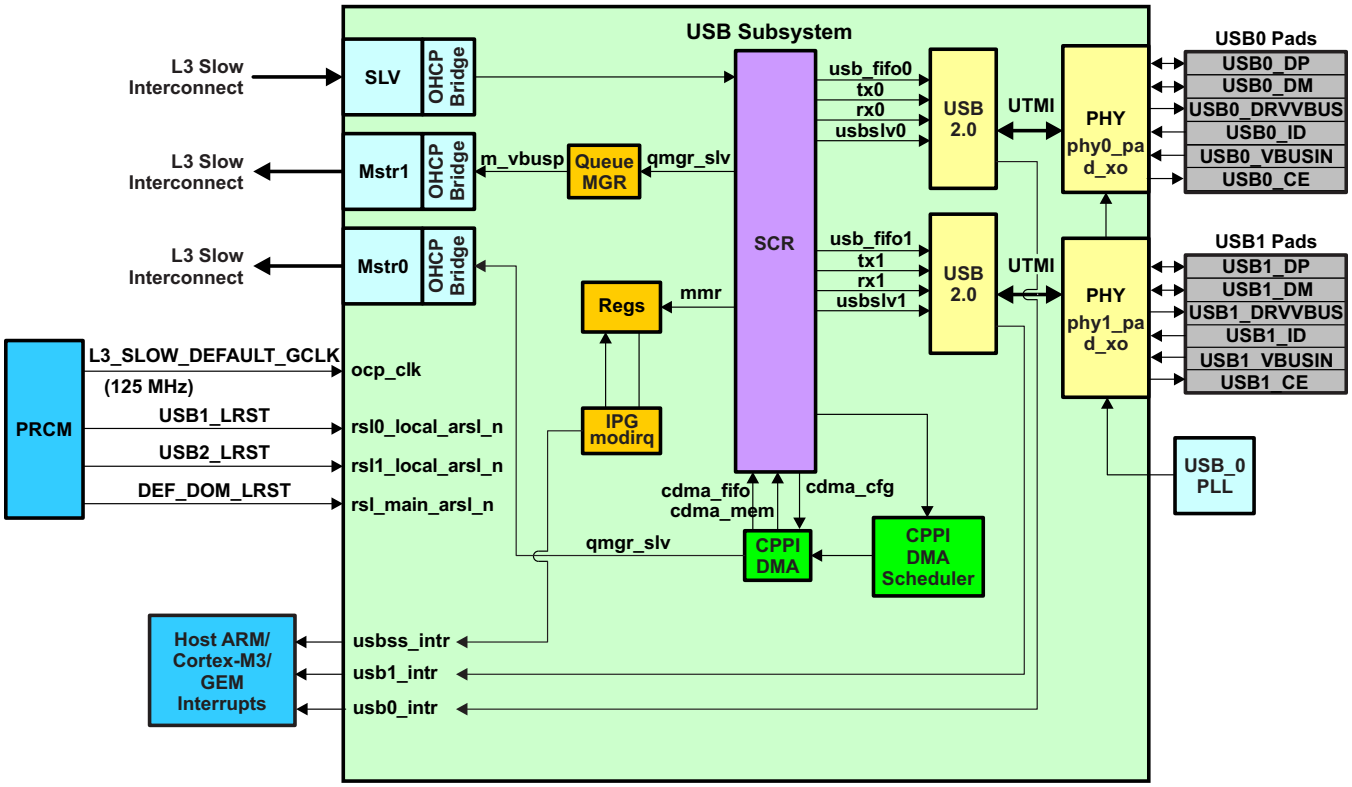
- Two USB modules where each USB module is built around a USB OTG controller and PHY.
- Supports USB Device operations at High Speed (Up to 480 Mb/s) or Full Speed (Up to 12 Mb/s).
- Supports USB Host operations at High Speed (Up to 480 Mb/s), Full Speed (Up to 12 Mb/s) and Low Speed (Up to 1.5Mb/s).
- USB 2.0 extensions for OTG Session Request Protocol (SRP).
- USB 2.0 extensions for OTG Host Request Protocol (HRP).
- Supports all USB standard transfer types (control, bulk, interrupt, and isochronous) as both a USB Host as well as a USB Device.
- Supports high bandwidth ISO mode.
- Supports 15 Transmit (Tx) and 15 Receive (Rx) Endpoints excluding Endpoint 0.
- Includes a 32KB Endpoint FIFO RAM per USB Module with user programmable Endpoint FIFO sizes.
- Includes RNDIS mode for accelerating RNDIS type protocols using short packet termination over USB.
- Includes CDC Linux mode for accelerating CDC type protocols using short packet termination over USB.
- Includes Generic RNDIS mode, an RNDIS-like mode for terminating RNDIS type protocols without using short packet termination for support for MSC class applications.
- Includes a DMA controller sub-module with 30 Rx and 30 Tx simultaneous data connections/channels.
- Includes DMA scheduler.
- The DMA supports host and packet descriptors formats.
- The DMA supports stall on buffer starvation.
- The DMA supports capability of allocating data buffer sizes up to 4Mbytes per single transfer.
- Provides a Queue Manager Module with 156 Queues for Queuing and De-Queuing packets.
- Supports DMA pacing logic for interrupts.


Copyright © 2011–2016, Texas Instruments Incorporated

## 25.3 Functional Block Diagram

Figure 25-1 displays a high-level functional block diagram covering the major blocks of the USB subsystem. It captures the general data paths and displays the interaction between each blocks of the subsystem at a high level.

**Figure 25-1. USB Subsystem Block Diagram**




### 25.3.1 L3 Slow Interconnect

This is the L3 master / slave interface to the device Memory Mapped Registers (MMRs) and system resources. The CPPI DMA and Queue Manager have also their own dedicated interface to the system resource through the master interfaces. CPU accesses the USB peripheral through the slave L3 Interface.

### 25.3.2 Memory Mapped Registers

The USB register file (USB core and subsystem registers and CPPI DMA registers that are accessible to the CPU) reside in a block within the USBSS.

### 25.3.3 USB 2.0 Controller

The USB subsystem supports two independent USB Modules. Each USB module comprises of a Dual Role USB 2.0 controllers with the support for OTG extensions.

### 25.3.4 USB PHY

The PHY is connected to the USB controller via 8-bit USB 2.0 Transceiver Macrocell Interface (UTMI). The PHY does not have a built-in charge pump thus an external power source is required to source the 5V VBUS power when operating as a USB Host. The PHY has built in charge detection for device mode and external charge control capability for a host application.


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.3.5 USB Pins

The USB subsystem external interface signals comprises two differential data lines, two Voltage Buses Level Detection Pins (VBUSIN), and two USB DRVVBUS signals which are used to enable/disable external 5V Power Source. The USB ID pins are used to configure the USB controller's role. Charge detect enable signals are present within both PHYs. For more details about the USB0 and USB1 mode registers, see Section 25.9.2.1.23 and Section 25.9.2.2.23, respectively.

### 25.3.6 DMA

The DMA alongside with the Queue Manager and the Scheduler are used to perform data transfer for Endpoints 1 to 15.

## 25.4 Industry Standard(s) Compliance

The USB subsystem complies with the following standards:

- Complies with the USB 2.0 Specification and the On-The-Go Supplement of the USB 2.0 Specification.


Copyright © 2011–2016, Texas Instruments Incorporated

## 25.5 Architecture

Each USB controller within the device supports 15 Transmit (Tx) endpoints and 15 Receive (Rx) endpoints in addition to Endpoint 0. These endpoints are used for IN transactions or OUT transactions depending on the role of the USB Controller.

When used as a USB Device, IN transactions are processed through Tx endpoints and OUT transactions are processed through Rx endpoints. When used as a USB Host, IN transactions are processed through Rx endpoints and OUT transactions is processed through Tx endpoints.

Endpoints can be individually configured in software to handle Bulk, Interrupt, Isochronous, or Control transfers. Further, the endpoints can also be dynamically allocated to different target device functions on the fly – maximizing the number of devices that can be simultaneously supported. Each endpoint requires a block of memory allocated within the Module FIFO to be associated with it.

Each USB module has a single block of FIFO RAM (32 Kbytes in size) that is shared by all endpoints. The FIFO for Endpoint 0 is required to be 64 bytes deep and is single buffered. The first 64 Bytes of the FIFO RAM is reserved by hardware for Endpoint 0 and for this reason user software does not need (nor has the capability) to allocate a FIFO block for Endpoint 0. The rest of the FIFO RAM is user configurable with regard to the other endpoints. The size of an individual endpoint FIFO may range from 8 to 4096 bytes and can either be single or double buffered. Supported sizes must conform to the following sizes:

- FS/LS: 8, 16, 32, 64, 1024 bytes
- HS: 64, 128, 512, 1024, 2048, 4096 bytes.

Separate FIFOs may be associated with each endpoint: alternatively a Tx endpoint and the Rx endpoint with the same Endpoint number can be configured to use the same FIFO memory space, for example to reduce the size of RAM block needed, provided they can never be active at the same time.

The user can process USB transactions entirely from the CPU via the L3 Interconnect or can optionally use the DMA to automatically perform data transfer for endpoints 1-15. A CPU must be used to service Endpoint 0 transactions.

### *25.5.1 USB Controller Operational Mode*

#### 25.5.1.1 USB Controller Role Assuming Method

The two USB modules can be used in a range of different environments. They can be used as a High-Speed or a Full-Speed USB Device attached to a conventional USB Host (such as a PC) in a point-to-point type of arrangement, they can be used as a USB Host in which the USB modules can also be used with another USB Device of any speed (High-,Full-, or Low-speed) in a point-to-point arrangement, or they can be used as a USB Host to a range of USB Devices in a multi-point setup via a USB Hub. Note that the role each USB module operates is independent of each other.

USB transactions from the controller are issued and received during a valid USB Session. A session is defined only when VBUS is above the VBUS_SESS_VLD as specified in the OTG Supplement.

The USB 2.0 controller session starts when one of the following events occurs

- Firmware assumes it will be operating as the USB Host and sets the DEVCTL[SESSION] bit

  If the firmware writes to the DEVCTL[SESSION] bit, the USB controller will attempt to determine if it is the 'A-Device', or if it is the 'B-Device' by sensing the resistance on the USBx_ID pin.

  An A-Device is a device with a the Standard A Receptacle or a device with a Micro-A plug inserted into it's receptacle. A B-Device is defined as a device with a (mini-)B receptacle, or a Micro-A/B receptacle with the B end of the plug inserted into it's receptacle.

  If the USB controller determines it is the A-Device it will turn on VBUS and wait until the USBx_VBUSIN pin is >= 4.4V. Assuming that the voltage level of the USBX_VBUSIN is found to within the VBUS Valid range according to the USB 2.0 Specification, the controller will then default to the USB Host and wait for a device to connect to the bus before attempting to enumerate the device.

  If the controller senses it is a B-Device, it will use the Session Request Protocol as defined in the USB OTG Supplement to request a session from the A-Device and default to operating as a USB Device. If desired, firmware can force the USB Controller to act as an A-Device or a B-Device by writing to IDDIG bit in the USB Mode register before it sets the DEVCTL[SESSION] bit. This will override the ID sense


Copyright © 2011–2016, Texas Instruments Incorporated

value and force the controller to operate as the USB Host or a USB Device.

- The USB Controller senses a voltage (0.8 <=USBx_VBUSIN <= 2.1V) on the USBx_VBUSIN pin.

If DEVCTL[SESSION] bit is clear when the USB Controller senses that a voltage of >= VBUS_SESS_VLD on the USBx_VBUSIN pin, firmware should assume was another device on the bus has attempted to start a session. The USB controller will assume that it is the B-Device and automatically set DEVCTL[SESSION] bit. When USBx_VBUSIN >= VBUS_SESS_VLD, it will enable the 1.5 KOhm D+ pull-up. The subsequent Reset Interrupt received by the USB controller signifies the start of the session and the USB Controller should then proceed to behave as a Full-Speed or High-Speed Capable USB Device during enumeration.

#### 25.5.1.2 Soft Connect

After a Power-on Reset (POR) or sofware initiated USB Module Reset, the SOFTCONN bit of POWER register is cleared to 0. The controller will therefore appear disconnected from the bus (D+/D- appear as high impedance) until firmware has set the SOFTCONN bit to 1. This feature allows the application software can then choose when to connect to the USB -or- when to remain disconnected. Applications with an overall lengthy initialization procedure may use this to ensure that all other initialization is complete and the CPU is ready to perform enumeration before setting the Soft Connect bit and connecting to the USB. Once the SOFTCONN bit has been set, the software can also simulate a disconnect by clearing this bit to 0.

#### 25.5.1.3 Suspend

When operating as an active USB Host, firmware sets the POWER[SUSPEND] bit to enter into Suspend mode. Following the completion of the current transaction, the controller will disable it's scheduler and stop the frame counter.

When operating as an active USB Device, the controller monitors activity on the D+/D- and automatically transitions into suspend mode when it doesn't detect any valid USB activity for a period of greater than or equal to 3 ms. The USB controller will automatically set the POWER[SUSPEND] bit and then generate an Suspend Interrupt. At this point, the controller can then be left active or the application may arrange to disable the controller by stopping its clock. If the software disabled the USB clock, the controller will not then be able to detect Resume signaling and must rely on external hardware to detect Resume signaling so that the clock to the controller can be restarted.

#### 25.5.1.4 Resume

When operating as a active USB Host, firmware clears the POWER[SUSPEND] bit to exit into Suspend mode and set the POWER[RESUME] bit for 20ms. The controller will automatically send Resume signaling onto the bus . Following the completion of this 20 ms period, firmware should clear the POWER[RESUME] bit to restart the transaction scheduler and frame counter. If remote Wake-up resume signaling is detected while the controller is in suspend mode, the controller will automatically exit suspend and set the POWER[RESUME] bit, take over generating the Resume Signal ,and finally generate an Resume Interrupt

When operating as a active USB Device, if resume signaling is detected on the bus, the clock to the controller must be restarted (if stopped during suspend). The controller will automatically exit Suspend mode and generate an interrupt. If the software wants to initiate a remote wakeup while the controller is in Suspend mode, it should set the POWER[RESUME] bit to 1. The software should leave then this bit set for approximately 10 ms (minimum of 2 ms, a maximum of 15 ms) before resetting it to 0. **NOTE:** No resume interrupt will be generated when the software initiates a remote wakeup.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.5.1.5 Reset

When the controller is operating as a USB Host, a USB Reset condition will be generated on the bus if the Reset bit is set by firmware. If the High-Speed Enable (HSENA) bit is set, the controller will attempt to negotiate for high-speed operation.

Once the RESET is cleared, the controller will automatically start the frame counter and schedule USB transactions.

When the controller is operating as an USB Device it will automatically perform the following functions when it sees a USB Reset bus condition.

- Functional Address (FADDR) is cleared 0
- Endpoint Index Register (INDEX) is cleared to 0
- All endpoint FIFOs are flushed
- All Endpoint Control/Status registers are cleared
- All Endpoint Interrupts are Enabled at the controller level
- A USB Reset Interrupt is generated.

If the High Speed Enable (HSENA) bit within the POWER register is set, the controller will automatically attempt to negotiate with the USB Host for High-Speed Operation. Whether high-speed operation is selected is indicated by HSMODE bit of POWER register. When the application software receives a reset interrupt, it should close any open pipes and wait for bus enumeration to begin.

#### 25.5.1.6 USB Controller Endpoint Control / Status Register Access

The USB controller provides two mechanisms for accessing endpoint control and status registers; Indexed and Non-Indexed method.

- **Indexed Endpoint Control/Status Register Space**.

  This register space can be thought of as a region populated with Proxy Registers. The endpoint register space mapped at this region is selected by programming the corresponding INDEX register of the controller. By programming the INDEX register with the corresponding endpoint number, the control and status register corresponding to that particular endpoint is accessible from this Indexed Region. In other words, Index Register region behaves as a proxy to access a selected endpoint registers. Indexed access enables the controller to access different endpoints without reconfiguring addressing..

- **Non-indexed Endpoint Control/Status Register Space**.

  The Endpoint Control / Status registers are located at dedicated endpoint registers in the respective USB Controllers

This use of Index / Non Indexed access modes allows the firmware to access an endpoint register region directly from its unique space (as specified within a non-indexed region), or from a proxy space by programming the corresponding Index register.

Note: The control and status registers/regions apply to both Tx and Rx of a given Endpoint. That is, Endpoint 1 Tx and Endpoint 1 Rx registers occupy the same register space.

#### 25.5.1.7 Dynamic Endpoint FIFO Sizing

Each USB module supports a total of 32K bytes of FIFO RAM to dynamically allocate FIFO to all endpoints in use. Each endpoint requires a FIFO RAM block to be associated with it. There is one set of registers per endpoint available for FIFO allocation, excluding endpoint 0. FIFO configuration registers are Indexed registers and cannot be accessed directly; configuration takes place via accessing proxy registers. The INDEX register at addresses must be set to the appropriate endpoint before writing to the respective Endpoint FIFO configuration registers. The Endpoint FIFO configuration registers do not have non-indexed registers.


Copyright © 2011–2016, Texas Instruments Incorporated

The allocation of FIFO space to the different endpoints requires programming for each Tx and Rx endpoint with the following information:

- The start address of the FIFO within the RAM block
- The maximum size of packet to be supported
- Whether the FIFO is single buffered or double buffered

The maximum packet size and double buffering together define the total of space that needs to be allocated to the FIFO per Endpoint

It is the responsibility of the firmware to ensure that all the Tx and Rx endpoints that are active in the current USB configuration have a block of FIFO RAM allocated for their use. The first 64 bytes of FIFO memory is statically configured for Endpoint 0 use and should be treated as reserved for all other endpoints. For endpoints other than endpoint 0, the FIFO RAM interface is configurable and must have a minimum size of 8 bytes and should be capable of buffering either 1 or 2 packets. Separate FIFOs may be associated with each endpoint: alternatively a Tx endpoint and the Rx endpoint with the same Endpoint number can be configured to use the same FIFO to reduce the size of RAM block needed, provided they are never active at the same time.

**NOTE:** The option of dynamically setting FIFO sizes only applies to Endpoints 1-15. Endpoint 0 FIFO has a fixed size (64 bytes) and a fixed location which is the first 64 Bytes of FIFO (FIFO addresses 0-63).

#### 25.5.1.8 USB 2.0 Test Modes

Each USB2.0 controller supports the four USB 2.0 test modes (Test_SE0_NAK, Test_J, Test_K, and Test_Packet) defined for high-speed functions. It also supports an additional “FIFO access” test mode that can be used to test the operation of the CPU interface, the DMA controller and the FIFO RAM block.

The test modes are entered by writing to the TESTMODE register. A test mode is usually requested by the host sending a SET_FEATURE request to Endpoint 0. When the software receives the request, it should wait until the Endpoint 0 transfer has completed (when it receives the Endpoint 0 interrupt indicating the status phase has completed) then write to the TESTMODE register.

**Note:** These test modes have no purpose in normal operation.

##### *25.5.1.8.1 TEST_SE0_NAK*

To enter the Test_SE0_NAK test mode, the software should set the TEST_SE0_NAK bit in the TESTMODE register to 1. The USB controller will then go into a mode in which it responds to any valid IN token with a NAK.

##### *25.5.1.8.2 TEST_J*

To enter the Test_J test mode, the software should set the TEST_J bit in the TESTMODE register to 1. The USB controller will then go into a mode in which it transmits a continuous J on the bus.

##### *25.5.1.8.3 TEST_K*

To enter the Test_K test mode, the software should set the TEST_K bit in the TESTMODE register to 1. The USB controller will then go into a mode in which it transmits a continuous K on the bus.

##### *25.5.1.8.4 TEST_PACKET*

To execute the Test_Packet, the software should:

1. Start a session (if the core is being used in Host mode.
2. Write the standard test packet (shown below) to the Endpoint 0 FIFO.
3. Write 8h to the TESTMODE register (TEST_PACKET = 1) to enter Test_Packet test mode.
4. Set the TxPktRdy bit in the HOST/PERI_CSR0 register (bit 1).

The 53 byte test packet to load is as follows (all bytes in hex). The test packet only has to be loaded once; the USB controller will keep re-sending the test packet without any further intervention from the software.


Copyright © 2011–2016, Texas Instruments Incorporated

Table 25-1 displays the 53 Bytes test packet content.

**Table 25-1. 53 Bytes Test Packet Content**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | AA | AA | AA | AA | AA | AA | AA |
| AA | EE | EE | EE | EE | EE | EE | EE |
| EE | FE | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | 7F | BF | DF |
| EF | F7 | FB | FD | FC | 7E | BF | DF |
| EF | F7 | FB | FD | 7E | | | |

This data sequence is defined in Universal Serial Bus Specification Revision 2.0 The USB controller will add the DATAA0 PID to the head of the data sequence and the CRC to the end.

#### *25.5.1.8.5 FIFO_ACCESS*

The FIFO Access test mode allows you to test the operation of CPU Interface, the DMA controller (if configured), and the RAM block by loading a packet of up to 64 bytes into the Endpoint 0 FIFO and then reading it back out again. Endpoint 0 is used because it is a bidirectional endpoint that uses the same area of RAM for its Tx and Rx FIFOs.

**NOTE:** The core does not need to be connected to the USB bus to run this test. If it is connected, then no session should be in progress when the test is run.

The test procedure is as follows:

1. Load a packet of up to 64 bytes into the Endpoint 0 Tx FIFO.
2. Set HOST/PERI_CSR0[TXPKTRDY].
3. Write 40h to the TESTMODE register (FIFO_ACCESS = 1).
4. Unload the packet from the Endpoint Rx FIFO, again.
5. Set HOST/PERI_CSR0[SERVICEDRXPKTRDY].

Writing 40h to the TESTMODE register causes the following sequence of events:

The Endpoint 0 CPU pointer (that records the number of bytes to be transmitted) is copied to the Endpoint 0 USB pointer (that records the number of bytes received).

1. The Endpoint 0 CPU pointer is reset.
2. HOST/PERI_CSR0[TXPKTRDY] is cleared.
3. HOST/PERI_CSR0[RXPKTRDY] is set.
4. An Endpoint 0 interrupt is generated (if enabled).

The effect of these steps is to make the Endpoint 0 controller act as if the packet loaded into the Tx FIFO has flushed and the same packet received over the USB. The data that was loaded in the Tx FIFO can now be read out of the Rx FIFO.

#### *25.5.1.8.6 FORCE_HOST*

The Force Host test mode enables you to instruct the core to operate in Host mode, regardless of whether it is actually connected to any peripheral; that is, the state of the CID input and the LINESTATE and HOSTDISCON signals are ignored. (While in this mode, the state of the HOSTDISCON signal can be read from the BDEVICE bit in the device control register (DEVCTL)) .

This mode, which is selected by writing 80h to the TESTMODE register (FORCE_HOST = 1), allows implementation of the USB Test_Force_Enable (USB 2.0 Specification Section 7.1.20). It can also be used for debugging PHY problems in hardware.


Copyright © 2011–2016, Texas Instruments Incorporated

While the FORCE_HOST bit remains set, the core enters the Host mode when the SESSION bit in DEVCTL is set to 1 and remains in the Host mode until the SESSION bit is cleared to 0 even if a connected device is disconnected during the session. The operating speed while in this mode is determined by the FORCE_HS and FORCE_FS bits in the TESTMODE register.

### 25.5.2 Clock Control

Two main clocks are used by the USB subsystem, the L3 clock and the UTMI/PHY clock.

The L3 Slow clock is the clock that is used to clock the controller and all other blocks of the USB Subsystem with the exception of the PHY. This clock is controlled by the PRCM. A clock rate of 60 MHz or greater must be used to guarantee core operation can deliver full USB 2.0 bandwidth (480 Mb/s). In other words, L3 Slow should not be less than 60 MHz.

The PHY clock is derived from the main device input clock. The DPLL is used to generate the required 960MHz clock frequency. The required UTMI clock is generated by the PHY from the input clock.

### 25.5.3 Signal Descriptions

The USB subsystem external interface signals are displayed within Table 25-2.

**Table 25-2. USBSS Interface Signals**

| Pin (x=0/1) | Type | Description |
|---|---|---|
| USBx_DP<br>USBx_DM | I/O | USBx data differential pair |
| USBx_DRVVBUS | O | USBx VBUS Supply Enable |
| USBx_VBUSIN | AI | USBx VBUS Voltage Sense |
| USB_ID | AI | USB ID Detection |
| USBx_CE(1) | I | USBx Charge Enable |

(1) "USBx_CE is not available on all instances of USB (for example, a pin may exist on USB0, but not USB1). Refer to the device Data Manual Terminal Functions section for details.

### 25.5.4 VBUS Voltage Sourcing Control

When any of the USB controllers assumes the role of a host, the USB is required to supply a +5V power source to an attached device(s) through the VBUS line. In order to achieve this task, the USB controller requires the use of an external power logic (or charge pump) capable of sourcing 5V power. The USB_DRVVBUS pin is used as a control signal to enable/disable this external power supply. The control on the USB_DRVVBUS is automatic and is handled by the USB controller. The control should be transparent to the user so long as the proper hardware connection and software initialization are in place. The USB controller drives the USB_DRVVBUS signal high when it assumes the role of a host during a valid USB session. When assuming the role of a device, the controller drives the USB_DRVVBUS signal low disabling the external charge pump; hence, no power is driven on the VBUS line by the controller. 5V VBUS is expected to be sourced by the external USB Host.

Note that when both USB Modules are self-powered and the device does not rely on the voltage on the VBUS voltage sourced by an external USB Host for controller operation when operating as a USB Device. The Voltage Level detected on the +5V VBUS is used to identify the presence of a Host and to power up the internal pull-up on the D+ line. The USB PHY continually monitors the voltage on the USB_VBUSIN and reports the status to USB controller.

### 25.5.5 D+/D- Pull-up/Pull-Down Resistors

Since the USB controllers are dual role controllers, the necessary required D+/D- pull-up/pull-down resistors need to be dynamically attached/detached and thus can not exist external to the device. The pull-up/pull-down resistors exist in the PHY, and are dynamically enabled and disabled based on the role the controller assumes during a USB Session.


Copyright © 2011–2016, Texas Instruments Incorporated

When assuming the role of a USB Host, the data lines are pulled to USB ground by the PHY enabling internal 15K Ohms pull-down resistors. When assuming the role of a USB Device the required 1.5K ohm pull-up resistor on the D+ line is enabled automatically signifying its capability to the external USB Host as a Full Speed or High Speed Capable device. If a High Speed capable device successfully negotiates to operate at the High Speed signal rate, the internal D+ is automatically disabled by the controller.

### 25.5.6 Interrupts

The USB Subsystem has three interrupts that can be routed to the CPU interrupt controller.

- USB Subsystem Interrupt
- USB0 Module Interrupt
- USB1 Module Interrupt

#### 25.5.6.1 USB Subsystem Interrupt

The following DMA interrupts are generated by the USB Subsystem DMA.

The USB Subsystem is capable of generating a single interrupt for DMA events. Table 25-3 describes the DMA interrupts generated from the USB Subsystem.

**Table 25-3. Subsystem Interrupts**

| Interrupt | CPPI DMA Packet Completion Grouping (EndPoint) | Description |
|---|---|---|
| rx_sop_starvation | n/a | DMA Queue Manager Start of Packet Starvation Error Interrupt |
| rx_mop_starvation | n/a | DMA Queue Manager Middle of Packet Starvation Error Interrupt |
| pd_cmp_flag | n/a | When the DMA packet is completed and the Packet Descriptor is pushed into the Queue Manager |
| tx_pkt_cmp_0 | 1, 2, …, 15 | USB0 Tx Combined DMA Packet Group completion Interrupt |
| rx_pkt_cmp_0 | 1, 2, …, 15 | USB0 Rx Combined DMA Packet Group completion Interrupt |
| tx_pkt_cmp_1 | 1, 2, …, 15 | USB1 Tx Combined DMA Packet Group completion Interrupt |
| rx_pkt_cmp_1 | 1, 2, …, 15 | USB1 Rx Combined DMA Packet Group completion Interrupt |

The USBSS DMA Queue Manager outputs a total of 60 DMA packet completion interrupts. Each Packet Completion Interrupt consists of a logical combination of the data from the 15 individual Endpoint packet completion signals from the Queue Manager. If any one of the Endpoint Packet Completion signals is set then an interrupt will be generated by the USB Subsystem

The starvation interrupts occur when the DMA Queue Manager is unable to allocate a buffer for an Rx buffer. This can occur either at the start of a packet or in the middle of a series of packets.

The pd_cmp_flag is only set when bit 31 of the original buffer length word (word 6) of a Packet Descriptor is set and when the packet is either transmitted to the XDMA or received from the XDMA using this PD.

In order to control the frequency of the generated USB Subsystem interrupts to the CPU, interrupt pacing has been added to each of the 60 individual packet completions. Each packet completion has its own 8-bit unsigned threshold. Each threshold can range from 0 to 255 based on the pd_cmp_flag. If the count of the packet completion exceeds the threshold; then an interrupt will be generated. This assumes that the packet completion is enabled. The thresholds for the 60 packet completions are programmed in the USBSS IRQ_DMA_THRESHOLD_(RX/TX)(0/1)_n registers.


Copyright © 2011–2016, Texas Instruments Incorporated

It is possible that the generation of interrupts has been reduced to an unacceptable delay due a lack of differed bits or thresholds set to too large of a number. A watch dog timer has been added which provides a secondary method of generating the interrupts. This system uses the USB start of frame pulse from the USB Controller, a counter, a threshold, and an enable. The frame sync pulse frequency is 1kHz in full-speed or low-speed mode, or 8kHz is high-speed mode.

**Example 1**: The DMA threshold is set to 10 and the frame threshold is set to 3 for a specific endpoint. Let's assume that the expected packet completion frequency is 5 completed packets per frame. In this example the first frame had 5 packets completed. The second frame had 5 packets completed. An interrupt is generated after the 2nd frame.

**Example 2**: The DMA threshold is set to 10 and the frame threshold is set to 3 for a specific endpoint. Let's assume that the expected packet completion frequency is 5 completed packets per frame. In this example the first frame had 5 packets completed. The second frame had only two packets completed and the third frame had 0 packets completed. There was some error or delay that caused the remaining packets to not complete. An interrupt is generated after the 3rd frame.

In both examples software will not be able to determine whether the interrupt was generated due to DMA threshold or frame threshold.

Each of the 60 Packet completions will have a frame counter, frame threshold, and frame enable. If the frame counter exceeds the frame threshold; then an interrupt will be generated. This assumes the frame enable counter has been enabled.

The registers for the 60 frame thresholds are stored in: USBSS_IRQ_FRAME_THRESHOLD_ab_c. Where:

(a) Tx or Rx

(b) 0 or 1

(c) 0, 1, 2, or 3

The 60 frame enables are stored in registers: USBSS IRQ_FRAME_ENABLE_0 and USBSS IRQ_FRAME_ENABLE_1.

The individual frame count / packet completion count registers will be reset to zero when one of the below conditions occurs:

- Reset signal is active.
- Packet completion threshold has been exceeded.
- Packet completion count is equal to 255.
- Frame count threshold has been exceeded.
- Frame count is equal to 255.

When the low to high transition occurs for any of the four CPPI DMA Packet completions, a hardware signal is sent to the MODIRQ module. This will cause an interrupt (if enabled via software). When all packets have been removed from the queue, the pending signal is returned to a 0. Software may choose to use one of the unassigned queues if an interrupt is not wanted upon the completion of a packet.

The interrupts listed in Table 25-3 can be enabled (or disabled) by setting (or clearing) the appropriate IRQENABLE_SET (or IRQENABLE_CLR) bits in the MMR registers.

To clear the interrupts it is required to write 1's to the IRQSTAUS registers. It is possible to manually set the interrupts by writing 1's to the IRQSTATUS_RAW registers.


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.5.6.2 USB Module Interrupts

Each of the USB modules generates an interrupt that can be routed to the CPU interrupt controller.

Each USB Module allows for the interrupt to be generated by one of two methods depending on the uint field of the Control Register of each USB Module

- USB Module Level Interrupts.
- USB Controller Level Interrupts.

#### *25.5.6.2.1 USB Module Level Interrupts*

There are 15 general-purpose Tx and Rx endpoints supported in addition to Endpoint 0. The Endpoint 0 Tx interrupt encapsulates both Transmit and Receive Operations.

The USB Controller will generate a corresponding EP0 or TXEPn (n=1…15) interrupt when that TX endpoint successfully completes transmitting a packet onto the bus and the endpoint buffer is empty and ready to send another packet, or when the TX endpoint had an error condition. The core will generate a corresponding RXEPn (n=1…15) interrupt when that RX endpoint successfully receives a packet and all the data is ready in the RX endpoint FIFO, or when a receive error occurs.

If the USB Controller is configured to use the DMA to handle Endpoint transactions, the Tx/Rx endpoint interrupts will only generate an interrupt on error conditions, which allows software to keep the interrupt enabled for error monitoring.

Interrupts USB[0] to USB[7] are generated by the USB controller in response to certain bus conditions. Refer to the USB Controller registers for more details on these which bus conditions trigger an interrupt. Interrupt USB[8] is generated outside the USB Controller whenever the level of DRVVBUS changes. This is usually due to entering or leaving host mode, or entering or leaving power saving mode.

In addition to the 31 Tx/Rx Controller Endpoint Interrupts, the USB Module generates 16 Tx FIFO Endpoint interrupts for each endpoint. These interrupts indicate when the TXn Endpoint FIFO is ready to accept new data.

It is important to clear the USB controller interrupts by reading the respective USB controller INTRUSB Register before clearing the IRQ_ENABLE_CLR Register and setting the IRQ_EOI Register.

**Table 25-4. Controller Interrupts**

| Interrupt | Description |
|---|---|
| TX_ENDP[15:0] | Interrupts for Transmit Endpoints 15 to 0 (TX_ENDP[0] is used for both Transmit and Receive) |
| RX_ENDP[15:1] | Interrupts for Receive Endpoints 15 to 1 |
| USB[8] DRVVBUS level change | Interrupt from the USB Module when VBUSBUS logic level changes |
| USB[7:0] | Interrupts detected by the USB Controller for Certain Bus Conditions |
| TX_FIFO[15:0] | TX FIFO endpoint ready for Endpoints 15 to 0 |

The interrupts listed in Table 25-4 can be enabled (or disabled) by setting (or clearing) the appropriate IRQENABLE_SET (or IRQENABLE_CLR) bits in the MMR registers.

The interrupt registers for all the interrupts in Table 132 are listed in both the USB0 and USB1 module memory maps. The interrupt registers for TX_ENDP[15:0], RX_ENDP[15:1], and USB[7:0] are listed in the USB Controller memory map.

#### *25.5.6.2.2 USB Controller Level Interrupts*

If the USB Controller Level interrupt mode is selected for the USB interrupts, then the USB Module interrupt sources are not valid. In this case, the USB controller interrupts will be aggregated into the module interrupt for the CPU to service. All interactions to clear and mask interrupts must be done with the USB Controller registers.

The additional USB interrupt for DRVVBUS level changes is not part of the USB controller interrupt set, and therefore is not available when using the USB Controller interrupt mode.


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.5.7 Subsystem Reset Considerations

The USB Subsystem controller can be reset by two sources: hardware reset and the soft reset.

#### 25.5.7.1 Software Reset Considerations

When the soft reset bit in the USBSS SYSCONFIG register is set, all the USB controller registers and DMA operations are reset. The bit is cleared automatically.

A software reset on the CPU does not affect the register values and operation of the USB controller.

#### 25.5.7.2 Hardware Reset Considerations

When a hardware reset is asserted, all the registers are set to their default values.

### 25.5.8 Clock, PLL, and PHY Initialization

Prior to configuring the USB Module Registers, the device USB SubSystem and PHY are required to be released from reset, provided with clocks as well as configured with appropriate settings. Not all registers related for this task are captured within this document. Please refer to the Power, Reset and Clock Manager (PRCM) for clocking and Control Module for PHY configuration registers definitions.

Note: The following initialization makes use of the software option where USB role is controlled by firmware programming USBx Mode Register.

Clock Related Configuration:

```
*0x48180B10 &= 0xFFFFFF9F; // RM_DEFAULT_RSTCTRL (Release USB Module from Reset)
while ((0x48180B14 & 0x00000060)>>5)!=0x3); // Wait until Module comes out of Reset
*0x48180B14 |= 0x00000060; // RM_DEFAULT_RSTST (Clear Reset Presence)
*0x48181400 = 0x2; // CM_ALWON_L3_SLOW_CLKSTCTRL  (Enable Interconnect Clock)
*0x48180558 = 0x2; // CM_DEFAULT_USB_CLKCTRL (Enable USB OCP Clock)
while(((*0x48180558) & 0x70000)>>16)==0x7); // Wait until USB Module is Ready
```

USB PLL Related:

```
PLL_SUBSYS_BASE  =  0x481C5000;
USB_PLL_BASE  =  PLL_SUBSYS_BASE + 0x260;
/*Output clock frequency from PLL (according to above function call) = (((OSC
frequency/(N+1))*M)/M2) = (((20MHz/(19+1))*960)/1) = 960MHz*/

*(0x481C5270) = 0x00010013; // USBPLL_M2NDIV
*(0x481C5274)  = 0x000003C0; // USBPLL_MN2DIV

/* Load M2, N2 dividers of ADPLL */
*(0x481C526C) = 0x00000001; // USBPLL_TENABLEDIV
   // Toggle to Complete Load
*(0x481C526C) = 0x00000000; // USBPLL_TENABLEDIV

/* Load M, N dividers of ADPLL */
*(0x481C5268) = 0x00000001; // USBPLL_TENABLE
   // Toggle to Complete Load
*(0x481C5268) = 0x00000000; // USBPLL_TENABLE

*(0x481C5264) = ((*(0x481C5264) & 0xff7fe3ff) | 0x200a080a); // USBPLL_CLKCTRL
/* Wait for phase and freq lock */
while (((*(0x481C5284)) & 0x00000600) != 0x00000600); // USBPLL_STATUS
```

USB PHY Related

```
// Clear bits 0,1,8,9,15,16,23
*(48140620) & = ~(0x00818303); // USB_CTRL0
// Set bits 6,7,10,13,14,17,18,19,20
*(48140620)  |= 0x001E64C0;    // USB_CTRL0
```

Firmware is now ready to continue programming the USB Controllers.


Copyright © 2011–2016, Texas Instruments Incorporated

## 25.6 Error Handling

When operating as an active USB Device, a control transfer may be aborted due to a protocol error on the USB, the USB Host prematurely ending the transfer, or if the USB Device software wishes to abort the transfer (for example, because it cannot process the request).

### 25.6.1 Error Handling as a USB Device

The USB controller automatically detects protocol errors and sends a STALL packet to the host under the following conditions:

- The USB Host sends more data during a write request than was specified in the 8 Byte Standard Request.
- The USB Host requests more data during a read request than was specified in the 8 Byte Standard Request.
- The USB Host sends more data bytes that the Maximum Packet size.
- The USB Host sends a non zero length DATA1 packet during the Status stage of a Control Transfer.

After the USB controller has sent the STALL packet to the USB Host, it automatically sets the SENTSTALL bit and generates an interrupt. When the software receives an Endpoint 0 interrupt with the SENTSTALL bit set, it should abort the current transfer, clear the SENTSTALL bit, and return to the IDLE state to await the next USB transfer from the Host.

If the host prematurely ends a transfer by entering the STATUS phase before all the data for the request has been transferred, or by sending a new SETUP packet before completing the current transfer, then the SETUPEND bit will be set and an Endpoint 0 interrupt generated. When the software receives an endpoint 0 interrupt with the SETUPEND bit set, it should abort the current transfer, set the SERV_SETUPEND bit, and return to the IDLE state. If the RXPKTRDY bit is set this indicates that the host has sent another SETUP packet and the software should then process this command.

A zero length OUT data packet is used to indicate the end of a Control transfer. In normal operation, such packets should only be received after the entire length of the device request has been transferred (i.e., after the software has set DataEnd). If, however, the host sends a zero length OUT data packet before the entire length of device request has been transferred, this signals the premature end of the transfer. In this case, the controller will automatically flush any IN token loaded by software ready for the Data phase from the FIFO and set SETUPEND bit (bit 4 of PERI_CSR0).

If the software wants to abort the current transfer, because it cannot process the command or has some other internal error, then it should set the SENDSTALL bit. The controller will then send a STALL packet to the host, set the SENTSTALL bit and generate an endpoint 0 interrupt.

### 25.6.2 Babble Event Detection as a USB Host

The USB2 standard defines a Babble as unexpected bus activity that persists beyond a specified point in a (micro)frame. The USB Controller, when operating as a USB Host, is able to detect a babble event. When a babble event is detected, the following steps are taken:

- Generate a babble interrupt (if enabled).
- De-assert the DRVVBUS signal which signals the VBUS charge pump to discontinue supplying 5V to VBUS.
- Go to the IDLE state, which indicates that no USB session is in progress.

  Software should restart the USB controller when a babble event/interrupt is detected.


Copyright © 2011–2016, Texas Instruments Incorporated

## 25.7 Protocol Description(s)

This section describes the functionally implementation of the USB protocol by the USB modules.

### *25.7.1 USB Device Operation*

This section describes how the USB Controller should be as when operating as a USB Device

#### 25.7.1.1 Control Transfers

Endpoint 0 is the main control endpoint of a USB Device. Software is required to handle all the standard USB Device Requests that may be sent or received via Endpoint 0. These are described in Universal Serial Bus Specification, Revision 2.0, Chapter 9. The protocol for these device requests involves a different number of sequences of transactions per USB Request. To accommodate this, the software needs to take a state machine approach to correctly decode and handle all Control Transactions on Endpoint 0.

The Standard USB Device Requests received by a USB Device can be divided into three categories: Zero Data Requests (in which all the information is included in the 8 byte request, Write Requests (in which the request will be followed by additional data, and Read Requests (in which the device is required to send data back to the USB Host).

This section looks at the sequence of actions that the software must perform to process these different types of device request.

##### *25.7.1.1.1 Zero Data Requests*

Zero data requests have all their information contained within the 8-byte request and require no additional data to be transferred. Examples of Zero Data standard device requests are:

- SET_FEATURE
- CLEAR_FEATURE
- SET_ADDRESS
- SET_CONFIGURATION
- SET_INTERFACE

The sequence of events will begin, as with all requests, when the software receives an Endpoint 0 interrupt. The RXPKTRDY bit of the Endpoint 0 Peripheral Control / Status Register (PERI_CSR0) will be automatically set by the controller. The 8-byte request should then be read from the Endpoint 0 FIFO, decoded and the appropriately handled. After the request is handled, the CPU should set the SERV_RXPKTRDY bit of PERI_CSR0 to indicate to the controller that the command has been read from the FIFO, and also set the DATAEND bit to indicate to the controller that no further data is expected for this request. The interval between setting SERV_RXPKTRDY bit and DATAEND bit should be very small to avoid getting a SETUPEND error condition. It is highly recommended to set both bits at the same time.

The USB Host will then transition to the status stage of the control transfer, a second Endpoint 0 interrupt will be generated to indicate that the request has successfully completed. No further action is required from the software. The second interrupt is just a confirmation that the request completed successfully.

If the command is an unrecognized command, or for some other reason cannot be executed, then when it has been decoded, the CPU should write to the PERI_CSR0 register to set the SERV_RXPKTRDY bit and to set the SENDSTALL bit. When the USB Host moves to the status stage of the request, the controller will send a STALL packet telling the host that the request was not executed. A second Endpoint 0 interrupt will be generated and the SENTSTALL bit will be automatically set by the controller.

If the host sends more data after the DATAEND bit has been set, then the controller will send a STALL packet automatically. An Endpoint 0 interrupt will be generated and the SENTSTALL bit will be set.

**NOTE:** The DMA cannot service Endpoint 0 transactions; Endpoint 0 must always be serviced via a CPU.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.1.1.2 Write Requests

Write requests involve an additional packet(s) of data being sent from the USB Host after the 8-byte USB request has been received. An example of a Write request is: SET_DESCRIPTOR.

The sequence of events will begin, as with the Zero Data Request, when the software receives an Endpoint 0 interrupt. The RXPKTRDY bit of the Endpoint 0 Peripheral Control / Status Register (PERI_CSR0) will have been automatically set by the controller. The 8-byte request should then be read from the Endpoint 0 FIFO, decoded and the appropriately handled. After the request is handled, the CPU should set the SERV_RXPKTRDY bit of PERI_CSR0 to indicate to the controller that the command has been read from the FIFO, however for write requests, the DATAEND bit should *not* be set by the user indicating that more data is expected from the USB Host.

When a second endpoint 0 interrupt is received, the PERI_CSR0 register should be read to check the endpoint status. The RXPKTRDY bit of PERI_CSR0 should be set to indicate that a data packet has been received. The COUNT0 register should then be read to determine the size of this data packet. The data packet can then be read from the endpoint 0 FIFO and stored into memory

If the length of the data associated with the request (indicated by the wLength field in the command) is greater than the maximum packet size for endpoint 0, further data packets are expected be sent. Again, PERI_CSR0 should be written to set the SERV_RXPKTRDY bit, but the DATAEND bit should not be set. When all the expected data packets have been received, the PERI_CSR0 register should be written to set the SERV_RXPKTRDY bit and to set the DATAEND bit .

The USB Host will then transition to the status stage of the control transfer, a second Endpoint 0 interrupt will be generated to indicate that the request has successfully completed. No further action is required from the software. The second interrupt is just a confirmation that the request completed successfully.

If the host sends more data after the DATAEND has been set, then the controller will automatically send a STALL. An endpoint 0 interrupt will be generated and the SENTSTALL bit of PERI_CSR0 will be set.

#### 25.7.1.1.3 Read Requests

Read requests have a packet (or packets) of data sent from the function to the host after the 8-byte command. Examples of Read Standard Device Requests are:

- GET_CONFIGURATION
- GET_INTERFACE
- GET_DESCRIPTOR
- GET_STATUS
- SYNCH_FRAME

The sequence of events will begin, as with the Zero Data Request, when the software receives an Endpoint 0 interrupt. The RXPKTRDY bit of the Endpoint 0 Peripheral Control / Status Register (PERI_CSR0) will have been automatically set by the controller. The 8-byte request should then be read from the Endpoint 0 FIFO, decoded and the appropriately handled. After the request is handled, the CPU should set the SERV_RXPKTRDY bit of PERI_CSR0 to indicate to the controller that the command has been read from the FIFO, however for read requests, the DATAEND bit should *not* be set by the user indicating that data is expected by the USB Host.

The data to be sent to the host should then be written to the endpoint 0 FIFO. If the data to be sent is greater than the maximum packet size for endpoint 0, only the maximum packet size should be written to the FIFO. The PERI_CSR0 register should then be written to set the TXPKTRDY bit to indicate to the controller that there is a packet in the FIFO to be sent. When the packet has been sent to the host, another endpoint 0 interrupt will be generated and the next data packet can be written to the FIFO.

When the last data packet has been written to the FIFO, the PERI_CSR0 register should be written to set the TXPKTRDY bit and to set the DATAEND bit indicating to the controller that no additional packets are to be send to the USB Host.

When the host moves to the status stage of the request, another endpoint 0 interrupt will be generated to indicate that the request has completed. No further action is required from the software: the interrupt is just a confirmation that the request completed successfully.


Copyright © 2011–2016, Texas Instruments Incorporated

If the host requests more data after DATAEND (bit 3) has been set, then the controller will send a STALL. An endpoint 0 interrupt will be generated and the SENTSTALL bit of PERI_CSR0 will be set.

#### *25.7.1.1.4 Endpoint 0 Interrupt Servicing*

When the USB controller is operating as a peripheral device, the Endpoint 0 control needs three modes – IDLE, TX and RX – corresponding to the different phases of the control transfer and the states endpoint 0 enters for the different phases of the transfer (described in later sections).

The default mode on power-up or reset should be IDLE. RXPKTRDY bit of PERI_CSR0 becoming set when endpoint 0 is in IDLE state indicates a new device request. Once the device request is unloaded from the FIFO, the controller decodes the descriptor to find whether there is a data phase and, if so, the direction of the data phase of the control transfer (in order to set the FIFO direction). See Figure 25-2.

Depending on the direction of the data phase, endpoint 0 goes into either TX state or RX state. If there is no Data phase, endpoint 0 remains in IDLE state to accept the next device request

The actions that the CPU needs to take at the different phases of the possible transfers (for example, loading the FIFO, setting TXPKTRDY) are indicated in Figure 25-3.

**Figure 25-2. CPU Actions at Transfer Phases**

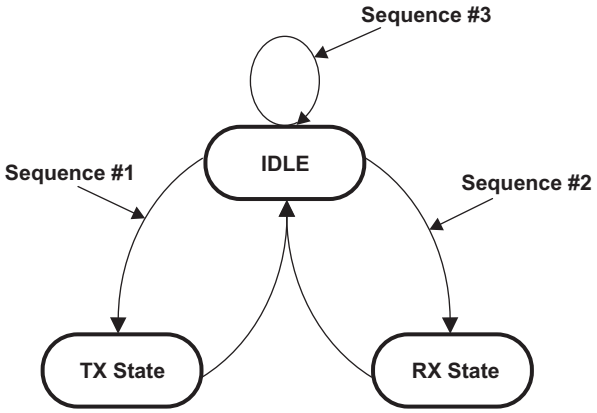
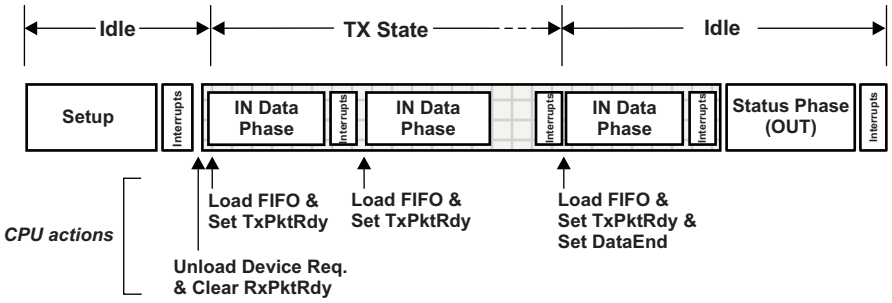



**Figure 25-3. TX State**





Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 25-4. RX State**

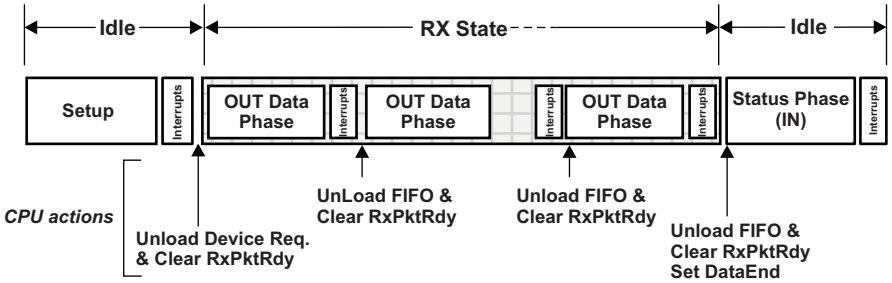
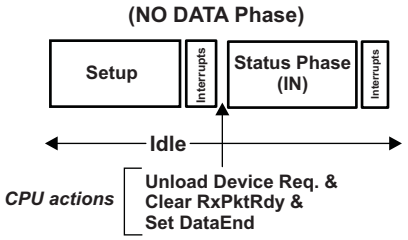



**Figure 25-5. No Data Phase**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.1.1.4.1 Endpoint 0 Interrupt Generating Events

An Endpoint 0 interrupt is generated when any of the below events occur and Interrupt Service handler should determine the right event and launch the proper handler

The following events will cause Endpoint 0 Interrupt to be generated while the controller is operating assuming the role of a USB Device

- The controller sets the RXPKTRDY bit after a valid token has been received and data has been written to the FIFO (legal status condition).
- The controller clears the TXPKTRDY bit after the packet of data in the FIFO has been successfully transmitted to the USB host (legal status condition).
- The controller sets the SENTSTALL bit after a control transaction is ended due to a protocol violation (illegal status condition).
- The controller sets the SETUPEND bit because a control transfer has ended before DATAEND (bit 3 of PERI_CSR0) is set (illegal status condition).

In order to determine the cause of the interrupt, the detail below recommends the order of cause determination and execution.

Whenever the endpoint 0 service routine is entered, the software must first check to see if the current control transfer has been ended due to either a STALL condition or a premature end of control transfer. If the control transfer ends due to a STALL condition, the SENTSTALL bit would be set. If the control transfer ends due to a premature end of control transfer, the SETUPEND bit would be set. In either case, the software should abort processing the current control transfer and set the state to IDLE.

Once the software has determined that the interrupt was not generated by an illegal bus state, the next action taken depends on the endpoint state.

If endpoint 0 is in IDLE state, the only valid reason an interrupt can be generated is as a result of the controller receiving data from the bus. The service routine must check for this by testing the RXPKTRDY bit of PERI_CSR0 (bit 0). If this bit is set, then the controller has received a SETUP packet. This must be unloaded from the FIFO and decoded to determine the action the controller must take. Depending on the command contained within the SETUP packet, endpoint 0 will enter one of three states:

- If the command is a single packet transaction (SET_ADDRESS, SET_INTERFACE etc.) without any data phase, the endpoint will remain in IDLE state.
- If the command has an OUT data phase (SET_DESCRIPTOR etc.), the endpoint will enter RX state.
- If the command has an IN data phase (GET_DESCRIPTOR etc.), the endpoint will enter TX state.
- If the endpoint 0 is in TX state, the interrupt indicates that the core has received an IN token and data from the FIFO has been sent. The software must respond to this either by placing more data in the FIFO if the host is still expecting more data or by setting the DATAEND bit to indicate that the data phase is complete. Once the data phase of the transaction has been completed, endpoint 0 should be returned to IDLE state to await the next control transaction (status stage).
- If the endpoint is in RX state, the interrupt indicates that a data packet has been received. The software must respond by unloading the received data from the FIFO. The software must then determine whether it has received all of the expected data. If it has, the software should set the DATAEND bit and return endpoint 0 to IDLE state. If more data is expected, the firmware should set the SERV_RXPKTRDY bit of PERI_CSR0 (bit 6) to indicate that it has read the data in the FIFO and leave the endpoint in RX state.

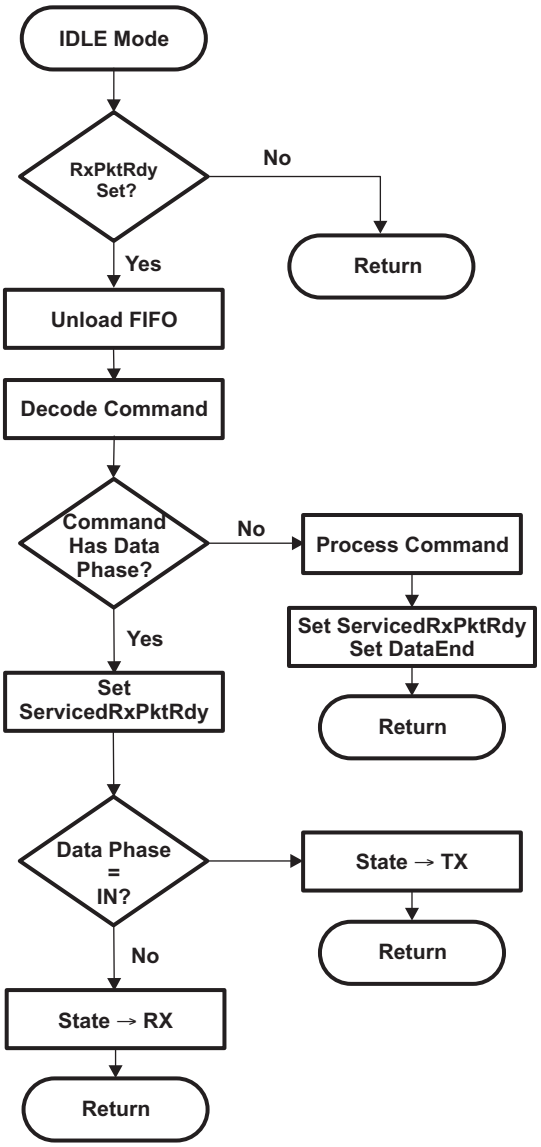

Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.1.1.4.2 Endpoint 0 Idle State

Idle Mode needs to be selected by the Endpoint 0 control after power-on or reset and is the mode in which the Endpoint 0 control should return when the Rx and Tx modes are terminated.

It is also the mode in which the SETUP phase of control transfer is handled (as outlined in Figure 25-6).

**Figure 25-6. Flow Chart of Setup Stage of a Control Transfer in Peripheral Mode**






Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.1.1.4.3 Endpoint 0 TX State

When the endpoint is in TX state, all arriving IN tokens need to be treated as part of a data phase until the required amount of data has been sent to the host. If either a SETUP or an OUT token is received while the endpoint is in the TX state, this will cause a SETUPEND condition to occur as the core expects only IN tokens.

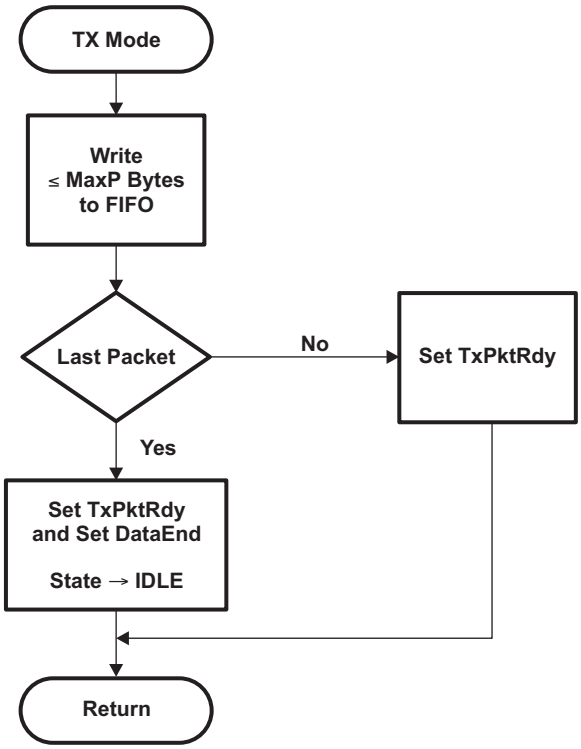
Three events can cause TX mode to be terminated before the expected amount of data has been sent as shown in Figure 25-7:

1. The host sends an invalid token causing a SETUPEND condition (bit 4 of PERI_CSR0 is set).
2. The firmware sends a packet containing less than the maximum packet size for Endpoint 0.
3. The firmware sends an empty data packet.

Until the transaction is terminated, the firmware simply needs to load the FIFO when it receives an interrupt which indicates that a packet has been sent from the FIFO. An interrupt is generated when TXPKTRDY is cleared.

When the firmware forces the termination of a transfer (by sending a short or empty data packet), it should set the DATAEND bit of PERI_CSR0 (bit 3) to indicate to the core that the data phase is complete and that the core should next receive an acknowledge packet.

**Figure 25-7. Flow Chart of Transmit Data Stage of a Control Transfer in Peripheral Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.1.1.4.4 Endpoint 0 Rx State

In RX mode, all arriving data should be treated as part of a data phase until the expected amount of data has been received. If either a SETUP or an IN token is received while the endpoint is in RX state, a SETUPEND condition will occur as the controller expects only OUT tokens.

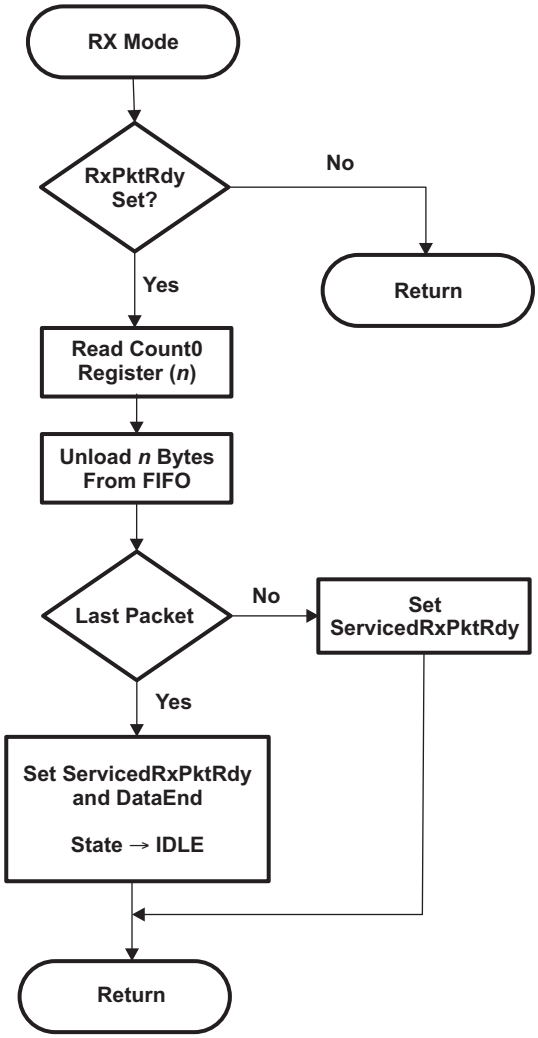
Three events can cause RX mode to be terminated before the expected amount of data has been received as shown in Figure 25-8:

1. The host sends an invalid token causing a SETUPEND condition (setting bit 4 of PERI_CSR0).
2. The host sends a packet which contains less than the maximum packet size for endpoint 0.
3. The host sends an empty data packet.

Until the transaction is terminated, the software unloads the FIFO when it receives an interrupt that indicates new data has arrived (setting RXPKTRDY bit of PERI_CSR0) and to clear RXPKTRDY by setting the SERV_RXPKTRDY bit of PERI_CSR0 (bit 6).

When the software detects the termination of a transfer (by receiving either the expected amount of data or an empty data packet), it should set the DATAEND bit (bit 3 of PERI_CSR0) to indicate to the controller that the data phase is complete and that the core should receive an acknowledge packet next.

**Figure 25-8. Flow Chart of Receive Data Stage of a Control Transfer in Peripheral Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

### 25.7.1.2 Bulk Transfers

Bulk transactions are handled by Endpoints other than Endpoint 0. Bulk Transfers are used to handle non-periodic, large bursty communication typically used for a transfer that use any available bandwidth and can also be delayed until bandwidth is available.

The following optional features are available for use with a Tx endpoint used in peripheral mode for Bulk IN transactions:

- Double packet buffering: When enabled, up to two sequential packets can be stored in the Endpoint FIFO awaiting transmission to the host. Double packet buffering is enabled by setting the DPB bit of TXFIFOSZ register
- DMA: If DMA is enabled for the Bulk Endpoint, a DMA request will be generated whenever the endpoint is able to accept another packet in its FIFO. This feature allows the DMA controller to load packets into the FIFO without processor intervention

#### *25.7.1.2.1 Bulk Transfer Setup*

In configuring an Endpoint for Bulk Transfers, the firmware must properly configure the USB Controller Registers for proper operation. The Max Packet Size Register (RX/TXMAXP) register must be written with the maximum packet size (in bytes) for the Endpoint. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the Endpoint. Additionally the Control / Status Register for the respective Endpoint(s) must be appropriately configured.

For IN Transactions, Endpoint Control Status Register for Transmit Endpoint (PERI_TXCSR) should be configured for desired operation. If the DMA is to be used to service the endpoint, DMAEN and DMAMODE bit fields should be set, else left cleared Table 25-5 displays the PERI_TXCSR setting when used for Bulk transfer.

When the endpoint is first configured (following a SET_CONFIGURATION or SET_INTERFACE command on Endpoint 0), the lower byte of PERI_TXCSR should be written to set the CLRDATATOG bit (bit 6). This will ensure that the data toggle (which is handled automatically by the controller) starts in the correct state.

Also if there are any data packets in the FIFO, indicated by the FIFONOTEMPTY bit being set, they should be flushed by setting the FLUSHFIFO bit

**NOTE:** It may be necessary to set this bit twice in succession if double buffering is enabled.

**Table 25-5. PERI_TXCSR Register Bit Configuration for Bulk IN Transactions**

| Bit Field | Bit Name | Description |
|---|---|---|
| Bit 15 | AUTOSET | Cleared to 0 if using DMA. For CPU Mode set to 1. if AUTOSET bit is set, the TXPKTRDY bit will be automatically set when data of the maximum packet size is loaded into the FIFO, else set manually. |
| Bit 14 | ISO | Cleared to 0 for Bulk Transfers. |
| Bit 13 | MODE | Set to 1 to make sure the FIFO is enabled (only necessary if the FIFO is shared with an RX endpoint) |
| Bit 12 | DMAEN | Set to 1 to enable DMA usage. Cleared to 0 if a CPU is being used to service the Tx Endpoint |
| Bit 11 | FRCDATATOG | Cleared to 0 to allow normal data toggle operations. |
| Bit 10 | DMAMODE | Set to 1 when DMA is used to service Tx FIFO. |
| Bit 9 | Reserved | Reserved |
| Bit 8 | Reserved | Reserved |
| Bit 7 | Reserved | Reserved |
| Bit 6 | ClrDataTog | This bit should be cleared to 0 by the CPU prior to Bulk Endpoint Operation |
| Bit 5 | SentStall | This bit should be cleared to 0 by the CPU prior to Bulk Endpoint Operation |
| Bit 4 | SendStall | This bit should be cleared to 0 by the CPU prior to Bulk Endpoint Operation |
| Bit 3 | FlushFIFO | This bit is set when the FIFO is non-Empty. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-5. PERI_TXCSR Register Bit Configuration for Bulk IN Transactions (continued)**

| Bit Field | Bit Name | Description |
|---|---|---|
| Bit 2 | UnderRun | This bit should be cleared to 0 by the CPU prior to Bulk Endpoint Operation |
| Bit 1 | FIFONotEmpty | This bit should be cleared to 0 by the CPU prior to Bulk Endpoint Operation |
| Bit 0 | TxPktRdy | This bit should be cleared to 0 by the CPU prior to Bulk Endpoint Operation |

When using DMA for OUT Endpoints, the PERI_RXCSR needs to have its DMAEN bit set and have its AUTOCLEAR and DMAMODE bits cleared. In addition, the relevant interrupt enable bit in the INTRRXE register should be set (if an interrupt is required for this endpoint) and the PERI_RXCSR register should be set as shown in Table 25-6

Also if there are any data packets in the FIFO (indicated by the RXPKTRDY bit (bit 0 of PERI_RXCSR) being set), they should be flushed by setting the FLUSHFIFO bit (bit 4 of PERI_RXCSR).

**NOTE:** It may be necessary to set this bit twice in succession if double buffering is enabled.

**Table 25-6. PERI_RXCSR Register Bit Configuration for Bulk OUT Transactions**

| Bit Field | Bit Name | Description |
|---|---|---|
| Bit 15 | AUTOCLEAR | Cleared to 0 if using DMA. In CPU Mode of usage, if the CPU sets AUTOCLEAR bit, the RXPKTRDY bit will be automatically cleared when a packet of RXMAXP bytes has been unloaded from the Receive FIFO. |
| Bit 14 | ISO | Cleared to 0 to enable Bulk protocol. |
| Bit 13 | DMAEN | Set to 1 if a DMA request is required for this Rx endpoint. |
| Bit 12 | DISNYET | Cleared to 0 to allow normal PING flow control. This will affect only high speed transactions. |
| Bit 11 | DMAMODE | Always clear this bit to 0. |

When the Rx endpoint is first configured (following a SET_CONFIGURATION or SET_INTERFACE command on Endpoint 0), the lower byte of PERI_RXCSR should be written to set the CLRDATATOG bit (bit 7). This will ensure that the data toggle (which is handled automatically by the USB controller) starts in the correct state.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.1.2.2 Bulk IN Transfers

When data is to be transferred over a Bulk IN pipe, a data packet needs to be loaded into the FIFO and the PERI_TXCSR register written to set the TXPKTRDY bit. The TxPktRDY bit will remain set until the packet has been sent to the USB Host and will be automatically cleared by the USB controller. An Endpoint(n) interrupt will also generated to signify to the firmware that the next packet can be loaded into the FIFO. If double packet buffering is enabled, then after the first packet has been loaded and the TXPKTRDY bit set, the TXPKTRDY bit will immediately be cleared by the USB controller and an interrupt generated so that a second packet can be loaded into the FIFO. The software should always load a packet into the Endpoint FIFO when it receives an interrupt, regardless of whether double packet buffering is enabled or not.

In the general case, the packet size must not exceed the size specified by the lower 11 bits of the TXMAXP register. This part of the register defines the payload (packet size) for transfers over the USB and is required by the USB Specification to be either 8, 16, 32, 64 (Full-Speed or High-Speed) or 512 bytes (High-Speed only).

The host may determine that all the data for a transfer has been sent by knowing the total amount of data that is expected. Alternatively it may infer that all the data has been sent when it receives a packet which is smaller than the maximum packet size configuration for that particular endpoint (TXMAXP[10-0]). In the latter case, if the total size of the data block is a multiple of this payload, it will be necessary for the function to send a null packet after all the data has been sent. This is done by setting TXPKTRDY when the next interrupt is received, without loading any data into the FIFO.

#### 25.7.1.2.3 Bulk OUT Transfers

When a data packet is received by a Bulk OUT pipe, the RXPKTRDY bit is set and an interrupt is generated. The software should read the RXCOUNT register for the endpoint to determine the size of the data packet. The data packet should be read from the FIFO, and then the RXPKTRDY bit should be cleared.

The packets received should not exceed the size specified in the RXMAXP register (as this should be the value set in the wMaxPacketSize field of the endpoint descriptor sent to the host). When a block of data larger than wMaxPacketSize needs to be sent to the function, it will be sent as multiple packets. All the packets will be wMaxPacketSize in size, except the last packet which will contain the residue. The software may use an application specific method of determining the total size of the block and hence when the last packet has been received. Alternatively it may infer that the entire block has been received when it receives a packet which is less than wMaxPacketSize in size. (If the total size of the data block is a multiple of wMaxPacketSize, a null data packet will be sent after the data to signify that the transfer is complete.)


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.1.2.4 Error Handling of Bulk ransfer

If the software wants to shut down the Bulk IN pipe, it should set the SENDSTALL bit (bit 4 of PERI_TXCSR). When the controller receives the next IN token, it will send a STALL to the host, set the SENTSTALL bit (bit 5 of PERI_TXCSR) and generate an interrupt.

When the software receives an interrupt with the SENTSTALL bit (bit 5 of PERI_TXCSR) set, it should clear the SENTSTALL bit. It should however leave the SENDSTALL bit set until it is ready to re-enable the Bulk IN pipe.

**NOTE:** If the host failed to receive the STALL packet for some reason, it will send another IN token, so it is advisable to leave the SENDSTALL bit set until the software is ready to re-enable the Bulk IN pipe. When a pipe is re-enabled, the data toggle sequence should be restarted by setting the CLRDATATOG bit in the PERI_TXCSR register (bit 6).

If the software wants to shut down the Bulk OUT pipe, it should set the SENDSTALL bit (bit 5 of PERI_RXCSR). When the controller receives the next packet it will send a STALL to the host, set the SENTSTALL bit (bit 6 of PERI_RXCSR) and generate an interrupt.

When the software receives an interrupt with the SENTSTALL bit (bit 6 of PERI_RXCSR) set, it should clear this bit. It should however leave the SENDSTALL bit set until it is ready to re-enable the Bulk OUT pipe.

**NOTE:** If the host failed to receive the STALL packet for some reason, it will send another packet, so it is advisable to leave the SENDSTALL bit set until the software is ready to re-enable the Bulk OUT pipe. When a Bulk OUT pipe is re-enabled, the data toggle sequence should be restarted by setting the CLRDATATOG bit (bit 7) in the PERI_RXCSR register.

### 25.7.1.3 Interrupt Transfer

An Interrupt IN transaction uses the same protocol as a Bulk IN transaction and the configuration and operation mentioned on prior sections apply to Interrupt transfer too with minor changes. Similarly, an Interrupt OUT transaction uses almost the same protocol as a Bulk OUT transaction and can be used the same way.

Tx endpoints in the USB controller have one feature for Interrupt IN transactions that they do not support in Bulk IN transactions. In Interrupt IN transactions, the endpoints support continuous toggle of the data toggle bit.

This feature is enabled by setting the FRCDATATOG bit in the PERI_TXCSR register (bit 11). When this bit is set, the controller will consider the packet as having been successfully sent and toggle the data bit for the endpoint, regardless of whether an ACK was received from the host.

Another difference is that interrupt endpoints do not support PING flow control. This means that the controller should never respond with a NYET handshake, only ACK/NAK/STALL. To ensure this, the DISNYET bit in the PERI_RXCSR register (bit 12) should be set to disable the transmission of NYET handshakes in high-speed mode.

Though DMA can be used with an interrupt OUT endpoint, it generally offers little benefit as interrupt endpoints are usually expected to transfer all their data in a single packet.


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.7.1.4 Isochronous Transfer

Isochronous transfers are used when working with isochronous data. Isochronous transfers provide periodic, continuous communication between host and device.

#### *25.7.1.4.1 Isochronous Transfer Setup*

In configuring a Tx endpoint for Isochronous IN transactions, the TXMAXP register must be written with the maximum packet size (in bytes) for the endpoint. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the endpoint unless it is a high bandwidth ISO. In this case, TXMAXP should be an integer multiple of 1024. In addition, the relevant interrupt enable bit in the INTRTXE register should be set (if an interrupt is required for this endpoint) and the PERI_TXCSR register should be set as shown in Table 25-7.

**Table 25-7. PERI_TXCSR Register Bit Configuration for Isochronous IN Transactions**

| Bit Field | Bit Name | Description |
|---|---|---|
| Bit 14 | ISO | Set to 1 to enable Isochronous transfer protocol. |
| Bit 13 | MODE | Set to 1 to ensure the FIFO is enabled (only necessary if the FIFO is shared with an Rx endpoint). |
| Bit 12 | DMAEN | Set to 1 if DMA Requests have to be enabled. |
| Bit 11 | FRCDATATOG | Ignored in Isochronous mode. |
| Bit 10 | DMAMODE | Set to 1 when DMA is enabled. |

In configuring an Rx endpoint for Isochronous OUT transactions, the RXMAXP register must be written with the maximum packet size (in bytes) for the endpoint. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the endpoint unless it is a high bandwidth ISO. In this case, RXMAXP should be an integer multiple of 1024. In addition, the relevant interrupt enable bit in the INTRRXE register should be set (if an interrupt is required for this endpoint) and the PERI_RXCSR register should be set as shown in Table 25-8.

**Table 25-8. PERI_RXCSR Register Bit Configuration for Isochronous OUT Transactions**

| Bit Field | Bit Name | Description |
|---|---|---|
| Bit 15 | AUTOCLEAR | Cleared to 0 if using DMA. In CPU Mode of usage, if the CPU sets AUTOCLEAR bit, the RXPKTRDY bit will be automatically cleared when a packet of RXMAXP bytes has been unloaded from the Receive FIFO. |
| Bit 14 | ISO | Set to 1 to configure endpoint usage for Isochronous transfer. |
| Bit 13 | DMAEN | Set to 1 if a DMA request is required for this Rx endpoint. |
| Bit 12 | DISNYET | Ignored in Isochronous Mode. |
| Bit 11 | DMAMODE | Always clear this bit to 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.1.4.2 Isochronous IN Transfers

The following optional features are available for use with a Tx endpoint used in Peripheral mode for Isochronous IN transactions:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO awaiting transmission to the host. Double packet buffering is enabled by setting the DPB bit of TXFIFOSZ register (bit 4).

  **NOTE:** Double packet buffering is generally advisable for isochronous transactions in order to avoid underrun errors as described in later section.

- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint is able to accept another packet in its FIFO. This feature allows the DMA controller to load packets into the FIFO without processor intervention.
  However, this feature is not particularly useful with Isochronous endpoints because the packets transferred are often not maximum packet size and the PERI_TXCSR register needs to be accessed following every packet to check for Underrun errors.
  When DMA is enabled and DMAMODE bit of PERI_TXCSR is set, endpoint interrupt will not be generated for completion of packet transfer. Endpoint interrupt will be generated only in the error conditions.

  An isochronous endpoint does not support data retries, so if data underrun is to be avoided, the data to be sent to the host must be loaded into the FIFO before the IN token is received. The host will send one IN token per frame (or microframe in High-speed mode), however the timing within the frame (or microframe) can vary. If an IN token is received near the end of one frame and then at the start of the next frame, there will be little time to reload the FIFO. For this reason, double buffering of the endpoint is usually necessary.

  An interrupt is generated whenever a packet is sent to the host and the software may use this interrupt to load the next packet into the FIFO and set the TXPKTRDY bit in the PERI_TXCSR register (bit 0) in the same way as for a Bulk Tx endpoint. As the interrupt could occur almost any time within a frame(/microframe), depending on when the host has scheduled the transaction, this may result in irregular timing of FIFO load requests. If the data source for the endpoint is coming from some external hardware, it may be more convenient to wait until the end of each frame(/microframe) before loading the FIFO as this will minimize the requirement for additional buffering. This can be done by using either the SOF interrupt or the external SOF_PULSE signal from the controller to trigger the loading of the next data packet. The SOF_PULSE is generated once per frame(/microframe) when a SOF packet is received. (The controller also maintains an external frame(/microframe) counter so it can still generate a SOF_PULSE when the SOF packet has been lost.) The interrupts may still be used to set the TXPKTRDY bit in PERI_TXCSR (bit 0) and to check for data overruns/underruns.

  Starting up a double-buffered Isochronous IN pipe can be a source of problems. Double buffering requires that a data packet is not transmitted until the frame(/microframe) after it is loaded. There is no problem if the function loads the first data packet at least a frame(/microframe) before the host sets up the pipe (and therefore starts sending IN tokens). But if the host has already started sending IN tokens by the time the first packet is loaded, the packet may be transmitted in the same frame(/microframe) as it is loaded, depending on whether it is loaded before, or after, the IN token is received. This potential problem can be avoided by setting the ISOUPDATE bit in the POWER register (bit 7). When this bit is set, any data packet loaded into an Isochronous Tx endpoint FIFO will not be transmitted until after the next SOF packet has been received, thereby ensuring that the data packet is not sent too early.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.1.4.3 Isochronous OUT Transfers

An isochronous OUT transaction is used to transfer periodic data from the host to the function controller.

Following optional features are available for use with an Rx endpoint used in Peripheral mode for Isochronous OUT transactions:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO on reception from the host. Double packet buffering is enabled by setting the DPB bit of RXFIFOSZ register (bit 4). **NOTE:** Double packet buffering is generally advisable for Isochronous transactions in order to avoid overrun errors.
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint has a packet in its FIFO. This feature can be used to allow the DMA controller to unload packets from the FIFO without processor intervention.
  However, this feature is not particularly useful with Isochronous endpoints because the packets transferred are often not maximum packet size and the PERI_RXCSR register needs to be accessed following every packet to check for Overrun or CRC errors.
  When DMA is enabled, endpoint interrupt will not be generated for completion of packet reception. Endpoint interrupt will be generated only in the error conditions.

  An Isochronous endpoint does not support data retries so, if a data overrun is to be avoided, there must be space in the FIFO to accept a packet when it is received. The host will send one packet per frame (or microframe in High-speed mode); however, the time within the frame can vary. If a packet is received near the end of one frame(/microframe) and another arrives at the start of the next frame, there will be little time to unload the FIFO. For this reason, double buffering of the endpoint is usually necessary.

  An interrupt is generated whenever a packet is received from the host and the software may use this interrupt to unload the packet from the FIFO and clear the RXPKTRDY bit in the PERI_RXCSR register (bit 0) in the same way as for a Bulk Rx endpoint. As the interrupt could occur almost any time within a frame(/microframe), depending on when the host has scheduled the transaction, the timing of FIFO unload requests will probably be irregular. If the data sink for the endpoint is going to some external hardware, it may be better to minimize the requirement for additional buffering by waiting until the end of each frame(/microframe) before unloading the FIFO. This can be done by using either the SOF interrupt or the external SOF_PULSE signal from the controller to trigger the unloading of the data packet. The SOF_PULSE is generated once per frame(/microframe) when a SOF packet is received. (The controller also maintains an external frame(/microframe) counter so it can still generate a SOF_PULSE when the SOF packet has been lost.) The interrupts may still be used to clear the RXPKTRDY bit in PERI_RXCSR and to check for data overruns/underruns.

#### 25.7.1.4.4 Isochronous Transfer Error Handling

If the endpoint has no data in its FIFO when an IN token is received, it will send a null data packet to the host and set the UNDERRUN bit in the PERI_TXCSR register (bit 2). This is an indication that the software is not supplying data fast enough for the host. It is up to the application to determine how this error condition is handled.

If the software is loading one packet per frame(/microframe) and it finds that the TXPKTRDY bit in the PERI_TXCSR register (bit 0) is set when it wants to load the next packet, this indicates that a data packet has not been sent (perhaps because an IN token from the host was corrupted). It is up to the application how it handles this condition: it may choose to flush the unsent packet by setting the FLUSHFIFO bit in the PERI_TXCSR register (bit 3), or it may choose to skip the current packet.

If there is no space in the FIFO to store a packet when it is received from the host, the OVERRUN bit in the PERI_RXCSR register (bit 2) will be set. This is an indication that the software is not unloading data fast enough for the host. It is up to the application to determine how this error condition is handled

If the controller finds that a received packet has a CRC error, it will still store the packet in the FIFO and set the RXPKTRDY bit (bit 0 of PERI_RXCSR) and the DATAERROR bit (bit 3 of PERI_RXCSR). It is left up to the application to determine how this error condition is handled.


Copyright © 2011–2016, Texas Instruments Incorporated

The number of USB packets sent in any microframe will depend on the amount of data to be transferred, and is indicated through the PIDs used for the individual packets. If the indicated number of packets have not been received by the end of a microframe, the INCOMPRX bit (bit 8) bit in the PERI_RXCSR register will be set to indicate that the data in the FIFO is incomplete. Equally, if a packet of the wrong data type is received, then the PID Error bit is the PERI_RXCSR register will be set. In each case, an interrupt will, however, still be generated to allow the data that has been received to be read from the FIFO.

**Note:** The circumstances in which a PID Error or INCOMPRX is reported depends on the precise sequence of packets received.

When the core is operating in peripheral mode, the details are in Table 25-9.

**Table 25-9. Isochronous OUT Error Handling: Peripheral Mode**

| No. Packet(s) Expected | Data Packet(s) Received | Response |
|---|---|---|
| 1 | DATA0 | OK |
| | DATA1 | PID Error set |
| | DATA2 | PID Error set |
| | MDATA | PID Error set |
| 2 | DATA0 | OK |
| | DATA1 | INCOMPRX Set |
| | DATA2 | INCOMPRX Set + PID Error set |
| | MDATA | INCOMPRX Set |
| | MDATA DATA0 | PID Error Set |
| | MDATA DATA1 | OK |
| | MDATA DATA2 | PID Error Set |
| | MDATA MDATA | PID Error Set |
| 3 | DATA0 | OK |
| | DATA1 | INCOMPRX Set |
| | DATA2 | INCOMPRX Set |
| | MDATA | INCOMPRX Set |
| | MDATA DATA0 | PID Error set |
| | MDATA DATA1 | OK |
| | MDATA DATA2 | INCOMPRX Set |
| | MDATA MDATA | INCOMPRX Set |
| | MDATA MDATA DATA0 | PID Error set |
| | MDATA MDATA DATA1 | PID Error set |
| | MDATA MDATA DATA2 | OK |
| | MDATA MDATA MDATA | PID Error set |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.7.2 USB Host Operation

This section describes how the USB Controller should be as when operating as a USB Host

#### 25.7.2.1 Control Transfers

Host control transactions are conducted through Endpoint 0. Firmware is required to handle all the Standard USB Device Requests that may be sent or received via Endpoint 0. Endpoint 0 cannot be serviced by the DMA.

There are three categories of Standard Device Requests to be handled: Zero Data Requests (in which all the information is included in the request); Write Requests (in which the request will be followed by additional data); and Read Requests (in which the device is required to send data back to the host).

1. Zero Data Requests comprise a SETUP Token packet followed by an IN Status Stage.
2. Write Requests comprise a SETUP Token packet followed by an OUT Data Phase which is in turn followed by an IN Status Phase.
3. Read Requests comprise a SETUP Token packet, followed by an IN Data Phase which is in turn followed by an OUT Status Phase.

A timeout may be set to limit the length of time for which the host controller will retry a transaction that is continually NAKed by the device. This limit can be between 2 and 2.15 (micro)frames and is set through the NAKLIMIT0 register.

The following sections describe the actions that the CPU needs to take in issuing these different types of request through looking at the steps to take in the different phases of a control transaction.

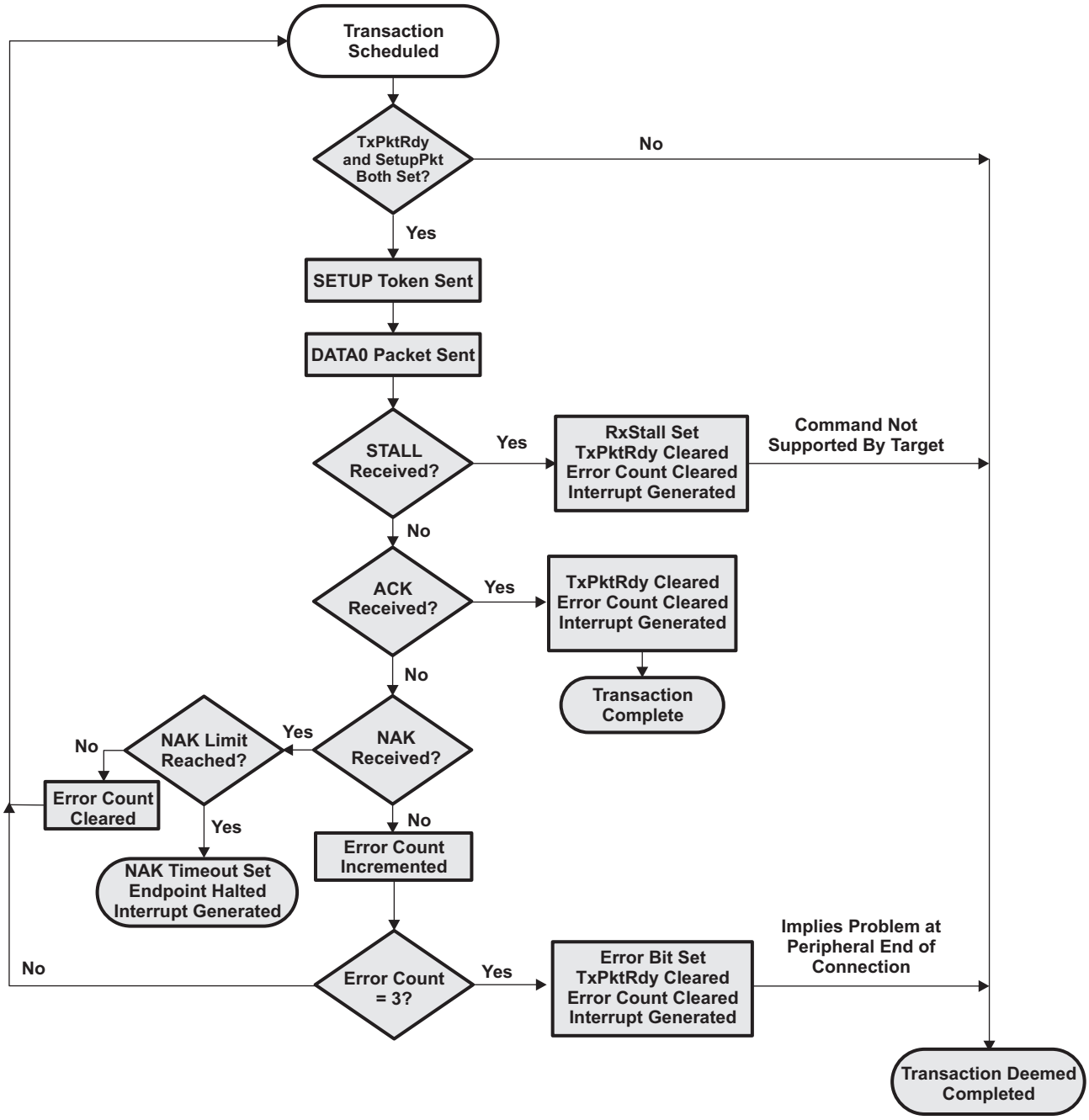
##### 25.7.2.1.1 Setup Stage of Control Transfer

For the SETUP stage of a control transaction (Figure 25-9), the software driving the USB host device needs to:

1. Load the Address of the USB Device into the FADDR Register
2. Load the 8 bytes of the required USB Standard Device Request into the Endpoint 0 FIFO
3. Set SETUPPKT and TXPKTRDY (bits 3 and 1 of HOST_CSR0, respectively).
   **NOTE:** These bits must be set together.
   The controller then proceeds to send a SETUP token followed by the 8-byte request to Endpoint 0 of the addressed device, retrying as necessary. *Note:* On errors, the controller retries the transaction three times.
4. At the end of the attempt to send the 8-byte request data, the controller will generate an Endpoint 0 interrupt. The software should then read HOST_CSR0 to establish whether the RXSTALL bit (bit 2), the ERROR bit (bit 4) or the NAK_TIMEOUT bit (bit 7) has been set.
   If RXSTALL is set, it indicates that the target did not accept the command (for example, because it is not supported by the target device) and so has issued a STALL response. If ERROR is set, it means that the controller has tried to send the SETUP Packet and the following data packet three times without getting any response from the device.
   If NAK_TIMEOUT is set, it means that the controller has received a NAK response to each attempt to send the SETUP packet, for longer than the time set in HOST_NAKLIMIT0. The controller can then be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by flushing the FIFO before clearing the NAK_TIMEOUT bit.
5. If neither of the RXSTALL, ERROR or NAK_TIMEOUT bits is set, the SETUP Stage has been correctly ACKed by the device and the firmware should proceed to following IN Data Phase, OUT Data Phase or Status Phase as specified for the particular Standard Device Request.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 25-9. Flow Chart of Setup Stage of a Control Transfer in Host Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.2.1.2 Data Stage of a Control Transfer

The Data Stage is an optional stage of a Control Transfer depending on the USB Request. The Data Stage and can be either an IN Data Stage or an OUT Data Stage. The following sections show how to handle an Data Stage for both an IN and OUT directions.

##### 25.7.2.1.2.1 Data Stage of a Control Transfer

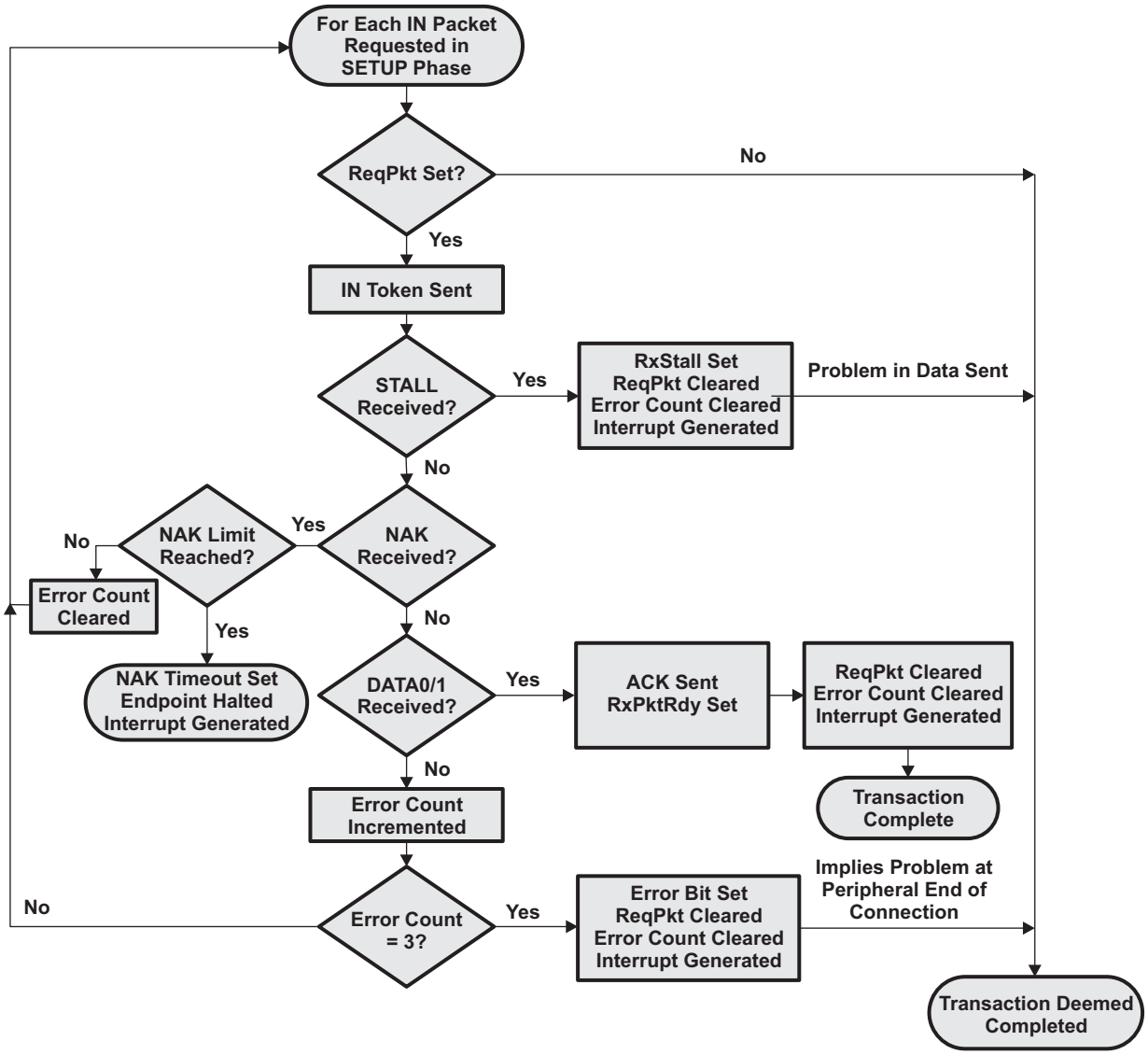
For the IN Data Stage of a control transfer (Figure 25-10), the software driving the USB host device needs to

1. Set REQPKT bit of HOST_CSR0 (bit 5)
2. Wait while the controller sends the IN token and receives the required data back.
3. When the controller generates the Endpoint 0 interrupt, read HOST_CSR0 to establish whether the RXSTALL bit (bit 2), the ERROR bit (bit 4), the NAK_TIMEOUT bit (bit 7) or RXPKTRDY bit (bit 0) has been set.
   If RXSTALL is set, it indicates that the target has issued a STALL response.
   If ERROR is set, it means that the controller has tried to send the required IN token three times without getting any response. If NAK_TIMEOUT bit is set, it means that the controller has received a NAK response to each attempt to send the IN token, for longer than the time set in HOST_NAKLIMIT0. The controller can then be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by clearing REQPKT before clearing the NAK_TIMEOUT bit
4. If RXPKTRDY has been set, the software should read the data from the Endpoint 0 FIFO, then clear RXPKTRDY.
5. If further data is expected, the software should repeat Steps 1-4.

When all the data has been successfully received, the CPU should proceed to the OUT Status Phase of the Control Transfer.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 25-10. Flow Chart of Data Stage (IN Data Phase) of a Control Transfer in Host Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

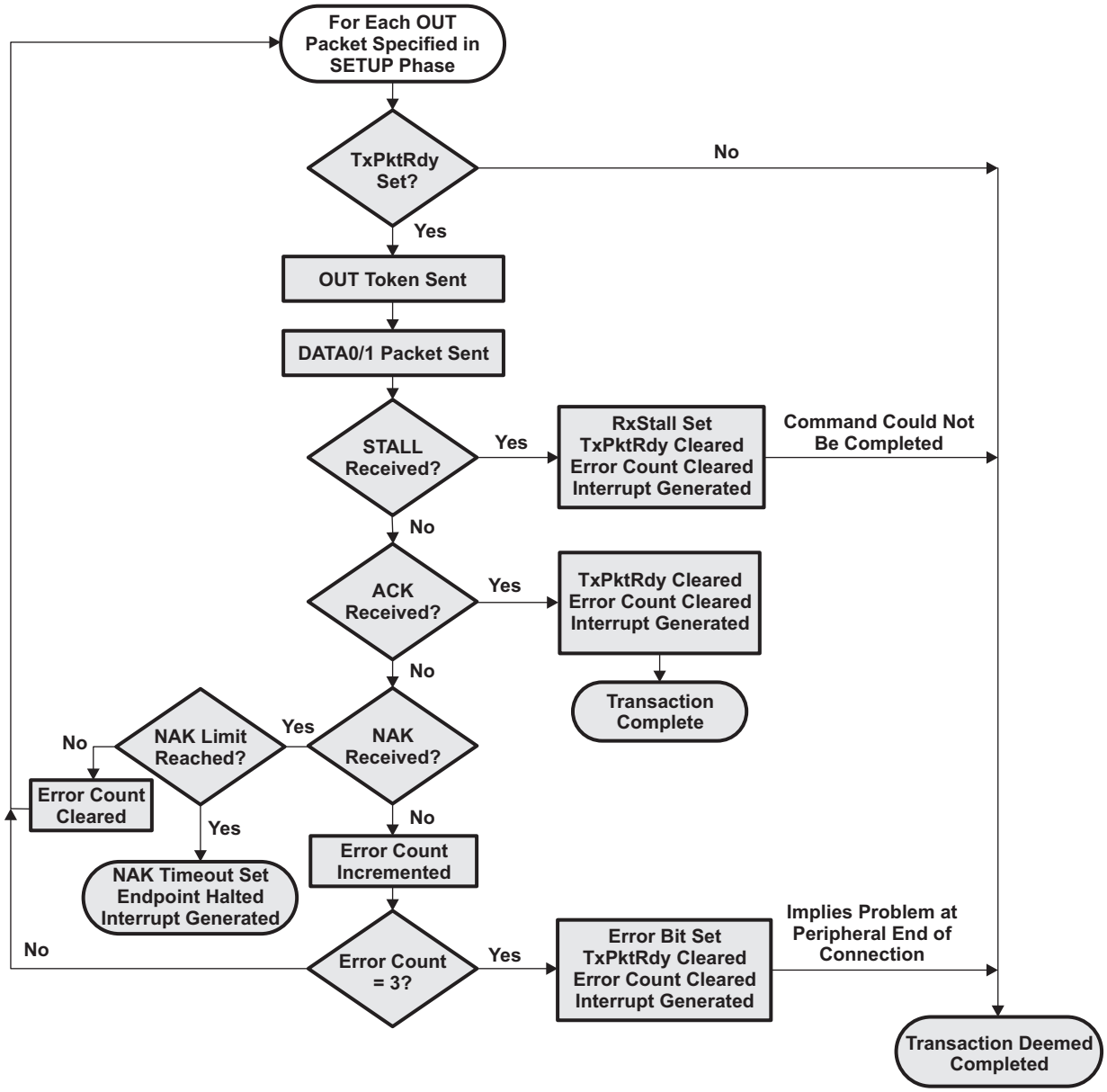
#### 25.7.2.1.2.2 OUT Data Stage of a Control Transfer

For the OUT Data Stage of a control transfer (Figure 25-11), the software driving the USB host device needs to:

1. Load the data to be sent into the endpoint 0 FIFO.
2. Set the TXPKTRDY bit of HOST_CSR0 (bit 1). The controller then proceeds to send an OUT token followed by the data from the FIFO to Endpoint 0 of the addressed device, retrying as necessary.
3. At the end of the attempt to send the data, the controller will generate an Endpoint 0 interrupt. The software should then read HOST_CSR0 to establish whether the RXSTALL bit (bit 2), the ERROR bit (bit 4) or the NAK_TIMEOUT bit (bit 7) has been set.
   If RXSTALL bit is set, it indicates that the target has issued a STALL response.
   If ERROR bit is set, it means that the controller has tried to send the OUT token and the following data packet three times without getting any response. If NAK_TIMEOUT is set, it means that the controller has received a NAK response to each attempt to send the OUT token, for longer than the time set in the HOST_NAKLIMIT0 register. The controller can then be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by flushing the FIFO before clearing the NAK_TIMEOUT bit.
   If none of RXSTALL, ERROR or NAKLIMIT is set, the OUT data has been correctly ACKed.
4. If further data needs to be sent, the software should repeat Steps 1-3.
   When all the data has been successfully sent, the software should proceed to the IN Status Phase of the Control Transfer.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 25-11. Flow Chart of Data Stage (OUT Data Phase) of a Control Transfer in Host Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.2.1.3 Status Stage

The directionality of the Status Stage of a USB Control Transfer will depend on the previous directionality of the Setep (or Data) Stage

##### 25.7.2.1.3.1 IN Status Stage of a Control Transfer

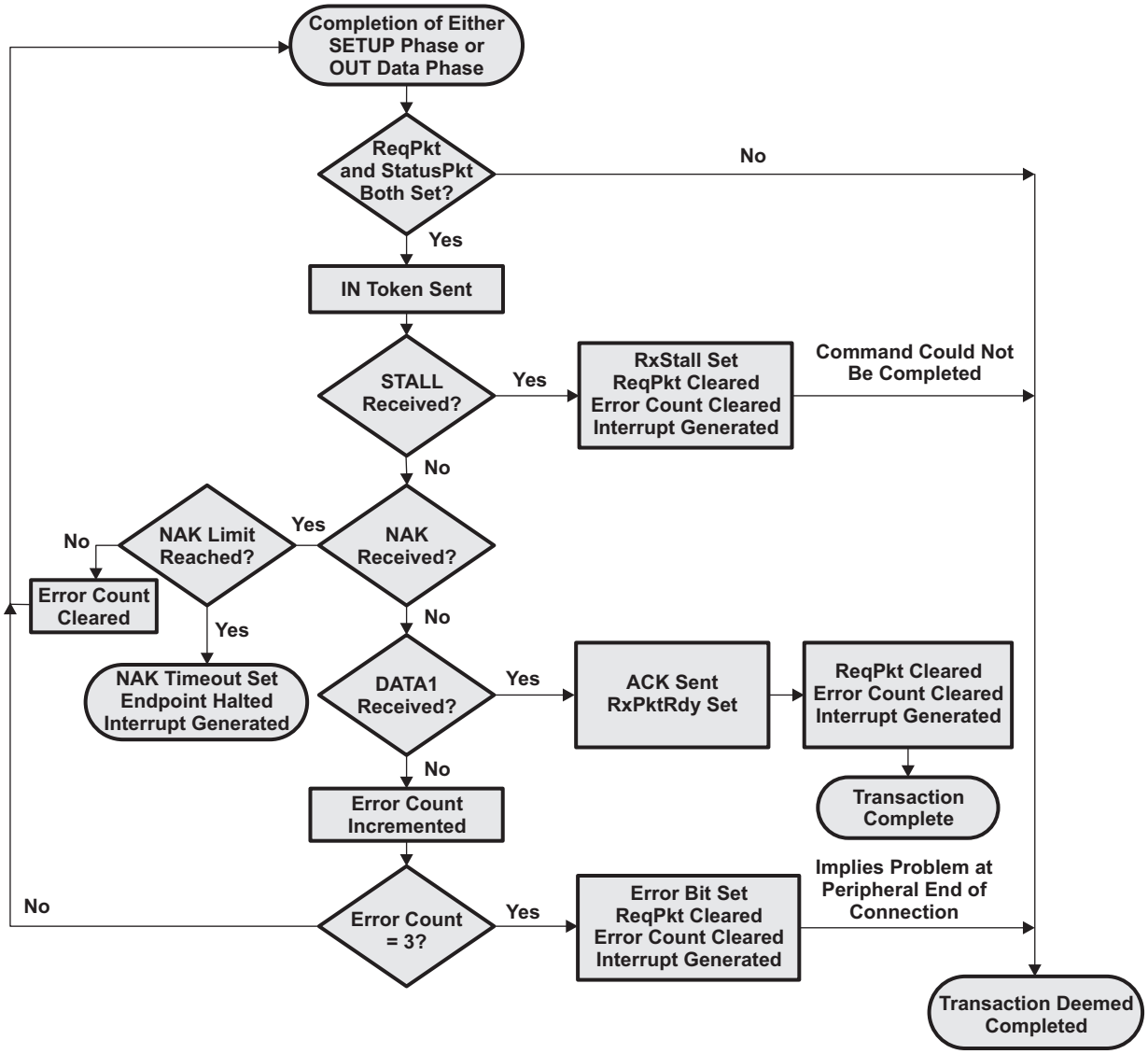
IN Status Phase of a control transfer exists for a Zero Data Request or for a Write Request of a control transfer. The IN Status Phase follows the Setup Stage, if no Data Stage of a control transfer exists, or OUT Data Phase of a Data Stage of a control transfer.

For the IN Status Stage of a control transaction (Figure 25-12), the software driving the USB Host device needs to:

1. Set the STATUSPKT and REQPKT bits of HOST_CSR0 (bit 6 and bit 5, respectively).
2. Wait while the controller sends an IN token and receives a response from the USB peripheral device.
3. When the controller generates the Endpoint 0 interrupt, read HOST_CSR0 to establish whether the RXSTALL bit (bit 2), the ERROR bit (bit 4), the NAK_TIMEOUT bit (bit 7) or RXPKTRDY bit (bit 0) has been set.
   If RXSTALL bit is set, it indicates that the target could not complete the command and so has issued a STALL response.
   If ERROR bit is set, it means that the controller has tried to send the required IN token three times without getting any response.
   If NAK_TIMEOUT bit is set, it means that the controller has received a NAK response to each attempt to send the IN token, for longer than the time set in the HOST_NAKLIMIT0 register. The controller can then be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by clearing REQPKT bit and STATUSPKT bit before clearing the NAK_TIMEOUT bit.
4. The CPU should clear the STATUSPPKT bit of HOST_CSR0 together with (i.e., in the same write operation as) RxPktRdy if this has been set.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 25-12. Flow Chart of Status Stage of Zero Data Request or Write Request of a Control Transfer in Host Mode**





Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.2.1.3.2 OUT Status Stage of a Control Transfer

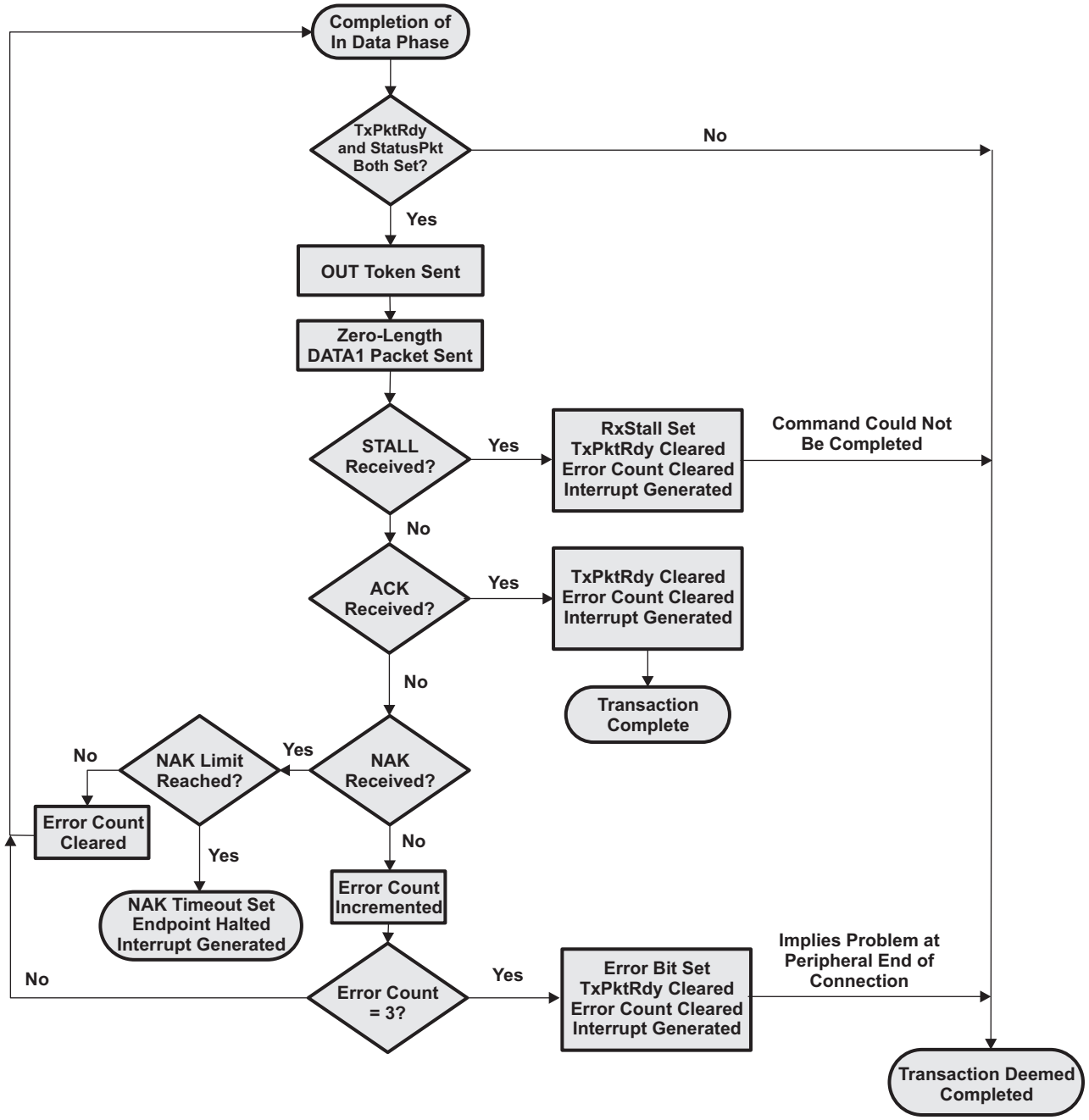
OUT Status Stage of a control transfer exist for a Read Request or a control transfer where data was received by the host controller. The OUT Status phase follows the IN Data Stage of a control transfer.

For the OUT Status Stage of a control transaction (Figure 25-13), the CPU driving the host device needs to:

1. Set STATUSPKT and TXPKTRDY bits of HOST_CSR0 (bit 6 and bit 1, respectively).
   NOTE: These bits need to be set together
2. Wait while the controller sends the OUT token and a zero-length DATA1 packet.
3. At the end of the attempt to send the data, the controller will generate an Endpoint 0 interrupt. The software should then read HOST_CSR0 to establish whether the RXSTALL bit (bit 2), the ERROR bit (bit 4) or the NAK_TIMEOUT bit (bit 7) has been set.
   If RXSTALL bit is set, it indicates that the target could not complete the command and so has issued a STALL response.
   If ERROR bit is set, it means that the controller has tried to send the STATUS Packet and the following data packet three times without getting any response.
   If NAK_TIMEOUT bit is set, it means that the controller has received a NAK response to each attempt to send the IN token, for longer than the time set in the HOST_NAKLIMIT0 register. The controller can then be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by flushing the FIFO before clearing the NAK_TIMEOUT bit.
4. If none of RXSTALL, ERROR or NAK_TIMEOUT bits is set, the STATUS Phase has been correctly ACKed.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 25-13. Chart of Status Stage of a Read Request of a Control Transfer in Host Mode**






Copyright © 2011–2016, Texas Instruments Incorporated

### 25.7.2.2 Bulk Transfer

Bulk transfers are handled by endpoints other than endpoint 0. They is used to handle non-periodic, large bursty communication typically used for a transfer that use any available bandwidth and can also be delayed until bandwidth is available.

#### *25.7.2.2.1 Bulk IN Transfers:*

A Bulk IN transfer may be used to transfer non-periodic data from the external USB peripheral to the host.

The following optional features are available for use with an Rx endpoint used in host mode to receive the data:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO on reception from the host. This allows that one packet can be received while another is being read. Double packet buffering is enabled by setting the DPB bit of RXFIFOSZ register (bit 4).
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint has a packet in its FIFO. This feature can be used to allow the DMA controller to unload packets from the FIFO without processor intervention.
  When DMA is enabled, endpoint interrupt will not be generated for completion of packet reception. Endpoint interrupt will be generated only in the error conditions. For more information see section on CPPI DMA

##### *25.7.2.2.1.1 Bulk IN Setup*

Before initiating any Bulk IN Transactions in Host mode:

- The HOST_RXTYPE register for the endpoint that is to be used needs to be programmed as:
  - Operating speed in the SPEED bit field (bits 7 and 6).
  - Set 10 (binary value) in the PROT field for bulk transfer.
  - Endpoint Number of the target device in RENDPN field. This is the endpoint number contained in the Rx endpoint descriptor returned by the target device during enumeration.
- The RXMAXP register for the controller endpoint must be written with the maximum packet size (in bytes) for the transfer. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the target endpoint.
- The HOST_RXINTERVAL register needs to be written with the required value for the NAK limit (2-215 frames/microframes), or set to zero if the NAK timeout feature is not required.
- The relevant interrupt enable bit in the INTRRXE register should be set (if an interrupt is required for this endpoint).

  **Note:** Whether interrupt is handled from PDR level or Core level, interrupt is required to be enabled at the core level. See details on Core Interrupt registers, CTRLR register, and sections on Interrupt Handling
- The AUTOREQ bit field should be used only when servicing transfers using CPU mode and must be cleared when using DMA Mode. For DMA Mode, dedicated registers USB0/1 Req Registers exist and the HOST_RXCSR[AUTOREQ] should be cleared.
  When DMA is enabled, the following bits of HOST_RXCSR register should be set as:
  - Clear AUTOCLEAR.
  - Set DMAEN (bit 13) to 1 if a DMA request is required for this endpoint.
  - Clear DSINYET (bit 12) to 0 to allow normal PING flow control. This will affect only High Speed transactions.
  - Clear DMAMODE (bit 11) to 0.

    **Note:** If DMA is enabled, the USB0/1 Auto Req register can be set for generating IN tokens automatically after receiving the data. Set the bit field RXn_AUTOREQ (where n is the endpoint number) with binary value 01 or 11.


Copyright © 2011–2016, Texas Instruments Incorporated

When the endpoint is first configured, the endpoint data toggle should be cleared to 0 either by using the DATATOGWREN and DATATOG bits of HOST_RXCSR (bit 10 and bit 9) to toggle the current setting or by setting the CLRDATATOG bit of HOST_RXCSR (bit 7). This will ensure that the data toggle (which is handled automatically by the controller) starts in the correct state. Also if there are any data packets in the FIFO (indicated by the RXPKTRDY bit (bit 0 of HOST_RXCSR) being set), they should be flushed by setting the FLUSHFIFO bit of HOST_RXCSR (bit 4).

**NOTE:** It may be necessary to set this bit twice in succession if double buffering is enabled.

#### *25.7.2.2.1.2 Bulk IN Operation*

When Bulk data is required from the USB peripheral device, the software should set the REQPKT bit in the corresponding HOST_RXCSR register (bit 5). The controller will then send an IN token to the selected peripheral endpoint and waits for data to be returned.

If data is correctly received, RXPKTRDY bit of HOST_RXCSR (bit 0) is set. If the USB peripheral device responds with a STALL, RXSTALL bit (bit 6 of HOST_RXCSR) is set. If a NAK is received, the controller tries again and continues to try until either the transaction is successful or the POLINTVL_NAKLIMIT set in the HOST_RXINTERVAL register is reached. If no response at all is received, two further attempts are made before the controller reports an error by setting the ERROR bit of HOST_RXCSR (bit 2).

The controller then generates the appropriate endpoint interrupt, whereupon the software should read the corresponding HOST_RXCSR register to determine whether the RXPKTRDY, RXSTALL, ERROR or DATAERR_NAKTIMEOUT bit is set and act accordingly. If the DATAERR_NAKTIMEOUT bit is set, the controller can be directed either to continue trying this transaction (until it times out again) by clearing the DATAERR_NAKTIMEOUT bit or to abort the transaction by clearing REQPKT bit before clearing the DATAERR_NAKTIMEOUT bit.

The packets received should not exceed the size specified in the RXMAXP register (as this should be the value set in the wMaxPacketSize field of the endpoint descriptor sent to the host).

In the general case, the application software (if CPU is servicing the endpoint) will need to read each packet from the FIFO individually. If large blocks of data are being transferred, the overhead of calling an interrupt service routine to unload each packet can be avoided by using DMA.

**Note:** When using DMA, see CPPI DMA section for the proper configuration of the core register, HOST_RXCSR.

#### *25.7.2.2.1.3 Bulk IN Error Handling*

If the target wants to shut down the Bulk IN pipe, it will send a STALL response to the IN token. This will result in the RXSTALL bit of HOST_RXCSR (bit 6) being set.

### *25.7.2.2.2 Bulk OUT Transfers*

A Bulk OUT transaction may be used to transfer non-periodic data from the host to the USB peripheral.

Following optional features are available for use with a Tx endpoint used in Host mode to transmit this data:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO awaiting transmission to the peripheral device. Double packet buffering is enabled by setting the DPB bit of TXFIFOSZ register (bit 4).
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint is able to accept another packet in its FIFO. This feature can be used to allow the DMA controller to load packets into the FIFO without processor intervention. For more information on using DMA, consult the section discussing CPPI DMA.
  When DMA is enabled and DMAMODE bit in HOST_TXCSR register is set, an endpoint interrupt will not be generated for completion of packet reception. An endpoint interrupt will be generated only in the error conditions.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.2.2.2.1 Bulk OUT Setup

Before initiating any bulk OUT transactions:

- The target function address needs to be set in the TXFUNCADDR register for the selected controller endpoint. (TXFUNCADDR register is available for all endpoints from EP0 to EP15.)
- The HOST_TXTYPE register for the endpoint that is to be used needs to be programmed as:
  - Operating speed in the SPEED bit field (bits 7 and 6).
  - Set PROT field to 10b for bulk transfer.
  - Enter Endpoint Number of the target device in TENDPN field. This is the endpoint number contained in the OUT(Tx) endpoint descriptor returned by the target device during enumeration.
- The TXMAXP register for the controller endpoint must be written with the maximum packet size (in bytes) for the transfer. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the target endpoint.
- The HOST_TXINTERVAL register needs to be written with the required value for the NAK limit (2-215 frames/microframes), or set to zero if the NAK timeout feature is not required.
- The relevant interrupt enable bit in the INTRTXE register should be set (if an interrupt is required for this endpoint).
- The following bits of HOST_TXCSR register should be set as:
  - Set the MODE bit (bit 13) to 1 to ensure the FIFO is enabled (only necessary if the FIFO is shared with an Rx endpoint).
  - Clear the FRCDATATOG bit (bit 11) to 0 to allow normal data toggle operations.
  - Clear/Set AUTOSET bit (bit 15). The setting of this bit depends on the desire of the user/application in automatically setting the TXPKTRDY bit when servicing transactions using CPU.

  **Note:** If DMA is needs to be used in place of the CPU, the following table displays the setting of the core register HOST_TXCSR register in Host mode. For the CPPI DMA registers settings consult the section on CPPI DMA within this document.

| Bit Field | Bit Name | Description |
|---|---|---|
| Bit 15 | AUTOSET | Cleared to 0 if using DMA. For CPU Mode use, if AUTOSET bit is set, the TXPKTRDY bit will be automatically set when data of the maximum packet size is loaded into the FIFO. |
| Bit 14 | ISO | Cleared to 0 for bulk mode operation. |
| Bit 13 | MODE | Set to 1 to make sure the FIFO is enabled (only necessary if the FIFO is shared with an RX endpoint) |
| Bit 12 | DMAEN | Set to 1 to enable DMA usage; not needed if CPU is being used to service the Tx Endpoint |
| Bit 11 | FRCDATATOG | Cleared to 0 to allow normal data toggle operations. |
| Bit 10 | DMAMODE | Set to 1 when DMA is used to service Tx FIFO. |

When the endpoint is first configured, the endpoint data toggle should be cleared to 0 either by using the DATATOGWREN bit and DATATOG bit of HOST_TXCSR (bit 9 and bit 8) to toggle the current setting or by setting the CLRDATATOG bit of HOST_TXCSR (bit 6). This will ensure that the data toggle (which is handled automatically by the controller) starts in the correct state. Also, if there are any data packets in the FIFO (indicated by the FIFONOTEMPTY bit of HOST_TXCSR register (bit 1) being set), they should be flushed by setting the FLUSHFIFO bit (bit 3 of HOST_TXCSR).

**NOTE:** It may be necessary to set this bit twice in succession if double buffering is enabled.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.2.2.2.2 Bulk OUT Operation

When Bulk data is required to be sent to the USB peripheral device, the software should write the first packet of the data to the FIFO (or two packets if double-buffered) and set the TXPKTRDY bit in the corresponding HOST_TXCSR register (bit 0). The controller will then send an OUT token to the selected peripheral endpoint, followed by the first data packet from the FIFO.

If data is correctly received by the peripheral device, an ACK should be received whereupon the controller will clear TXPKTRDY bit of HOST_TXCSR (bit 0). If the USB peripheral device responds with a STALL, the RXSTALL bit (bit 5) of HOST_TXCSR is set. If a NAK is received, the controller tries again and continues to try until either the transaction is successful or the NAK limit set in the HOST_TXINTERVAL register is reached. If no response at all is received, two further attempts are made before the controller reports an error by setting ERROR bit in HOST_TXCSR (bit 2).

The controller then generates the appropriate endpoint interrupt, whereupon the software should read the corresponding HOST_TXCSR register to determine whether the RXSTALL (bit 5), ERROR (bit 2) or NAK_TIMEOUT (bit 7) bit is set and act accordingly. If the NAK_TIMEOUT bit is set, the controller can be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by flushing the FIFO before clearing the NAK_TIMEOUT bit.

If large blocks of data are being transferred, then the overhead of calling an interrupt service routine to load each packet can be avoided by using DMA.

#### 25.7.2.2.2.3 Bulk OUT Error Handling

If the target wants to shut down the Bulk OUT pipe, it will send a STALL response. This is indicated by the RXSTALL bit of HOST_TXCSR register (bit 5) being set.

### 25.7.2.3 Interrupt Transfer

When the controller is operating as the host, interactions with an Interrupt endpoint on the USB peripheral device are handled in very much the same way as the equivalent Bulk transactions (described in previous sections).

The principal difference as far as operational steps are concerned is that the PROT field of HOST_RXTYPE and HOST_TXTYPE (bits 5-4) need to be set (binary value) to represent an Interrupt transaction. The required polling interval also needs to be set in the HOST_RXINTERVAL and HOST_TXINTERVAL registers.

### 25.7.2.4 Isochronous Transfer

Isochronous transfers are used when working with isochronous data. Isochronous transfers provide periodic, continuous communication between host and device.

An Isochronous IN transaction is used to transfer periodic data from the USB peripheral to the host.

#### 25.7.2.4.1 Isochronous IN Transfers

The following optional features are available for use with an Rx endpoint used in Host mode to receive this data:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO on reception from the host. This allows that one packet can be received while another is being read. Double packet buffering is enabled by setting the DPB bit of RXFIFOSZ register (bit 4).
- AutoRequest: When the AutoRequest feature is enabled, the REQPKT bit of HOST_RXCSR (bit 5) will be automatically set when the RXPKTRDY bit is cleared. This only applies when the CPU is being used to service the endpoint. When using DMA, this bit field needs to be cleared to Zero. CPPI DMA has its own configuration registers that renders a similar task, USB0/1 Auto Req registers, that needs to be used to have a similar effect. For more information, see section on CPPI DMA.
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint has a packet in its FIFO. This feature can be used to allow the DMA controller to unload packets from the FIFO without processor intervention. However, this feature is not particularly useful with isochronous endpoints because the packets transferred are often not maximum packet size.


Copyright © 2011–2016, Texas Instruments Incorporated

- When DMA is enabled, endpoint interrupt will not be generated for completion of packet reception. Endpoint interrupt will be generated only in the error conditions.

#### 25.7.2.4.1.1 Isochronous IN Transfer Setup

Before initiating an Isochronous IN Transactions in Host mode:

- The target function address needs to be set in the RXFUNCADDR register for the selected controller endpoint (RXFUNCADDR register is available for all endpoints from EP0 to EP4).
- The HOST_RXTYPE register for the endpoint that is to be used needs to be programmed as:
  - Operating speed in the SPEED bit field (bits 7 and 6).
  - Set 01 (binary value) in the PROT field for isochronous transfer.
  - Endpoint Number of the target device in RENDPN field. This is the endpoint number contained in the Rx endpoint descriptor returned by the target device during enumeration.
- The RXMAXP register for the controller endpoint must be written with the maximum packet size (in bytes) for the transfer. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the target endpoint.
- The HOST_RXINTERVAL register needs to be written with the required transaction interval (usually one transaction per frame/microframe).
- The relevant interrupt enable bit in the INTRRXE register should be set (if an interrupt is required for this endpoint).
- The following bits of HOST_RXCSR register should be set as:
  - Clear AUTOCLEAR
  - Set the DMAEN bit (bit 13) to 1 if a DMA request is required for this endpoint.
  - Clear the DISNYET it (bit 12) to 0 to allow normal PING flow control. This will only affect High Speed transactions.
  - Clear DMAMODE bit (bit 11) to 0.
- If DMA is enabled, AUTOREQ register can be set for generating IN tokens automatically after receiving the data. Set the bit field RXn_AUTOREQ (where n is the endpoint number) with binary value 01 or 11. For detailed information on using CPPI DMA, consult related section within this document.

#### 25.7.2.4.1.2 Isochronous IN Operation

The operation starts with the software setting REQPKT bit of HOST_RXCSR (bit 5). This causes the controller to send an IN token to the target.

When a packet is received, an interrupt is generated which the software may use to unload the packet from the FIFO and clear the RXPKTRDY bit in the HOST_RXCSR register (bit 0) in the same way as for a Bulk Rx endpoint. As the interrupt could occur almost any time within a frame(/microframe), the timing of FIFO unload requests will probably be irregular. If the data sink for the endpoint is going to some external hardware, it may be better to minimize the requirement for additional buffering by waiting until the end of each frame before unloading the FIFO. This can be done by using the SOF_PULSE signal from the controller to trigger the unloading of the data packet. The SOF_PULSE is generated once per frame(/microframe). The interrupts may still be used to clear the RXPKTRDY bit in HOST_RXCSR.

#### 25.7.2.4.1.3 Isochronous IN Error Handling

If a CRC or bit-stuff error occurs during the reception of a packet, the packet will still be stored in the FIFO but the DATAERR_NAKTIMEOUT bit of HOST_RXCSR (bit 3) is set to indicate that the data may be corrupt.

**Note:** The number of USB packets sent in any microframe will depend on the amount of data to be transferred, and is indicated through the PIDs used for the individual packets. If the indicated numbers of packets have not been received by the end of a microframe, the INCOMPRX bit in the HOST_RXCSR register will be set to indicate that the data in the FIFO is incomplete. Equally, if a packet of the wrong data type is received, then the PID Error bit is the HOST_RXCSR register will be set. In each case, an interrupt will, however, still be generated to allow the data that has been received to be read from the FIFO.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.2.4.2 Isochronous OUT Transfers

An Isochronous OUT transaction may be used to transfer periodic data from the host to the USB peripheral.

Following optional features are available for use with a Tx endpoint used in Host mode to transmit this data:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO awaiting transmission to the peripheral device. Double packet buffering is enabled by setting the DPB bit of TXFIFOSZ register (bit 4).

DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint is able to accept another packet in its FIFO. This feature can be used to allow the DMA controller to load packets into the FIFO without processor intervention.

However, this feature is not particularly useful with isochronous endpoints because the packets transferred are often not maximum packet size.

When DMA is enabled and endpoint interrupt will not be generated for completion of packet reception. Endpoint interrupt will be generated only in the error conditions.

##### 25.7.2.4.2.1 Isochronous OUT Transfer Setup

Before initiating any Isochronous OUT transactions:

- The target function address needs to be set in the TXFUNCADDR register for the selected controller endpoint (TXFUNCADDR register is available for all endpoints from EP0 to EP4).
- The HOST_TXTYPE register for the endpoint that is to be used needs to be programmed as:
  - Operating speed in the SPEED bit field (bits 7 and 6).
  - Set 01 (binary value) in the PROT field for isochronous transfer.
  - Endpoint Number of the target device in TENDPN field. This is the endpoint number contained in the OUT(Tx) endpoint descriptor returned by the target device during enumeration.
- The TXMAXP register for the controller endpoint must be written with the maximum packet size (in bytes) for the transfer. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the target endpoint.
- The HOST_TXINTERVAL register needs to be written with the required transaction interval (usually one transaction per frame/microframe).
- The relevant interrupt enable bit in the INTRTXE register should be set (if an interrupt is required for this endpoint).
- The following bits of HOST_TXCSR register should be set as:
  - Set the MODE bit (bit 13) to 1 to ensure the FIFO is enabled (only necessary if the FIFO is shared with an Rx endpoint).
  - Set the DMAEN bit (bit 12) to 1 if a DMA request is required for this endpoint
  - The FRCDATATOG bit (bit 11) is ignored for isochronous transactions.
  - Set the DMAMODE bit (bit 10) to 1 when DMA is enabled.
    For more details in using DMA, consult CPPI DMA section within this document.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.7.2.4.2.2 Isochronous OUT Transfer Operation

The operation starts when the software writes to the FIFO and sets TXPKTRDY bit of HOST_TXCSR (bit 0). This triggers the controller to send an OUT token followed by the first data packet from the FIFO.

An interrupt is generated whenever a packet is sent and the software may use this interrupt to load the next packet into the FIFO and set the TXPKTRDY bit in the HOST_TXCSR register (bit 0) in the same way as for a Bulk Tx endpoint. As the interrupt could occur almost any time within a frame, depending on when the host has scheduled the transaction, this may result in irregular timing of FIFO load requests. If the data source for the endpoint is coming from some external hardware, it may be more convenient to wait until the end of each frame before loading the FIFO as this will minimize the requirement for additional buffering. This can be done by using the SOF_PULSE signal from the controller to trigger the loading of the next data packet. The SOF_PULSE is generated once per frame(/microframe). The interrupts may still be used to set the TXPKTRDY bit in HOST_TXCSR.

## 25.8 DMA

The Communications Port Programming Interface (CPPI) Version 4.1 DMA module supports the transmission and reception of USB packets. The DMA is designed to facilitate the segmentation and reassembly of packets to/from smaller data blocks that are natively compatible with the specific requirements of each networking port. The DMA controller maintains state information for each of the ports which allows packet segmentation and reassembly operations to be time division multiplexed between USB Endpoints in order to share the underlying DMA hardware. A DMA scheduler is used to control the ordering and rate at which this multiplexing occurs.

The CPPI DMA controller sub-module is a common 15 port DMA controller. It supports 15 Tx and 15 Rx ports attaches to the associated endpoint in the USB controller. Port 1 maps to endpoint 1 and Port 2 maps to endpoint 2 and so on with Port 15 mapped to endpoint 15 of USB0. Port 16 is mapped to USB1 EP1 and so on with Port 30 mapped to USB1 EP15

### 25.8.1 CPPI Terminology

**Host/CPU**— The host is an intelligent system resource that configures and manages each communications control module. The host is responsible for allocating memory, initializing all data structures, and responding to port interrupts.

**Main Memory** — The area of data storage managed by the CPU. The CPPI DMA (CDMA) reads and writes DMA packets from and to main memory. This memory can exist internal or external from the device.

**CPPI FIFO** — The CPPI FIFO provides FIFO interfaces (for each of the 15 transmit and 15 receive endpoints) between the USB Controllers and Main Memory.

**CPPI DMA (CDMA)** — The CDMA is responsible for transferring data between the CPPI FIFO and Main Memory.

**Transfer DMA (XDMA)** — The XDMA receives DMA requests from the USB Controller and initiates DMAs to the CPPI FIFO.

**Packet Queue** — A packet queue is hardware managed list of valid (i.e. populated) packet descriptors that is used for forwarding a packet from one entity to another for any number of purposes.
**NOTE:** All descriptors (regardless of type) must be allocated at addresses that are naturally aligned to the smallest power of 2 that is equal to or greater than the descriptor size.

**Queue Manager (QM)** — The QM is responsible for accelerating management of a variety of Packet Queues and Free Descriptor / Buffer Queues.

**Data Buffer** — A data buffer is a single data structure that contains payload information for transmission to or reception from a port. A data buffer is a byte aligned contiguous block of memory used to store packet payload data. A data buffer may hold any portion of a packet and may be linked together (via descriptors) with other buffers to form DMA packets. Data buffers may be allocated anywhere within the 32-bit memory space. The Buffer Length field of the packet descriptor indicates the number of valid data bytes in the buffer. There may be from 1 to 4M-1 valid data bytes in each buffer.


Copyright © 2011–2016, Texas Instruments Incorporated

**Buffer Descriptor** — A buffer descriptor is a single data structure that contains information about one or more data buffers. This type of descriptor is required when more than one descriptor is needed to define an entire packet, i.e., it either defines the middle of a packet or end of a packet.

**Packet Descriptor** — A packet descriptor is another name for the first buffer descriptor within a packet. Some fields within a data buffer descriptor are only valid when it is a packet descriptor including the tags, packet length, packet type, and flags. This type of descriptor is always used to define a packet since it provides packet level information that is useful to both the ports and the Host in order to properly process the packet. It is the only descriptor used when single descriptor solely defines a packet. When multiple descriptors are needed to define a packet, the packet descriptor is the first descriptor used to define a packet.

**Teardown Descriptor** — Teardown Descriptor is a special structure which is not used to describe either a packet or a buffer but is instead used to describe the completion of a channel halt and teardown event. Channel teardown is an important function because it ensures that when a connection is no longer needed that the hardware can be reliably halted and any remaining packets which had not yet been transmitted can be reclaimed by the Host without the possibility of losing buffer or descriptor references (which results in a memory leak).

**Free Descriptor/Buffer Queue** — A free descriptor/buffer queue is a hardware managed list of available descriptors with pre-linked empty buffers that are to be used by the receive ports for host type descriptors. Free Descriptor/Buffer Queues are implemented by the Queue Manager.

**CDMA Scheduler (CDMAS)** — The CDMAS is responsible for scheduling CDMA transmit and receive operations. It uses Queue Indicators from the QM and the CDMA to determine the types of operations to schedule.

**Endpoint FIFOs** — The Endpoint FIFOs are the USB packet storage elements used by the USB Controller for packet transmission or reception. The XDMA transfers data between the CPPI FIFO and the Endpoint FIFOs for transmit operations and between the Endpoint FIFOs and the CPPI FIFO for receive operations.

**Port** — A port is the communications module (peripheral hardware) that contains the control logic for Direct Memory Access for a single transmit/receive interface or set of interfaces. A port is usually subdivided into transmit and receive pairs which are independent of each other. Each endpoint, excluding endpoint 0, has its own dedicated port.


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.8.2 Data Structures

Two data structures are mainly used to identify data buffers used by packet and buffer descriptors. A third Descriptor, Teardown Descriptor, exists. The purpose of this Descriptor is a channel halt and teardown event. Each of these Descriptor layouts as well as bit fields are shown below.

#### 25.8.2.1 Packet Descriptor

Packet Descriptors are designed to be used when USB like application requires support for true, unlimited fragment count scatter/gather type operations. The DMA Packet Descriptor is synonymous with the first Buffer Descriptor in a sequence of linked Buffer Descriptors. The Packet Descriptor is the first Buffer Descriptor on multiple descriptors DMA Packet or the only Buffer Descriptor in a single descriptor DMA Packet. The Packet Descriptor contains the following information:

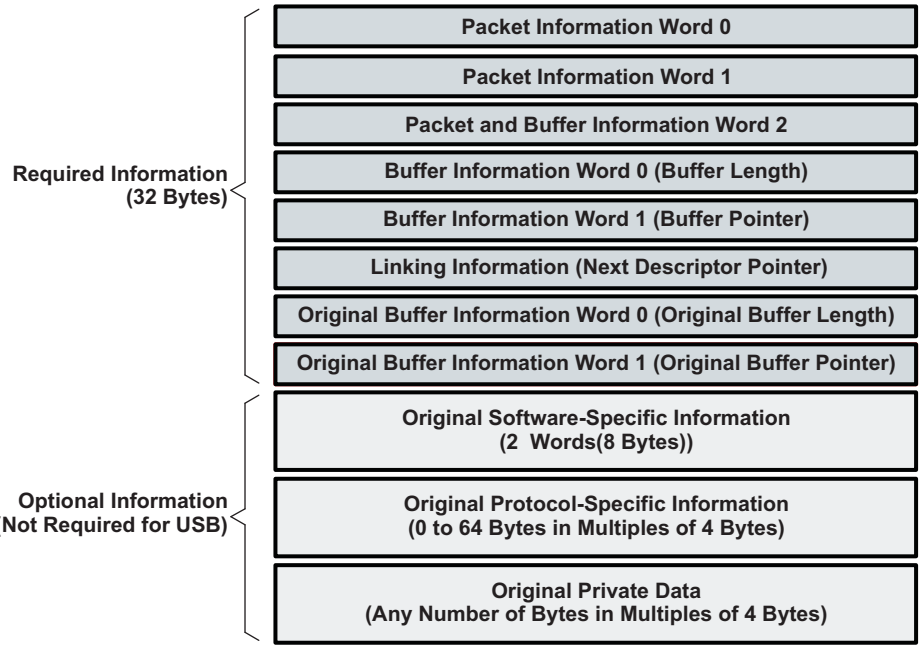
- Indicator which identifies the descriptor as a packet descriptor (always 10h
- Source and destination tags (reserved)
- Packet type
- Packet length
- Protocol specific region size
- Protocol specific control/status bits
- Pointer to the first valid byte in the SOP data buffer
- Length of the SOP data buffer
- Pointer to the next buffer descriptor in the packet

Packet descriptors can vary in size by design of their defined fields from 32 bytes up to 104 bytes. Within this range, packet descriptors always contain at least 32 bytes of required information and may also contain an additional 8 bytes of software specific tagging information and up to 64 bytes (indicated in 4 byte increments) of protocol specific information. How much protocol specific information (and therefore the allocated size of the descriptors) is application dependent. Port will make use of the first 32 bytes only.

From a general USB use perspective, a 32-byte descriptor size is sufficient and the use of this size is expected for a normal USB usage.

The packet descriptor layout is shown in Figure 25-14 and described within Table 25-10 through Table 25-17.

**Figure 25-14. Packet Descriptor Layout**





Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-10. Packet Descriptor Word 0 (PD0) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-27 | Descriptor type | The host packet descriptor type is 16 decimal (10h). The CPU initializes this field. |
| 26-22 | Protocol-specific valid word count | This field indicates the valid number of 32-bit words in the protocol-specific region. The CPU initializes this field. This is encoded in increments of 4 bytes as:<br>0 = 0 byte<br>1 = 4 bytes<br>...<br>16 = 64 bytes<br>17-31 = Reserved |
| 21-0 | Packet length | The length of the packet in bytes. If the packet length is less than the sum of the buffer lengths, then the packet data will be truncated. A packet length greater than the sum of the buffers is an error. The valid range for the packet length is 0 to (4M - 1) bytes. The CPU initializes this field for transmitted packets; the DMA overwrites this field on packet reception. |

**Table 25-11. Packet Descriptor Word 1 (PD1) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-27 | Source Tag; Port # | This field indicates the port number (0-31) from which the packet originated. The DMA overwrites this field on packet reception. This is the RX Endpoint number from which the packet originated. |
| 26-21 | Source Tag; Channel # | This field indicates the channel number within the port from which the packet originated. The DMA overwrites this field on packet reception. This field is always 0-67. |
| 20-16 | Source Tag: Sub-channel # | This field indicates the sub-channel number (0-31) within the channel from which the packet originated. The DMA overwrites this field on packet reception. This field is always 0. |
| 15-0 | Destination Tag | This field is application-specific. This field is always 0. |

**Table 25-12. Packet Descriptor Word 2 (PD2) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31 | Packet error | This bit indicates if an error occurred during reception of this packet (0 = no error occurred; 1 = error occurred). The DMA overwrites t his field on packet reception. Additional information about different errors may be encoded in the protocol specific fields in the descriptor. |
| 30-26 | Packet type | This field indicates the type of this packet. The CPU initializes this field for transmitted packets; the DMA overwrites this field on packet reception. This field is encoded as:<br>5 = USB<br>8-31 = Reserved |
| 25-20 | Reserved | Reserved |
| 19 | Zero-length packet indicator | If a zero-length USB packet is received, the XDMA will send the CDMA a data block with a byte count of 0 and this bit is set. The CDMA will then perform normal EOP termination of the packet without transferring data. For transmit, if a packet has this bit set, the XDMA will ignore the CPPI packet size and send a zero-length packet to the USB controller. |
| 18-16 | Protocol-specific | This field contains protocol-specific flags/information that can be assigned based on the packet type. Not used for USB. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-12. Packet Descriptor Word 2 (PD2) Bit Field Descriptions (continued)**

| Bits | Field Name | Description |
|---|---|---|
| 15 | Return policy | This field indicates the return policy for this packet. The CPU initializes this field.<br>0 = Entire packet (still linked together) should be returned to the queue specified in bits 13-0.<br>1 = Each buffer should be returned to the queue specified in bits 13-0 of Word 2 in their respective descriptors. The Tx DMA will return each buffer in sequence. |
| 14 | On-chip | This field indicates whether or not this descriptor is in a region which is in on-chip memory space (1) or in external memory (0). |
| 31-12 | Packet return queue mgr # | This field indicates which queue manager in the system the descriptor is to be returned to after transmission is complete. This field is not altered by the DMA during transmission or reception and is initialized by the CPU. There is only one queue manager in the USB HS/FS device controller. This field must always be 0. |
| 11-0 | Packet return queue # | This field indicates the queue number within the selected queue manager that the descriptor is to be returned to after transmission is complete. This field is not altered by the DMA during transmission or reception and is initialized by the CPU. |

**Table 25-13. Packet Descriptor Word 3 (PD3) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-22 | Reserved | Reserved |
| 21-0 | Buffer 0 length | The buffer length field indicates how many valid data bytes are in the buffer. The CPU initializes this field for transmitted packets; the DMA overwrites this field on packet reception. |

**Table 25-14. Packet Descriptor Word 4 (PD4) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | Buffer 0 pointer | The buffer pointer is the byte-aligned memory address of the buffer associated with the buffer descriptor. The CPU initializes this field for transmitted packets; the DMA overwrites this field on packet reception. |

**Table 25-15. Packet Descriptor Word 5 (PD5) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | Next descriptor pointer | The 32-bit word aligned memory address of the next buffer descriptor in the packet. If the value of this pointer is zero, then the current buffer is the last buffer in the packet. The CPU initializes this field for transmitted packets; the DMA overwrites this field on packet reception. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-16. Packet Descriptor Word 6 (PD6) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-22 | Reserved | Reserved |
| 21-0 | Original buffer 0 length | The buffer length field indicates the original size of the buffer in bytes. This value is not overwritten during reception. This value is read by the Rx DMA to determine the actual buffer size as allocated by the CPU at initialization. Since the buffer length in Word 3 is overwritten by the Rx port during reception, this field is necessary to permanently store the buffer size information. |

**Table 25-17. Packet Descriptor Word 7 (PD7) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-22 | Reserved | Reserved |
| 21-0 | Original buffer 0 pointer | The buffer pointer is the byte-aligned memory address of the buffer associated with the buffer descriptor. This value is not overwritten during reception. This value is read by the Rx DMA to determine the actual buffer location as allocated by the CPU at initialization. Since the buffer pointer in Word 4 is overwritten by the Rx port during reception, this field is necessary to permanently store the buffer pointer information. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.8.2.2 Buffer Descriptor (BD)

The Buffer Descriptor is identical in size and organization to a Packet Descriptor but does not include valid information in the packet level fields and does not include a populated region for protocol specific information. The packet level fields are not needed since the Packet Descriptor already contains this information and additional copy of this data is not needed/necessary.

Buffer descriptors are designed to be linked onto a Packet Descriptor or another buffer descriptor to provide support for unlimited scatter / gather type operations. Buffer descriptors provide information about a single corresponding data buffer. Every buffer descriptor stores the following information:

- Pointer to the first valid byte in the data
- Length of the data buffer
- Pointer to the next buffer descriptor in the packet

Buffer descriptors always contain 32 bytes of required information. Since it is a requirement that it is possible to convert a descriptor between a Buffer Descriptor and a Packet Descriptor (by filling in the appropriate fields) in practice, Buffer Descriptors will be allocated using the same sizes as Packet Descriptors. In addition, since the 5 LSBs of the Descriptor Pointers are used in CPPI 4.1 for the purpose of indicating the length of the descriptor, the minimum size of a descriptor is always 32 bytes

The buffer descriptor layout is shown in Figure 25-15 and described within Table 25-18 through Table 25-25 .

**Figure 25-15. Buffer Descriptor (BD) Layout**

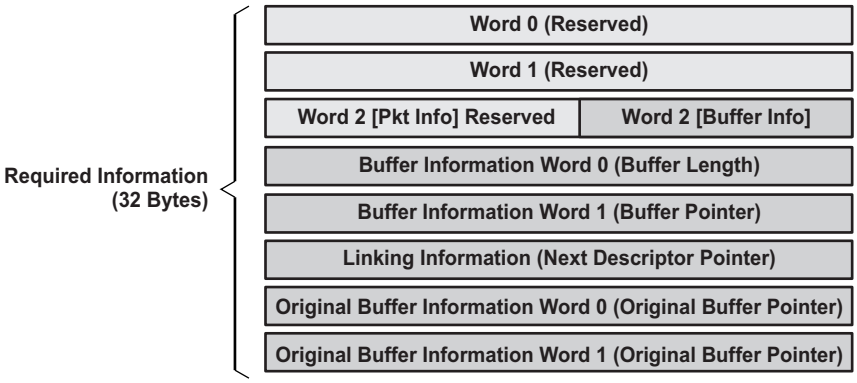



**Table 25-18. Buffer Descriptor Word 0 (BD0) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | Reserved | Reserved |

**Table 25-19. Buffer Descriptor Word 1 (BD1) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | Reserved | Reserved |

**Table 25-20. Buffer Descriptor Word 2 (BD2) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-15 | Reserved | Reserved |
| 14 | On-chip | This field indicates whether or not this descriptor is in a region which is on-chip memory space (1) or in external memory (0). |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-20. Buffer Descriptor Word 2 (BD2) Bit Field Descriptions (continued)**

| Bits | Field Name | Description |
|---|---|---|
| 13-12 | Packet return queue mgr # | This field indicates which queue manager in the system the descriptor is to be returned to after transmission is complete. This field is not altered by the DMA during transmission or reception and is initialized by the CPU. There is only 1 queue manager in the USB HS/FS device controller. This field must always be 0. |
| 11-0 | Packet return queue # | This field indicates the queue number within the selected queue manager that the descriptor is to be returned to after transmission is complete. This field is not altered by the DMA during transmission or reception and is initialized by the CPU. |

**Table 25-21. Buffer Descriptor Word 3 (BD3) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-22 | Reserved | Reserved |
| 21-0 | Buffer 0 length | The buffer length field indicates how many valid data bytes are in the buffer. The CPU initializes this field for transmitted packets. The DMA overwrites this field upon packet reception. |

**Table 25-22. Buffer Descriptor Word 4 (BD4) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | Buffer 0 pointer | The buffer pointer is the byte-aligned memory address of the buffer associated with the buffer descriptor. The CPU initializes this field for transmitted packets. The DMA overwrites this field upon packet reception. |

**Table 25-23. Buffer Descriptor Word 5 (BD5) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | Next description pointer | The 32-bit word aligned memory address of the next buffer descriptor in the packet. If the value of this pointer is zero, then the current descriptor is the last descriptor in the packet. The CPU initializes this field for transmitted packets. The DMA overwrites this field upon packet reception. |

**Table 25-24. Buffer Descriptor Word 6 (BD6) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-22 | Reserved | Reserved |
| 21-0 | Original buffer 0 length | The buffer length field indicates the original size of the buffer in bytes. This value is not overwritten during reception. This value is read by the Rx DMA to determine the actual buffer size as allocated by the CPU at initialization. Since the buffer length in Word 3 is overwritten by the Rx port during reception, this field is necessary to permanently store the buffer size information. |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-25. Buffer Descriptor Word 7 (BD7) Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | Original buffer 0 pointer | The buffer pointer is the byte-aligned memory address of the buffer associated with the buffer descriptor. This value is not overwritten during reception. This value is read by the Rx DMA to determine the actual buffer location as allocated by the CPU at initialization. Since the buffer pointer in Word 4 is overwritten by the Rx port during reception, this field is necessary to permanently store the buffer pointer information. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.8.2.3 Teardown Descriptor

The Teardown Descriptor is used to describe a channel halt and teardown event. Channel teardown ensures that when a connection is no longer needed that the hardware can be reliably halted and any remaining packets which had not yet been transmitted can be reclaimed by the CPU without the possibility of losing buffer or descriptor references (which results in a memory leak).

The Teardown descriptor is always 32 bytes long and is comprised of 4 bytes of actual teardown information and 28 bytes of pad. The teardown descriptor layout and associated bit field descriptions are shown in the Figure below and Tables that follows. Since the 5 LSBs of the descriptor pointers are used in CPPI 4.1 for the purpose of indicating the length of the descriptor, the minimum size of a descriptor is 32 bytes.

The teardown descriptor contains the following information:

- Indicator which identifies the descriptor as a teardown packet descriptor
- DMA controller number where teardown
- Channel number within DMA where teardown occurred
- Indicator of whether this teardown was for the Tx or Rx channel

**Figure 25-16. Teardown Descriptor Layout**

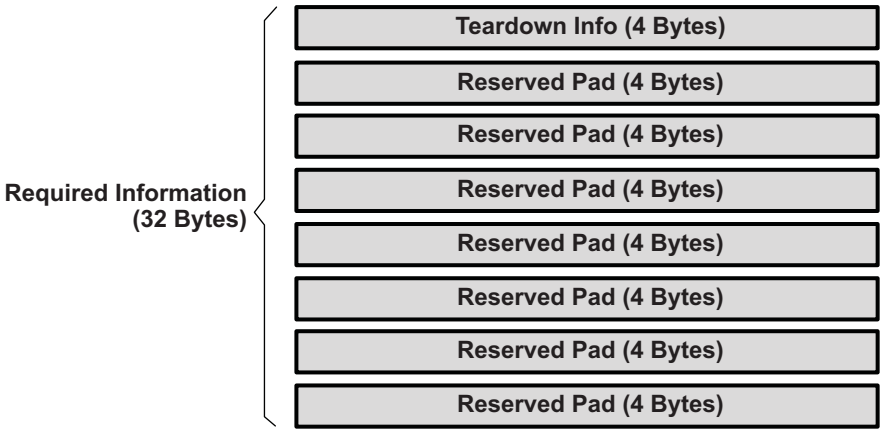



**Table 25-26. Teardown Descriptor Word 0 Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-27 | Descriptor type | The teardown descriptor type is 19 decimal (13h) |
| 26-17 | Reserved | Reserved |
| 16 | TX_RX | Indicates whether teardown is a TX (0) or RX (1). |
| 15-10 | DMA number | Indicates the DMA number for this teardown. |
| 9-6 | Reserved | Reserved |
| 5-0 | Channel number | Indicates the channel number within the DMA that was torn down. |

**Table 25-27. Teardown Descriptor Words 1 to 7 Bit Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | Reserved | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

Teardown operation of an endpoint requires three operations. The teardown register in the CPPI DMA must be written, the corresponding endpoint bit in TEARDOWN of the USB module must be set, and the FlushFIFO bit in the USB controller Tx/RxCSR register must be set.

The following is the Transmit teardown procedure highlighting the steps required to be followed:

1. Set the TX_TEARDOWN bit in the CPPI DMA TX channel n global configuration register (TXGCRn).
2. Set the appropriate TX_TDOWN bit in the USBOTG controller's USB teardown register (TEARDOWN). Write Tx Endpoint Number to teardown to TEARDOWN[TX_TDOWN] field.
3. Check if the teardown descriptor has been received on the teardown queue: The Completion Queue (see Queue Assignment table) is usually used as the Teardown queue when the Teardown descriptor has been received, the descriptor address will be loaded onto CTRLD[Completion Queue #] register:
   (a) If not, go to step 2
   (b) If so, go to step 4
4. Set the appropriate TX_TDOWN bit in the USBOTG controller's USB teardown register (TEARDOWN). Set the bit corresponding to the Channel Number within TEARDOWN[TX_TDOWN] field.
5. Flush the TX FIFO in the USB Controller: Set PERI_TXCSR[FLUSHFIFO] for the corresponding Endpoint.
6. Re-enable the Tx DMA channel.
   (a) Clear TXGCRn[TX_TEARDOWN and TX_ENABLE] bit.
   (b) Set TXGCRn[TX_ENABLE] bit.


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.8.3 Queue Manager

The Queue Manager (QM) is a hardware module that is responsible for accelerating management of the queues, i.e. queues are maintained within a queue manager module. Packets are added to a queue by writing the 32-bit descriptor address to a particular memory mapped location in the queue manager module. Packets are de-queued by reading the same location for that particular queue. A single queue manager exists within the subsystem and handles all available 156 queues.

Descriptors are queued(pushed) onto a logical queue by writing the descriptor pointer to the Queue[n] Register D. When the Queue[n] Register D is written, this kicks off the queue manager causing it to add the descriptor to the tail of queue.

The Queue Manager keeps track of the order of the descriptors pushed within a queue and the linking status of the descriptors. To accomplish this linking status, the Queue Manager will first resolve the 32-bit descriptor pointer into a 16-bit index which is used for linking a queue pointer purposes. Once the address to index computation is performed, i.e., the physical index information is determined, the Queue Manager will link that descriptor onto the descriptor chain that is maintained for that logical queue by writing the linking information out to a linking RAM which is external to the queue manager. More on linking RAM discussion will follow in latter sections. The Queue Manager will also update the queue tail pointers appropriately. Since logical queues within the queue manager are maintained using linked lists, queues cannot become full and no check for fullness is required before queuing a packet descriptor.

#### 25.8.3.1 Queue Types

Several types of queues exist (a total of 156 queues) within the CPPI 4.1 DMA. Regardless of the type of queue, queues are used to hold pointers to Packet or Buffer Descriptors while they are being passed between the CPU and / or any of the ports in the USB subsystem.

The following types of Queues exist:

- Transmit Submit Queue
- Transmit Completion Queue
- Receive Submit Queue
- Receive Completion Queue

Table 25-28 displays queue-endpoint assignments.

##### 25.8.3.1.1 Transmit Submission Queue

Transmit ports use transmit submit queues to store the Packet Descriptors that are waiting to be transmitted. Each transmit endpoint has dedicated queues (2 queues per port) that are reserved exclusively for a use by a single endpoint. Multiple queues per port are allocated to facilitate quality of service (QoS) for applications that require QoS.

##### 25.8.3.1.2 Transmit Completion Queue

Transmit ports use transmit Completion Queues to return Packet Descriptor to the CPU after the payload associated with the Packet Descriptors have been transmitted.

Transmit Completion Queues are also used to return Packet Descriptors when performing a Transmit channel teardown operation.

##### 25.8.3.1.3 Receive Submission Queue (Free Descriptor Queue)

Receive endpoints use receive submit queues (also known as free descriptor queues) to forward received packets to the CPU. The entries on the free descriptor queues have pre-attached empty buffers whose size and location are described in the descriptor. The host is required to allocate both the descriptor and buffer and pre-link them prior to adding (submitting) a descriptor to one of the available free descriptor queue.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.8.3.1.4 Receive Completion Queue

Receive ports use receive Completion Queues to return packet descriptors to the port after DMA packets have been received by the CPU and the payload has been unloaded.

Receive channel teardown is not necessary for receive transactions and no channel teardown information nor resource is available

The Queues Assignment within the USB Subsystem are split between Transmit and Receive operations.

- Each USB Endpoint has dedicated transmit queues that are used for DMA Packet Submission and Completion
- Each USB Endpoint has 15 dedicated Receive queues that are used for DMA Packet Completion. An additional pool of 32 queues are allocated to be used as DMA Packet Submission Queues across both USB Modules. Any one of the 32 Receive Submit queues can be used to service any receive endpoint within the USB Subsystem

.

**Table 25-28. Queue-Endpoint Mapping**

| Queue Start Number | Queue Count | Queue-Endpoint Association |
|---|---|---|
| 0 | 32 | USB0/1 Endpoint[n] Submission Queues (Free Descriptor Queues), where n = 1,2,3...15 |
| 32 | 2 | USB0 Endpoint 1 Transmit Submission Queues |
| 34 | 2 | USB0 Endpoint 2 Transmit Submission Queues |
| 36 | 2 | USB0 Endpoint 3 Transmit Submission Queues |
| 38 | 2 | USB0 Endpoint 4 Transmit Submission Queues |
| 40 | 2 | USB0 Endpoint 5 Transmit Submission Queues |
| 42 | 2 | USB0 Endpoint 6 Transmit Submission Queues |
| 44 | 2 | USB0 Endpoint 7 Transmit Submission Queues |
| 46 | 2 | USB0 Endpoint 8 Transmit Submission Queues |
| 48 | 2 | USB0 Endpoint 9 Transmit Submission Queues |
| 50 | 2 | USB0 Endpoint 10 Transmit Submission Queues |
| 52 | 2 | USB0 Endpoint 11 Transmit Submission Queues |
| 54 | 2 | USB0 Endpoint 12 Transmit Submission Queues |
| 56 | 2 | USB0 Endpoint 13 Transmit Submission Queues |
| 58 | 2 | USB0 Endpoint 14 Transmit Submission Queues |
| 60 | 2 | USB0 Endpoint 15 Transmit Submission Queues |
| 62 | 2 | USB1 Endpoint 1 Transmit Submission Queues |
| 64 | 2 | USB1 Endpoint 2 Transmit Submission Queues |
| 66 | 2 | USB1 Endpoint 3 Transmit Submission Queues |
| 68 | 2 | USB1 Endpoint 4 Transmit Submission Queues |
| 70 | 2 | USB1 Endpoint 5 Transmit Submission Queues |
| 72 | 2 | USB1 Endpoint 6 Transmit Submission Queues |
| 74 | 2 | USB1 Endpoint 7 Transmit Submission Queues |
| 76 | 2 | USB1 Endpoint 8 Transmit Submission Queues |
| 78 | 2 | USB1 Endpoint 9 Transmit Submission Queues |
| 80 | 2 | USB1 Endpoint 10 Transmit Submission Queues |
| 82 | 2 | USB1 Endpoint 11 Transmit Submission Queues |
| 84 | 2 | USB1 Endpoint 12 Transmit Submission Queues |
| 86 | 2 | USB1 Endpoint 13 Transmit Submission Queues |
| 88 | 2 | USB1 Endpoint 14 Transmit Submission Queues |
| 90 | 2 | USB1 Endpoint 15 Transmit Submission Queues |
| 92 | 1 | Reserved |
| 93 | 1 | USB0 Tx Endpoint 1 Completion Queue |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-28. Queue-Endpoint Mapping (continued)**

| Queue Start Number | Queue Count | Queue-Endpoint Association |
|---|---|---|
| 94 | 1 | USB0 Tx Endpoint 2 Completion Queue |
| 95 | 1 | USB0 Tx Endpoint 3 Completion Queue |
| 96 | 1 | USB0 Tx Endpoint 4 Completion Queue |
| 97 | 1 | USB0 Tx Endpoint 5 Completion Queue |
| 98 | 1 | USB0 Tx Endpoint 6 Completion Queue |
| 99 | 1 | USB0 Tx Endpoint 7 Completion Queue |
| 100 | 1 | USB0 Tx Endpoint 8 Completion Queue |
| 101 | 1 | USB0 Tx Endpoint 9 Completion Queue |
| 102 | 1 | USB0 Tx Endpoint 10 Completion Queue |
| 103 | 1 | USB0 Tx Endpoint 11 Completion Queue |
| 104 | 1 | USB0 Tx Endpoint 12 Completion Queue |
| 104 | 1 | USB0 Tx Endpoint 13 Completion Queue |
| 106 | 1 | USB0 Tx Endpoint 14 Completion Queue |
| 107 | 1 | USB0 Tx Endpoint 15 Completion Queue |
| 108 | 1 | Reserved |
| 109 | 1 | USB0 Rx Endpoint 1 Completion Queue |
| 110 | 1 | USB0 Rx Endpoint 2 Completion Queue |
| 111 | 1 | USB0 Rx Endpoint 3 Completion Queue |
| 112 | 1 | USB0 Rx Endpoint 4 Completion Queue |
| 113 | 1 | USB0 Rx Endpoint 5 Completion Queue |
| 114 | 1 | USB0 Rx Endpoint 6 Completion Queue |
| 115 | 1 | USB0 Rx Endpoint 7 Completion Queue |
| 116 | 1 | USB0 Rx Endpoint 8 Completion Queue |
| 117 | 1 | USB0 Rx Endpoint 9 Completion Queue |
| 118 | 1 | USB0 Rx Endpoint 10 Completion Queue |
| 119 | 1 | USB0 Rx Endpoint 11 Completion Queue |
| 120 | 1 | USB0 Rx Endpoint 12 Completion Queue |
| 121 | 1 | USB0 Rx Endpoint 13 Completion Queue |
| 122 | 1 | USB0 Rx Endpoint 14 Completion Queue |
| 123 | 1 | USB0 Rx Endpoint 15 Completion Queue |
| 124 | 1 | Reserved |
| 125 | 1 | USB1 Tx Endpoint 1 Completion Queue |
| 126 | 1 | USB1 Tx Endpoint 2 Completion Queue |
| 127 | 1 | USB1 Tx Endpoint 3 Completion Queue |
| 128 | 1 | USB1 Tx Endpoint 4 Completion Queue |
| 129 | 1 | USB1 Tx Endpoint 5 Completion Queue |
| 130 | 1 | USB1 Tx Endpoint 6 Completion Queue |
| 131 | 1 | USB1 Tx Endpoint 7 Completion Queue |
| 132 | 1 | USB1 Tx Endpoint 8 Completion Queue |
| 133 | 1 | USB1 Tx Endpoint 9 Completion Queue |
| 134 | 1 | USB1 Tx Endpoint 10 Completion Queue |
| 135 | 1 | USB1 Tx Endpoint 11 Completion Queue |
| 136 | 1 | USB1 Tx Endpoint 12 Completion Queue |
| 137 | 1 | USB1 Tx Endpoint 13 Completion Queue |
| 138 | 1 | USB1 Tx Endpoint 14 Completion Queue |
| 139 | 1 | USB1 Tx Endpoint 15 Completion Queue |
| 140 | 1 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-28. Queue-Endpoint Mapping (continued)**

| Queue Start Number | Queue Count | Queue-Endpoint Association |
|---|---|---|
| 141 | 1 | USB1 Rx Endpoint 1 Completion Queue |
| 142 | 1 | USB1 Rx Endpoint 2 Completion Queue |
| 143 | 1 | USB1 Rx Endpoint 3 Completion Queue |
| 144 | 1 | USB1 Rx Endpoint 4 Completion Queue |
| 145 | 1 | USB1 Rx Endpoint 5 Completion Queue |
| 146 | 1 | USB1 Rx Endpoint 6 Completion Queue |
| 147 | 1 | USB1 Rx Endpoint 7 Completion Queue |
| 148 | 1 | USB1 Rx Endpoint 8 Completion Queue |
| 149 | 1 | USB1 Rx Endpoint 9 Completion Queue |
| 150 | 1 | USB1 Rx Endpoint 10 Completion Queue |
| 151 | 1 | USB1 Rx Endpoint 11 Completion Queue |
| 152 | 1 | USB1 Rx Endpoint 12 Completion Queue |
| 153 | 1 | USB1 Rx Endpoint 13 Completion Queue |
| 154 | 1 | USB1 Rx Endpoint 14 Completion Queue |
| 155 | 1 | USB1 Rx Endpoint 15 Completion Queue |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.8.4 Memory Regions and Linking RAM

To accomplish the linking of submitted descriptors, the queue manager will first resolve a 32-bit descriptor pointer into a 16-bit index which is used for linking and queue pointer purposes. Once the physical index information is determined, the queue manager will link that descriptor onto the descriptor chain that is maintained for that logical queue by writing the linking information out to a linking RAM which is external to the queue manager. The queue manager will also update the queue tail pointer appropriately. Since logical queues within the queue manager are maintained using linked lists, queues cannot become full and no check for fullness is required before queuing a packet descriptor.

In addition to allocating memory for Descriptors and associated buffers, the CPU is responsible for allocating additional memory for exclusive use of the CPPI DMA Queue Manager to be used as a scratch pad RAM. The queue manager uses this memory to manage states of descriptors submitted within the submit queues. The allocated memory can be a single contiguous block of memory or two separate blocks of memory. These blocks of memory are referred to as a Linking RAM Regions and should not be confused with memory regions that are used to store descriptors. That is, the use of the term *region* should be used in the context of its use.

The actual physical size of the Linking RAM region(s) to be allocated depends on the total number of descriptors defined within all memory regions. A minimum of four bytes of memory needs to be allocated for each Descriptor defined within all 16 memory regions.

The queue manager has the capability of managing up to 16 memory regions. These memory regions are used to store descriptors of variable sizes. The total number of descriptors that can be managed by the queue manager should not exceed 64K. Each memory region has descriptors of one configurable size. Descriptors with different sizes cannot be housed within a single memory region. These 64K descriptors are internally referenced in the queue manager using a 16-bit quantity index.

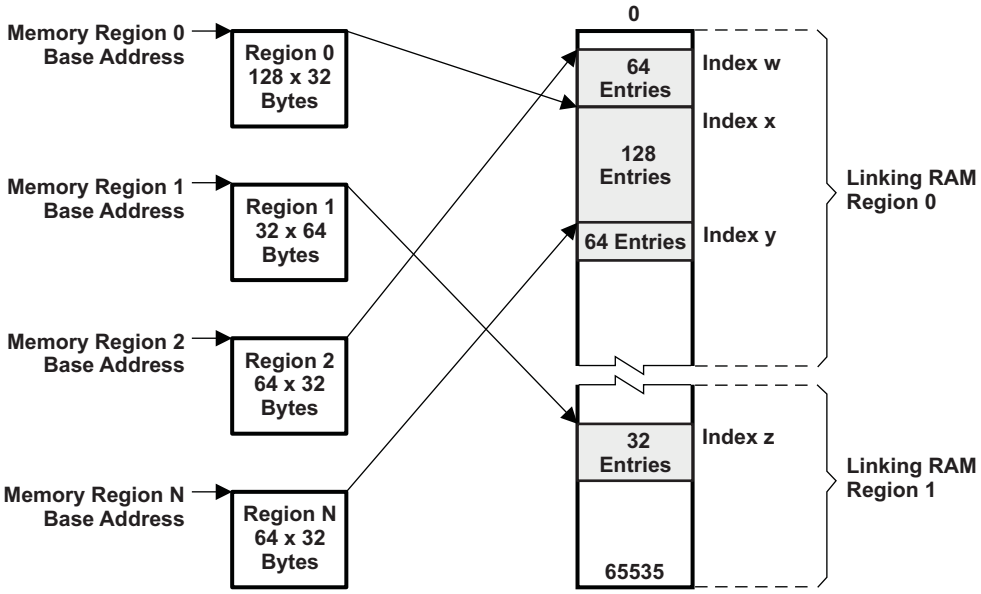
The information about the Linking RAM regions and the size that are allocated is communicated to the CPPI DMA via three registers dedicated for this purpose. Two of the three registers are used to store the 32-bit aligned start addresses of the Linking RAM regions. The remaining one register is used to store the size of the first Linking RAM. The linking RAM size value stored here is the number of descriptors that is to be managed by the queue manager within a particular Linking RAM region not the physical size of the buffer, which is four times the number of descriptors.

Note that application is not required to use both linking RAM regions, if the size of the Linking RAM for the first region is large enough to accommodate all descriptors defined. No linking RAM size register for linking RAM region 1 exists. The size of the second linking RAM, when used, is indirectly computed from the total number of descriptors defined less the number of descriptors managed by the first linking RAM

Figure 25-17 displays the relationship between several memory regions and linking RAM.


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 25-17. Relationship Between Memory Regions and Linking RAM**




### 25.8.5 Zero Length Packets

A special case is the handling of null packets with the DMA controller. Upon receiving a zero length USB packet, the XFER DMA will send a data block to the DMA controller with byte count of zero and the zero byte packet bit of INFO Word 2 set. The DMA controller will then perform normal End of Packet termination of the packet, without transferring data.

If a zero-length USB packet is received, the XDMA will send the CDMA a data block with a byte count of 0 and this bit set. The CDMA will then perform normal EOP termination of the packet without transferring data. For transmit, if a packet has this bit set, the XDMA will ignore the CPPI packet size and send a zero-length packet to the USB controller.

### 25.8.6 CPPI DMA Scheduler

The CPPI DMA scheduler is responsible for controlling the rate and order between the different Tx and Rx threads/transfers that are provided in the CPPI DMA controller. The scheduler table RAM exists within the scheduler.

The DMA controller maintains state information for each of the endpoints which allows packet segmentation and reassembly operations to be time division multiplexed between channels in order to share the underlying DMA hardware. A DMA scheduler is used to control the ordering and rate at which this multiplexing occurs.

#### 25.8.6.1 CPPI DMA Scheduler Operation

Once the scheduler is enabled it will begin processing the entries in the table. When appropriate the scheduler will pass credits to the DMA controller to perform a Tx or Rx operation. The operation of the DMA controller is as follows:

1. After the DMA scheduler is enabled it begins with the table index set to 0.
2. The scheduler reads the entry pointed by the index and checks to see if the channel in question is currently in a state where a DMA operation can be accepted.
   (a) The DMA channel must be enabled AND
   (b) The CPPI FIFO that the channel talks to has free space on TX (FIFO full signal is not asserted) or a valid block on Rx (FIFO empty signal is not asserted)
3. If the DMA channel is capable of processing a credit to transfer a block, the DMA scheduler will issue that credit via the DMA scheduling interface which is a point to point connection between the DMA


Copyright © 2011–2016, Texas Instruments Incorporated

Scheduler and the DMA Controller.

(a) The DMA controller may not be ready to accept the credit immediately and is provided a sched_ready signal which is used to stall the scheduler until it can accept the credit. The DMA controller only asserts the sched_ready signal when it is in the IDLE state.

(b) Once a credit has been accepted (indicated by sched_req and sched_ready both asserted), the scheduler will increment the index to the next entry and will start again at step 2.

4. If the channel in question is not currently capable of processing a credit, the scheduler will increment the index in the scheduler table to the next entry and will start at step 2.
5. Note that when the scheduler attempts to increment its index, to the value programmed in the table size register, the index will be reset to 0.

#### *25.8.6.1.1 CPPI DMA Scheduler Initialization*

Before the scheduler can be used, the host is required to initialize and enable the block. This initialization is performed as follows:

1. The Host initializes entries within an internal memory array in the scheduler. This array contains up to 256 entries (4 entries per table word n where n=0-63) and each entry consists of a DMA channel number and a bit indicating if this is a Tx or Rx opportunity. These entries represent both the order and frequency that various Tx and Rx channels will be processed. A table size of 256 entries allows channel bandwidth to be allocated with a maximum precision of 1/256th of the total DMA bandwidth. The more entries that are present for a given channel, the bigger the slice of the bandwidth that channel will be given. Larger tables can be accommodated to allow for more precision. This array can only be written by the Host, it cannot be read.
2. If the application does not need to use the entire 256 entries, firmware can initialize the portion of the 256 entries and indicate the size of the entries used by writing onto an internal register in the scheduler which sets the actual size of the array (it can be less than 256 entries).
3. The host writes an internal register bit to enable the scheduler. The scheduler is not required to be disabled in order to change the scheduler array contents.

#### *25.8.6.1.2 CPPI DMA Scheduler Programming Examples*

Consider a three endpoints use on a system with the following configurations: EP1-Tx, EP2-Rx, and EP2-Tx. Two assumptions are considered:

**Case 1:** Assume that you would like to service each enabled endpoints (EP1-Tx, EP2-Rx, and EP2-Tx) with equal priority.

The scheduler handles the rate at which an endpoint is serviced by the number of credits programmed (entries) for that particular endpoint within the scheduler Table Words. The scheduler has up to 256 credits that it can grant and for this example the user can configure the number of entries/credits to be anywhere from 3 to 256 with a set of 3 entries.

However, the optimum and direct programming for this scenario would be programming only the first three entries of the scheduler via scheduler Table WORD[0]. Since this case expects the Scheduler to use only the first three entries, you communicate that by programming DMA_SCHED_CTRL.LAST_ENTRY with 2 (that is, 3 -1). The Enabled Endpoint numbers and the data transfer direction is then communicated by programming the first three entries of WORD[0] (ENTRY0_CHANNEL = 1: ENTRY0_RXTX = 0; ENTRY1_CHANNEL = 2: ENTRY1_RXTX = 1;ENTRY2_CHANNEL = 2: ENTRY2_RXTX = 0). With this programming, the Scheduler will only service the first three entries in a round-robin fashion, checking each credited endpoint for transfer one after the other, and servicing the endpoint that has data to transfer.

**Case 2:** Enabled endpoint EP1-Tx is serviced at twice the rate as the other enabled endpoints (EP2-Rx and EP2-Tx).


Copyright © 2011–2016, Texas Instruments Incorporated

The number of entries/credit that has to be awarded to EP1-Tx has to be twice as much of the others. Four entries/credits would suffix to satisfy our requirement with two credits for EP1-Tx, one credit for EP2-Rx, and one credit for EP2-Tx. This requirement is satisfied by allocating any two of the four entries to EP1-Tx endpoint. Again for this example, scheduler Table WORD[0] would suffice since it can handle the first 4 entries. Even though several scenarios exist to programming the order of service for this case, one scenario would be to allow servicing EP1-Tx to back-to-back followed by the other enabled endpoints. Program DMA_SCHED_CTRL.LAST_ENTRY with 3 (that is, 4 -1). Program WORD[0] (ENTRY0_CHANNEL = 1: ENTRY0_RXTX = 0; ENTRY1_CHANNEL = 1: ENTRY1_RXTX = 0; ENTRY2_CHANNEL = 2: ENTRY2_RXTX = 1; ENTRY3_CHANNEL = 2: ENTRY3_RXTX = 0).

### *25.8.7 CPPI DMA State Registers*

The port must store and maintain state information for each transmit and receive port/channel. The state information is referred to as the Tx DMA State and Rx DMA State.

#### 25.8.7.1 Transmit DMA State Registers

The Tx DMA State is a combination of control fields and protocol specific port scratchpad space used to manipulate data structures and transmit packets. Each transmit Endpoint has two queues. Each queue has a one head descriptor pointer and one completion pointer. There are thirty Tx DMA State registers; one for each port.

The following information is stored in the Tx DMA State:

- Tx Queue Head Descriptor Pointer(s)
- Tx Completion Pointer(s)
- Protocol specific control/status (port scratchpad)

#### 25.8.7.2 Receive DMA State Registers

The Rx DMA State is a combination of control fields and protocol specific port scratchpad space used to manipulate data structures in order to receive packets. Each receive channel has only one queue. Each channel queue has one head descriptor pointer and one completion pointer. There are thirty Rx DMA State registers; one for each port/channel.

The following information is stored in the Rx DMA State:

- Rx Queue Head Descriptor Pointer
- Rx Queue Completion Pointer
- Rx Buffer Offset

### *25.8.8 CPPI DMA Protocols Supported*

Four different type of DMA transfers protocols are supported by the DMA; Transparent, RNDIS, Generic RNDIS, and Linux CDC. The following sections will outline the details on these DMA transfer types.

#### 25.8.8.1 Transparent DMA Transfer

Transparent Mode DMA operation is the default DMA mode where DMA interrupt is generated whenever a DMA packet is transferred. In the transparent mode, DMA packet size cannot be greater than USB MaxPktSize for the endpoint. This transfer type is ideal for transfer (not packet) sizes that are less than a max packet size.

##### *25.8.8.1.1 Transparent DMA Transfer Setup*

The following will configure all 30 ports for Transparent DMA Transfer type.

Make sure that RNDIS Mode is disabled globally. This allows the application to configure the CPPI DMA protocol in use to be configured per endpoint need. To disable RNDIS operation, clear RNDIS bit in the USB Control Registers corresponding to the USB Module (default setting), that is, CTRL0[RNDIS] = 0 and CTRL1[RNDIS] = 0.


Copyright © 2011–2016, Texas Instruments Incorporated

Configure the USB0/1 Tx/Rx DMA Mode Registers (USB0/1 TX(RX)MODE0/1) for the Endpoint field in use is programmed for Transparent Mode, that is, TXMODE0/1[TXn_MODE] = 00b and RXMODE0/1[RXn_MODE] = 00b.

#### 25.8.8.2 RNDIS DMA Transfer

RNDIS mode DMA is used for large transfers (that is, total data size to be transferred is greater than USB MaxPktSize where the MaxPktSize is a multiple of 64 bytes) that requires multiple USB packets. This is accomplished by breaking the larger packet into smaller packets, where each packet size being USB MaxPktSize except the last packet where its size is less than USB MaxPktSize, including zero bytes. This implies that multiple USB packets of MaxPktSize will be received and transferred together as a single large DMA transfer and the DMA interrupt is generated only at the end of the complete reception of DMA transfer. The protocol defines the end of the complete transfer by receiving a short USB packet (smaller than USB MaxPktSize as mentioned in USB specification 2.0). If the DMA packet size is an exact multiple of USB MaxPktSize, the DMA controller waits for a zero byte packet at the end of complete transfer to signify the completion of the transfer.

**NOTE:** RNDIS Mode DMA is supported only when USB MaxPktSize is an integral multiple of 64 bytes.

##### *25.8.8.2.1 RNDIS DMA Transfer Setup*

If all endpoints in use are desired to operate in RNDIS mode, it is only suffix to configure RNDIS DMA operation in RNDIS mode at the global level and application can ignore individual endpoint DMA mode configuration. This is achieved by programming CTRLR0/1[RNDIS] with a '1.'

However if DMA mode is required to be configured at the endpoint level, it is required to disable the use of RNDIS at the global level, this is achieved by clearing RNDIS bit field (CTRLR0/1[RNDIS] = 0), since global configuration over-rides endpoint configuration.

To configure RNDIS DMA mode use, configure the field that correspond to the USB module endpoint using the corresponding USB0/1 TX(RX) Mode Register, that is, TXMODE0/1[TXn_MODE] = 01b and RXMODE0/1[RXn_MODE] = 01b

#### 25.8.8.3 Generic RNDIS DMA Transfer

Generic RNDIS DMA transfer mode is identical to the normal RNDIS mode in nearly all respects, except for the exception case where the last packet of the transfer can either be a short packet or the MaxPktSize. When the last packet size is equal to the MaxPktSize then no additional zero-byte packet is sent when using Generic RNDIS transfer. Generic RNDIS transfer makes use of a USB0/1 GENERIC RNDIS EPn Size register that must be programmed with the transfer size (in bytes) of the transfer for the USB Module (USB0 or USB1) prior to posting a transfer transaction. If the transfer size is an integer multiple of USB MaxPktSize then no additional zero-byte packet is sent when using Generic RNDIS transfer. However, if the a short packet has been sent prior to programmed size count, the transfer would end in a similar fashion an RNDIS transfer would behave. For example, if the USB MaxPktSize (Tx/RxMaxP) is programmed with a value of 64, the Generic RNDIS EP Size register for that endpoint must be programmed with a value that is an integer multiple of 64 (for example, 64, 128, 192, 256, etc.) for it behave differently than RNDIS transfer. In other words, when using Generic RNDIS mode and the DMA is tasked to transfer data transfer size that is less than or equal to the size value programmed within the USB0/1 GENERIC RNDIS EPn Size register.

This means that Generic RNDIS mode will perform data transfer in the same manner as RNDIS mode, closing the CPPI packet if a USB packet with a size less than the USB MaxPktSize size value is received. Otherwise, the packet will be closed when the value in the Generic RNDIS EP Size register is reached.

Using USB0/1 GENERIC RNDIS EPn Size register, a packet of up to 64K bytes (65536 bytes) can be transferred. This is to allow the host software to program the USB module to transfer data that is an exact multiple of the USB MaxPktSize (Tx/RxMaxP programmed value) without having to send an additional short packet to terminate.

**NOTE:** As in RNDIS mode, the USB max packet size (Tx/RxMaxp programmed value) of any Generic RNDIS mode enabled endpoints must be a multiple of 64 bytes. Generic RNDIS acceleration should not be enabled for endpoints where the max packet size is not a multiple of 64 bytes. Only transparent mode should be used for such endpoints.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.8.8.3.1 Generic RNDIS DMA Transfer Setup

Disable the use of RNDIS at the global level, this is achieved by clearing RNDIS bit field (CTRLR0/1[RNDIS]=0), since global configuration over-rides endpoint configuration.

Configure the field that correspond to the USB module endpoint using the corresponding USB0/1 TX(RX) Mode Register, that is, TXMODE0/1[TXn_MODE] = 11b and RXMODE0/1[RXn_MODE] = 11b.

### 25.8.8.4 Linux CDC DMA Transfer

Linux CDC DMA transfer mode acts in the same manner as RNDIS packets, except for the case where the last data matches the max USB packet size requiring additional zero-byte packet transfer in RNDIS mode. If the last data packet of a transfer is a short packet where the data size is greater than zero and less the USB MaxPktSize, then the behavior of the Linux CDC DMA transfer type is identical with the RNDIS DMA transfer type. The only exception is when the short packet length terminating the transfer is a Null Packet. In this case, instead of transferring the Null Packet, it will transfer a data packet of size 1 byte with the data value of 00h.

In transmit operation, if an endpoint is configured or CDC Linux mode, upon receiving a Null Packet from the CPPI DMA, the XFER DMA will then generate a packet containing 1 byte of data, whose value is 00h, indicating the end of the transfer. During receive operation, the XFER DMA will recognize the one byte zero packet as a termination of the data transfer, and sends a block of data with the EOP indicator set and a byte count of one to the CPPI DMA controller. The CPPI DMA realizing the end of the transfer termination will not update/increase the packet size count of the Host Packet Descriptor.

#### 25.8.8.4.1 Linux CDC DMA Transfer Setup

Disable the use of RNDIS at the global level, this is achieved by clearing RNDIS bit field (CTRLR0/1[RNDIS]=0), since global configuration over-rides endpoint configuration.

Configure the field that correspond to the USB module endpoint using the corresponding USB0/1 TX(RX) Mode Register, that is, TXMODE0/1[TXn_MODE] = 10b and RXMODE0/1[RXn_MODE] = 10b.


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.8.9 USB Data Flow Using DMA

The necessary steps required to perform a USB data transfer using the CPPI 4.1 DMA is expressed using an example for both transmit and receive cases. Assume USB0 is ready to perform a USB data transfer of size 608 bytes (see Figure 25-18).

The 608 bytes data to be transferred is described using three buffer descriptors with a buffer size of 256 bytes.

The following nomenclature is Defined:

**PD** — Host Packet Descriptor

**PPD** — Pointer to Host Packet Descriptor

**BD** — Buffer Descriptor or Host Buffer Descriptor

**PBD** — Pointer to Host Buffer Descriptor

**DB** — Data Buffer Size of 256 Bytes

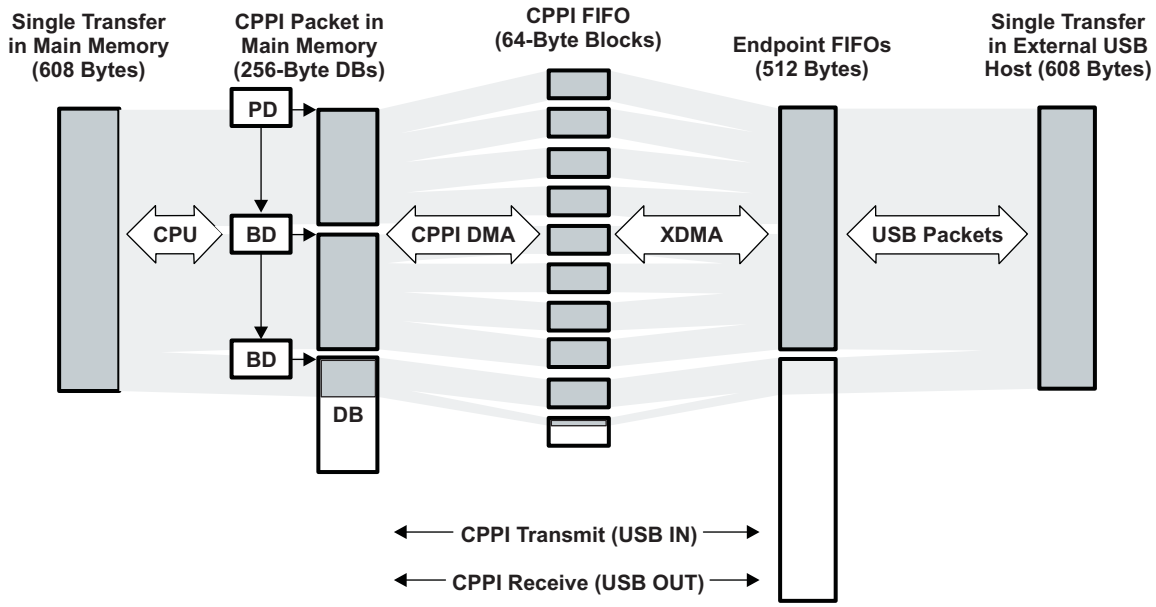
**TXSQ** — Transmit Submit Queue

**TXCQ** — Transmit Completion Queue

**RXCQ** — Receive Completion Queue

**RXSQ** — Receive Submit Queue (Receive Free/Buffer Descriptor Queue)

**Figure 25-18. High-level Transmit and Receive Data Transfer Example**




**Example assumptions:**

- The CPPI data buffers are 256 bytes in length.
- USB0 module is to be used to perform this transfer. Note that the steps required is similar for USB1 use.
- The USB0 endpoint 1 Tx and Rx endpoint 1 size are programmed to handle max USB packet size of 512 bytes.
- A single transfer length is 608 bytes.
- The SOP offset is 0.

The above translates to the following multi-descriptor setup:


Copyright © 2011–2016, Texas Instruments Incorporated

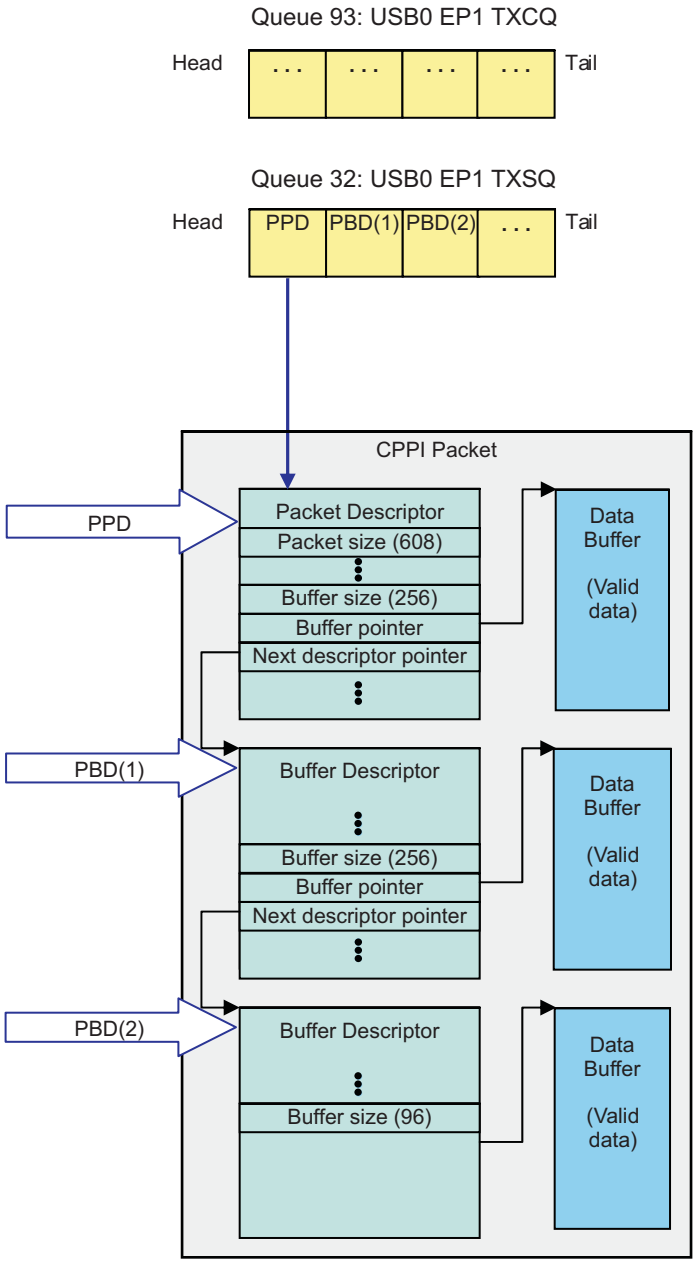
### 25.8.9.1 Transmit USB Data Flow Using DMA

The transmit descriptors and queue status configuration prior to the transfer taking place is shown in Figure 25-19.

**Transmit Case**

- One packet descriptor with DMA packet length field of 608 bytes and a data Buffer of size 256 Bytes linked to the 1st buffer descriptor.
- One buffer descriptor with buffer size of 256 bytes.
- One buffer descriptor with a data buffer size of 96 bytes.

**Figure 25-19. Transmit Descriptors and Queue Status Configuration**





Copyright © 2011–2016, Texas Instruments Incorporated

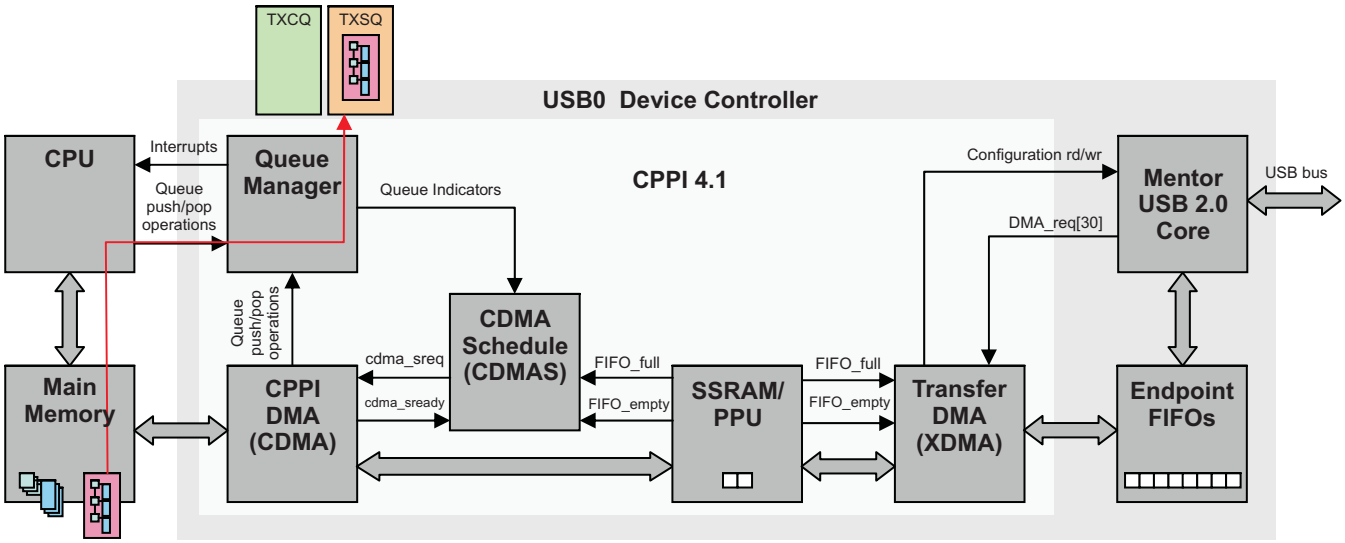
#### 25.8.9.1.1 Transmit Initialization (Step 1)

The CPU performs the following steps for transmit initialization:

1. Initializes Memory Region 0 base address and Memory Region 0 size, Link RAM0 Base address, Link RAM0 data size, and Link RAM1 Base address.
2. Creates Packet Descriptors, Buffer Descriptors , and data buffers in main memory and link as indicated in Figure 25-20.
3. Initializes and configures the Queue Manager, Channel Setup, DMA Scheduler, and the USB Controller.
4. Adds (pushes) the Pointer to the Packet Descriptor and the two Pointers to the Buffer Descriptors to the Transmit Submit Queue by writing the Packet Descriptor address to the Transmit Submit Queue Control D Register.

Figure 25-20 captures the Buffer Descriptor/Data Buffer pair in main memory and later submitted within the Transmit Submit Queue.

**Figure 25-20. Transmit USB Data Flow Example (Initialization)**




#### 25.8.9.1.2 CDMA and XDMA Transfer Packet Data Into Endpoint FIFO (Step 2)

The steps for CDMA and XDMA to transfer packet data into endpoint FIFO are as follows:

1. The Queue Manager informs the CDMA Scheduler that the Transmit Submit Queue is not empty.
2. CDMA Scheduler checks that the CPPI FIFO FIFO_full is not asserted, then issues a credit to the CDMA.
3. CDMA reads the Packet Descriptor pointer (PPD) and descriptor size from the queue manager
4. For each 64-byte block of data in the DMA Packet Descriptor payload
   (a) The CDMA transfers a max burst of 64-byte block from the data to be transferred in main memory to the CPPI FIFO
   (b) The XDMA sees FIFO_empty not asserted and transfers 64-byte block from CPPI FIFO to USB Controller Endpoint FIFO.
5. The CDMA reads the first Buffer Descriptor Pointer .
6. For each 64-byte block of data in the Buffer Descriptor payload:
   (a) The CDMA transfers a max burst of 64-byte block from the data to be transferred in main memory to the CPPI FIFO.
   (b) The XDMA sees FIFO_empty not asserted and transfers 64-byte block from CPPI FIFO to Endpoint FIFO.


Copyright © 2011–2016, Texas Instruments Incorporated

7. The CDMA reads the second Buffer Descriptor Pointer
8. For each 64-byte block of data in the Buffer Descriptor Payload:
   (a) The CDMA transfers a max burst of 64-byte block from the data to be transferred in main memory to the CPPI FIFO.
   (b) The XDMA sees FIFO_empty not asserted and transfers 64-byte block from CPPI FIFO to Endpoint FIFO.
   (c) The CDMA transfers the last remaining 32-byte from the data to be transferred in main memory to the CPPI FIFO.
   (d) The XDMA sees FIFO_empty not asserted and transfers 32-byte block from CPPI FIFO to Endpoint FIFO.

#### *25.8.9.1.3 USB 2.0 Core Transmits USB Packets for Tx (Step 3)*

1. Once the XDMA has transferred enough 64-byte blocks of burst of data from the CPPI FIFO to fill the Endpoint FIFO (accumulated 512 bytes), it signals the USB 2.0 Controller that a TX packet is ready (sets the endpoint's TxPktRdy bit).
2. The USB 2.0 Controller will transmit the packet from the Endpoint FIFO out on the USB BUS when it receives a corresponding IN request from the attached USB Host.
3. After the USB packet is transferred, the USB 2.0 Controller issues a TX DMA_req to the XDMA.
4. This process is repeated until (the above steps '1', '2', and '3') the entire packet has been transmitted. The XDMA will also generate the required termination packet depending on the termination mode configured for the endpoint.

#### *25.8.9.1.4 Return Packet to Completion Queue and Interrupt CPU for Tx (Step 4)*

1. After all data for the packet has been transmitted (as specified by the packet size field), the CDMA will write the pointer of the Packet Descriptor only to the TX Completion Queue specified in the return queue manager / queue number fields of the packet descriptor.
2. The Queue Manager then indicates the status of the TXSQ (empty) to the CDMAS and the TXCQ to the CPU via an interrupt.

### 25.8.9.2 Receive USB Data Flow Using DMA

The receive buffer descriptors and queue status configuration prior to the transfer taking place is shown in Figure 25-21.

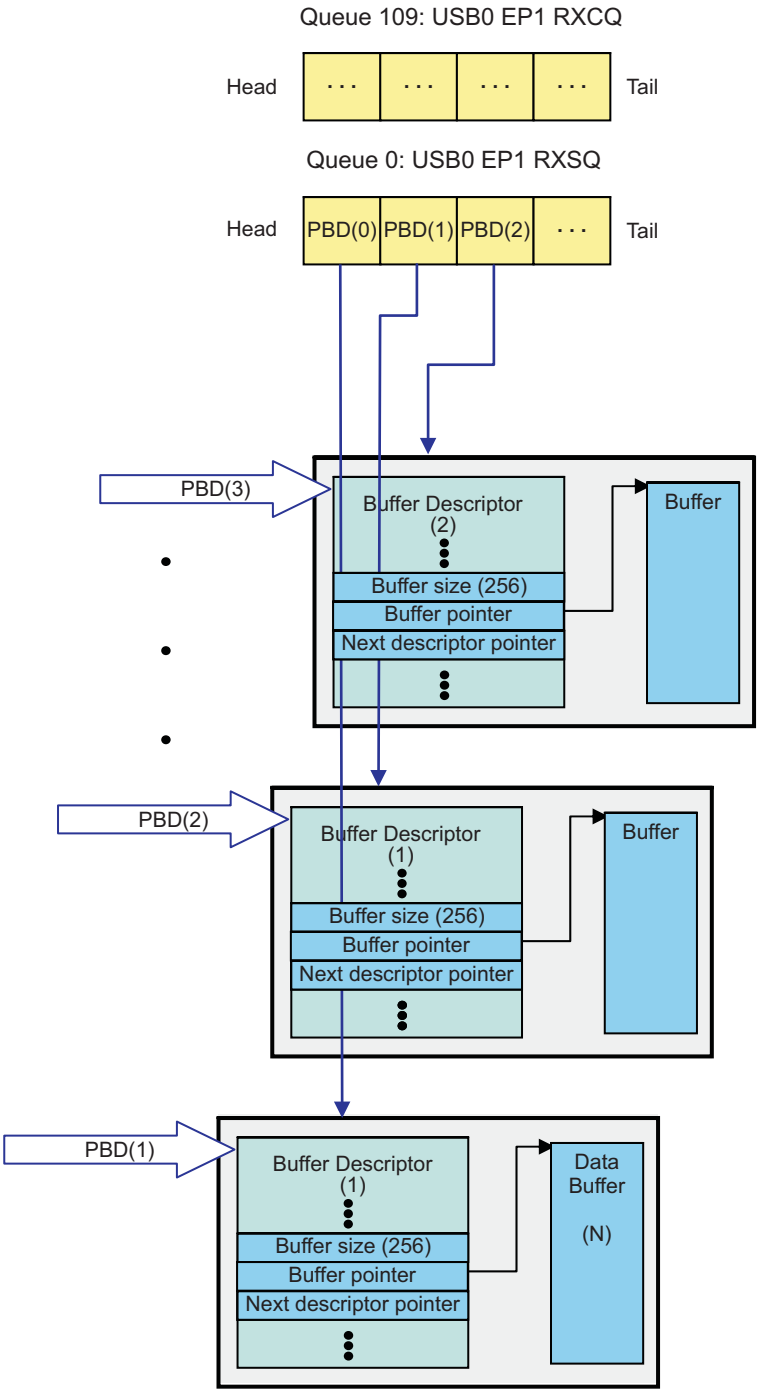
**Receive Case**

- Two buffer descriptors with a 256 byte data buffer
- One buffer descriptor with a 96 byte data buffer

For this example since each data buffer size is 256 bytes, we would require a minimum of three descriptors that would define data buffer size of 608 bytes. The receive setup is as follows:


Copyright © 2011–2016, Texas Instruments Incorporated

**Figure 25-21. Receive Buffer Descriptors and Queue Status Configuration**



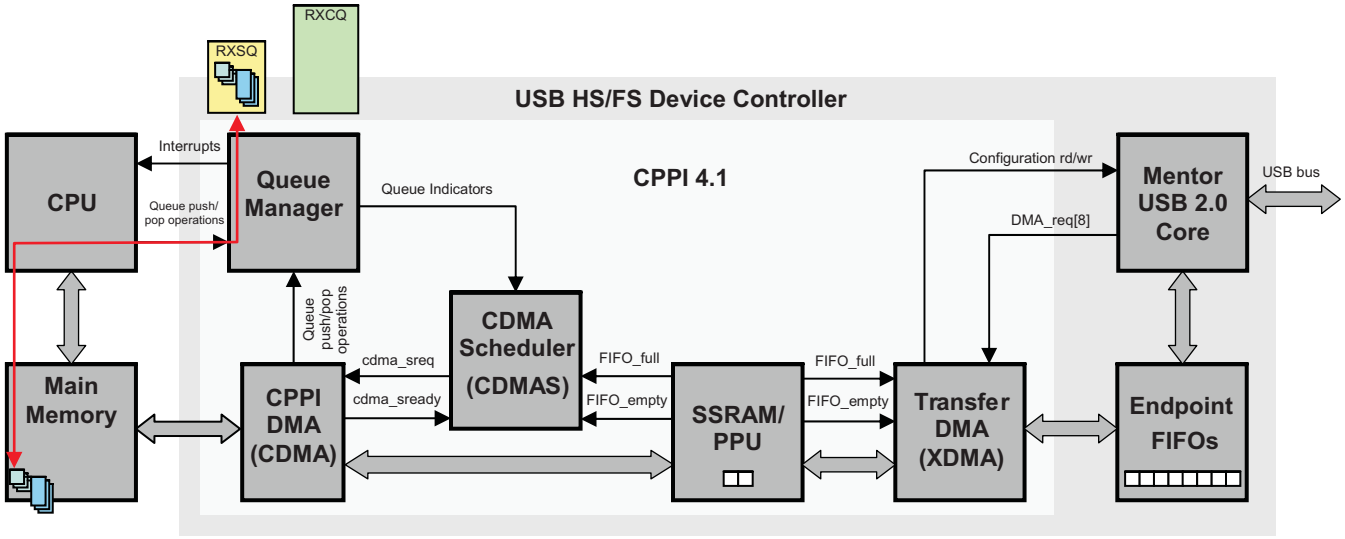

#### *25.8.9.2.1 Receive Initialization (Step 1)*

1. The CPU initializes Queue Manager with the Memory Region 0 base address and Memory Region 0 size, Link RAM0 Base address, Link RAM0 data size, and Link RAM1 Base address.
2. The CPU creates Buffer Descriptors, and data buffers in main memory and link them.
3. CPU then initializes the Receive Completion Queue and configures the Queue Manager, Channel Setup, and DMA Scheduler.
4. The CPU then adds (pushes) the pointers of three Buffer Descriptors into the Receive Submit Queue by writing onto CTRL D register.


Copyright © 2011–2016, Texas Instruments Incorporated

Figure 25-22 displays receive operation Initialization.

**Figure 25-22. Receive USB Data Flow Example (Initialization)**




### *25.8.9.2.2 USB 2.0 Core Receives a Packet, XDMA Starts Data Transfer for Receive (Step 2)*

1. The USB 2.0 Controller receives a USB packet from the USB Host and stores it in the Controller Endpoint FIFO.
2. It then asserts a DMA_req to the XDMA informing it that data is available in the Endpoint FIFO.
3. The XDMA verifies the corresponding CPPI FIFO is not full via the FIFO_full signal, then starts transferring 64-byte data blocks from the Endpoint FIFO into the CPPI FIFO.

### *25.8.9.2.3 CDMA Transfers Data from SSRAM / PPU to Main Memory for Receive (Step 3)*

1. The CDMA Scheduler see FIFO_empty de-asserted (there is RX data in the FIFO) and issues a transaction credit to the CDMA.
2. The CDMA begins packet reception by fetching the first Pointer to the Packet Descriptor from the Queue Manager using the Receive Submit Queue index that was initialized in the RX port DMA state for that channel.
3. The CDMA will then begin writing the 64-byte block of packet data into this Data Buffer.
4. The CDMA will continue filling the buffer with additional 64-byte blocks of data from the CPPI FIFO and will fetch additional pointers to Buffer Descriptors as needed using the Receive Submit Queue 1, 2, and 3 indexes for the 2nd, 3rd, and remaining buffers in the packet. After each buffer is filled, the CDMA writes the buffer descriptor to main memory.

### *25.8.9.2.4 CDMA Completes the Packet Transfer for Receive (Step 4)*

1. After the entire DMA packet has been received, the CDMA writes the packet descriptor to main memory.
2. The CDMA then writes the packet descriptor to the Receive Completion Queue specified in the Queue Manager / Queue Number fields in the RX Global Configuration Register.
3. The Queue Manager then indicates the status of the Receive Completion Queue to the CPU via an interrupt.
4. The CPU can then process the received packet by popping the received packet information from the Receive Submit Queue and accessing the packet's data from main memory.


Copyright © 2011–2016, Texas Instruments Incorporated

## 25.9 Registers

The USB20OTG_REGS submodule contains the MMRs that are used in the submodules within the USB controller. The complete list of registers is listed in the following sections.

The registers required to access and control the USBSS are partitioned in blocks based on their functions. Table 25-29 lists the starting base offset and size of the USBSS and its submodules.

The USBSS base address is 4740 0000h.

**Table 25-29. USBSS Submodule Base Addresses and Size**

| Submodule Name | Submodule Base Address Offset | Size (Kbytes) |
|---|---|---|
| USBSS registers | 0000 0000h | 1000h |
| USB0 controller registers | 0000 1000h | 800h |
| USB1 controller registers | 0000 1800h | 800h |
| CPPI DMA controller registers | 0000 2000h | 1000h |
| CPPI DMA scheduler registers | 0000 3000h | 1000h |
| CPI DMA Queue Manager registers | 0000 4000h | 4000h |

### *25.9.1 USBSS Registers*

The USBSS registers contain the registers that are used to control at the global level and apples to all submodules. Table 25-30 lists the registers for the USBSS submodule.

**Table 25-30. USBSS Registers**

| Address Offset | Acronym | USBSS Register |
|---|---|---|
| 0h | REVREG | USBSS REVISION |
| 10h | SYSCONFIG | USBSS SYSCONFIG |
| 20h | EOI | USBSS IRQ_EOI |
| 24h | IRQSTATRAW | USBSS IRQ_STATUS_RAW |
| 28h | IRQSTAT | USBSS IRQ_STATUS |
| 2Ch | IRQENABLER | USBSS IRQ_ENABLE_SET |
| 30h | IRQCLEARR | USBSS IRQ_ENABLE_CLR |
| 100h | IRQDMATHOLDTX00 | USBSS IRQ_DMA_THRESHOLD_TX0_0 |
| 104h | IRQDMATHOLDTX01 | USBSS IRQ_DMA_THRESHOLD_TX0_1 |
| 108h | IRQDMATHOLDTX02 | USBSS IRQ_DMA_THRESHOLD_TX0_2 |
| 10Ch | IRQDMATHOLDTX03 | USBSS IRQ_DMA_THRESHOLD_TX0_3 |
| 110h | IRQDMATHOLDRX00 | USBSS IRQ_DMA_THRESHOLD_RX0_0 |
| 114h | IRQDMATHOLDRX01 | USBSS IRQ_DMA_THRESHOLD_RX0_1 |
| 118h | IRQDMATHOLDRX02 | USBSS IRQ_DMA_THRESHOLD_RX0_2 |
| 11Ch | IRQDMATHOLDRX03 | USBSS IRQ_DMA_THRESHOLD_RX0_3 |
| 120h | IRQDMATHOLDTX10 | USBSS IRQ_DMA_THRESHOLD_TX1_0 |
| 124h | IRQDMATHOLDTX11 | USBSS IRQ_DMA_THRESHOLD_TX1_1 |
| 128h | IRQDMATHOLDTX12 | USBSS IRQ_DMA_THRESHOLD_TX1_2 |
| 12Ch | IRQDMATHOLDTX13 | USBSS IRQ_DMA_THRESHOLD_TX1_3 |
| 130h | IRQDMATHOLDRX10 | USBSS IRQ_DMA_THRESHOLD_RX1_0 |
| 134h | IRQDMATHOLDRX11 | USBSS IRQ_DMA_THRESHOLD_RX1_1 |
| 138h | IRQDMATHOLDRX12 | USBSS IRQ_DMA_THRESHOLD_RX1_2 |
| 13Ch | IRQDMATHOLDRX13 | USBSS IRQ_DMA_THRESHOLD_RX1_3 |
| 140h | IRQDMAENABLE0 | USBSS IRQ_DMA_ENABLE_0 |
| 144h | IRQDMAENABLE1 | USBSS IRQ_DMA_ENABLE_1 |
| 200h | IRQFRAMETHOLDTX00 | USBSS IRQ_FRAME_THRESHOLD_TX0_0 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-30. USBSS Registers (continued)**

| Address Offset | Acronym | USBSS Register |
|---|---|---|
| 204h | IRQFRAMETHOLDTX01 | USBSS IRQ_FRAME_THRESHOLD_TX0_1 |
| 208h | IRQFRAMETHOLDTX02 | USBSS IRQ_FRAME_THRESHOLD_TX0_2 |
| 20Ch | IRQFRAMETHOLDTX03 | USBSS IRQ_FRAME_THRESHOLD_TX0_3 |
| 210h | IRQFRAMETHOLDRX00 | USBSS IRQ_FRAME_THRESHOLD_RX0_0 |
| 214h | IRQFRAMETHOLDRX01 | USBSS IRQ_FRAME_THRESHOLD_RX0_1 |
| 218h | IRQFRAMETHOLDRX02 | USBSS IRQ_FRAME_THRESHOLD_RX0_2 |
| 21Ch | IRQFRAMETHOLDRX03 | USBSS IRQ_FRAME_THRESHOLD_RX0_3 |
| 220h | IRQFRAMETHOLDTX10 | USBSS IRQ_FRAME_THRESHOLD_TX1_0 |
| 224h | IRQFRAMETHOLDTX11 | USBSS IRQ_FRAME_THRESHOLD_TX1_1 |
| 228h | IRQFRAMETHOLDTX12 | USBSS IRQ_FRAME_THRESHOLD_TX1_2 |
| 22Ch | IRQFRAMETHOLDTX13 | USBSS IRQ_FRAME_THRESHOLD_TX1_3 |
| 230h | IRQFRAMETHOLDRX10 | USBSS IRQ_FRAME_THRESHOLD_RX1_0 |
| 234h | IRQFRAMETHOLDRX11 | USBSS IRQ_FRAME_THRESHOLD_RX1_1 |
| 238h | IRQFRAMETHOLDRX12 | USBSS IRQ_FRAME_THRESHOLD_RX1_2 |
| 23Ch | IRQFRAMETHOLDRX13 | USBSS IRQ_FRAME_THRESHOLD_RX1_3 |
| 240h | IRQFRAMEENABLE0 | USBSS IRQ_FRAME_ENABLE_0 |
| 244h | IRQFRAMEENABLE1 | USBSS IRQ_FRAME_ENABLE_1 |

### 25.9.1.1 USBSS Revision Register (REVREG)

The revision register contains the major and minor revisions for the USB subsystem module. The USBSS revision register is shown in Figure 25-23 and described in Table 25-31.

**Figure 25-23. USBSS Revision Register (REVREG)**

| 31 | 30 | 29 | 28 | 27 | 16 |
|---|---|---|---|---|---|
| Scheme | | Reserved | | func | |
| R-1 | | R-0h | | R-EA2h | |

| 15 | 11 | 10 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| R_rtl | | X_major | | Custom | | Y_minor | |
| R-1 | | R-0h | | R-0h | | R-Bh | |

LEGEND: R = Read only; -*n* = value after reset

**Table 25-31. USBSS Revision Register (REVREG) Field Description**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-30 | Scheme | | Used to distinguish between old scheme and current. |
| | | 1 | Highlander 0.8 |
| 29-28 | Reserved | 0 | Reserved |
| 27-16 | func | EA2h | Function indicates a software compatible module family. |
| 15-11 | R_rtl | 1 | RTL revision. Will vary depending on release. |
| 10-8 | X_major | 0 | Major revision. |
| 7-6 | Custom | 0 | Custom revision |
| 5-0 | Y_minor | Bh | Minor revision. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.2 USBSS SYSCONFIG Register (SYSCONFIG)

The SYSCONFIG register is the clock management configuration register for the USBSS.

The clock enable bits 8 to 11 should only be configured/modified during the firmware initialization stage of the USBSS. In other words, these bits should not be modified while the system is running or after enabling the clock. These bits control the clock generation modules. Dynamically setting these bits can disable the power management logic. The USBSS SYSCONFIG register is shown in shown in Figure 25-24 and described in Table 25-32.

**Figure 25-24. USBSS SYSCONFIG Register (SYSCONFIG)**

| 31 | 18 | 17 | 16 |
|---|---|---|---|
| Reserved | | usb1 d2 ocp en_n | usb0 d2 ocp en_n |
| R-0h | | R/W-0h | R/W-0h |

| 15 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|
| Reserved | | usb0 ocp en_n | phy0 utmi en_n | usb1 ocp en_n | phy1 utmi en_n |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | Standby mode | | Idlemode | | Freeemu | Soft reset |
| R-0h | | R/W-2h | | R/W-2h | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-32. USBSS SYSCONFIG Register (SYSCONFIG) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-18 | Reserved | | Reserved |
| 17 | usb1 d2 ocp en_n | | Active low clock enable for usb1_d2_ocp_clk |
| | | 0 | Enable |
| | | 1 | Disable |
| 16 | usb0 d2 ocp en_n | | Active low clock enable for usb0_d2_ocp_clk |
| | | 0 | Enable |
| | | 1 | Disable |
| 15-12 | Reserved | | Reserved |
| 11 | usb0 ocp en_n | | Active low clock enable for usb0_ocp_clk |
| | | 0 | Enable |
| | | 1 | Disable |
| 10 | phy0 utmi en_n | | Active low clock enable for phy0_utmi_clk |
| | | 0 | Enable |
| | | 1 | Disable |
| 9 | usb1 ocp en_n | | Active low clock enable for usb1_ocp_clk |
| | | 0 | Enable |
| | | 1 | Disable |
| 8 | phy1 utmi en_n | | Active low clock enable for phy1_utmi_clk |
| | | 0 | Enable |
| | | 1 | Disable |
| 7-6 | Reserved | | Reserved |
| 5-4 | Standby mode | | Configuration of the local initiator state management mode. |
| | | 0 | Force-standby mode |
| | | 1 | No-standby mode |
| | | 2 | Smart-standby mode |
| | | 3 | Smart-standby wakeup capable mode |
| 3-2 | Idlemode | | Configuration of the local target state management mode. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-32. USBSS SYSCONFIG Register (SYSCONFIG) Field Descriptions (continued)**

| Bits | Field | Value | Description |
|---|---|---|---|
| | | 0 | Force-idle mode |
| | | 1 | No-idle mode |
| | | 2 | Smart-idle mode |
| | | 3 | Smart-idle wakeup capable mode |
| 1 | Freeemu | | Sensitivity to emulation (debug) suspend input signal. |
| | | 0 | Sensitive to emulation suspend |
| | | 1 | NOT sensitive to emulation suspend |


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-32. USBSS SYSCONFIG Register (SYSCONFIG) Field Descriptions (continued)**

| Bits | Field | Value | Description |
|---|---|---|---|
| 0 | Soft reset | | Software reset of USBSS, USB0, and USB1 modules |
| | | Write 0 | No action |
| | | Write 1 | Initiate software reset |
| | | Read 0 | Reset done, no action |
| | | Read 1 | Reset ongoing |

### 25.9.1.3 USBSS End of Interrupt Register (EOI)

The USBSS end of interrupt register (EOI) allows the CPU to acknowledge completion of an interrupt by writing 0, zero, to the EOI_VECTOR field. An eoi_write signal will be generated and another interrupt will be triggered if interrupt sources remain.

This register will be reset one cycle after it has been written to. The USBSS end of interrupt register is shown in Figure 25-25 and described in Table 25-33.

**Figure 25-25. USBSS End of Interrupt Register (EOI)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 1 | 0 |
|---|---|---|
| Reserved | | EOI_VECTOR |
| R-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-33. USBSS End of Interrupt Register (EOI) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-1 | Reserved | Always read as 0. Writes have no effect. |
| 0 | EOI_VECTOR | EOI for USBSS interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.4 USBSS IRQ_STATUS_RAW (IRQSTATRAW)

The USBSS IRQSTATRAW register allows the USBSS interrupt sources to be manually triggered when writing a 1 to a specific bit. A read from this register returns the USBSS interrupt event pending value. This register is shown in Figure 25-26 and described in Table 25-34.

General actions per bit:
Write 0: No action
Write 1: Set event
Read 0: No event pending
Read 1: Event pending

**Figure 25-26. USBSS IRQ_STATUS_RAW (IRQSTATRAW)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|
| Reserved | | rx_pkt_cmp_1 | tx_pkt_cmp_1 | rx_pkt_cmp_0 | tx_pkt_cmp_0 |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 7 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | pd_cmp_flag | rx_mop_ starvation | rx_sop_ starvation |
| R-0h | | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-34. USBSS IRQ_STATUS_RAW (IRQSTATRAW) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-12 | Reserved | Always read as 0. Writes have no effect. |
| 11 | rx_pkt_cmp_1 | Interrupt status for USB1 Rx CPPI DMA packet completion status |
| 10 | tx_pkt_cmp_1 | Interrupt status for USB1 Tx CPPI DMA packet completion status |
| 9 | rx_pkt_cmp_0 | Interrupt status for USB0 Rx CPPI DMA packet completion status |
| 8 | tx_pkt_cmp_0 | Interrupt status for USB0 Tx CPPI DMA packet completion status |
| 7-3 | Reserved | Always read as 0. Writes have no effect. |
| 2 | pd_cmp_flag | Interrupt status when the packet is completed and the Packet Descriptor is pushed into the queue manager |
| 1 | rx_mop_starvation | Interrupt status when queue manager cannot allocate an Rx buffer in the middle of a packet. |
| 0 | rx_sop_starvation | Interrupt status when queue manager cannot allocate an Rx buffer at the start of a packet. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.5 USBSS IRQ_STATUS Register (IRQSTAT)

The USBSS IRQSTAT register contains the interrupt sources status of the enabled interrupt. IRQSTAT register is a subset of IRQSTATRAW register. Writing a '1' to a bit field will clear a pending interrupt event. This register is shown in Figure 25-27 and described in Table 25-35.

General actions per bit:
Write 0: No action
Write 1: Clear event
Read 0: No event pending
Read 1: Event pending

**Figure 25-27. USBSS IRQ_STATUS (IRQSTAT)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | rx_pkt_cmp_1 | tx_pkt_cmp_1 | rx_pkt_cmp_0 | tx_pkt_cmp_0 |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | pd_cmp_flag | rx_mop_ starvation | rx_sop_ starvation |
| R-0h | | | | | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-35. USBSS IRQ_STATUS (IRQSTAT) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-12 | Reserved | Always read as 0. Writes have no effect. |
| 11 | rx_pkt_cmp_1 | Interrupt status for USB1 Rx CPPI DMA packet completion status |
| 10 | tx_pkt_cmp_1 | Interrupt status for USB1 Tx CPPI DMA packet completion status |
| 9 | rx_pkt_cmp_0 | Interrupt status for USB0 Rx CPPI DMA packet completion status |
| 8 | tx_pkt_cmp_0 | Interrupt status for USB0 Tx CPPI DMA packet completion status |
| 7-3 | Reserved | Always read as 0. Writes have no effect. |
| 2 | pd_cmp_flag | Interrupt status when the packet is completed and the Packet Descriptor is pushed into the queue manager. |
| 1 | rx_mop_starvation | Interrupt status when queue manager cannot allocate an Rx buffer in the middle of a packet. |
| 0 | rx_sop_starvation | Interrupt status when queue manager cannot allocate an Rx buffer at the start of a packet. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.6 USBSS IRQ_ENABLE_SET Register (IRQENABLER)

The USB interrupt mask set register (IRQENABLER) allows the USB interrupts to be enabled (unmasked). A read from this register returns the USB interrupts enabled.

The USBSS IRQ_ENABLE_SET register is shown in Figure 25-28 and described in Table 25-36.

General actions per bit:
Write 0: No action
Write 1: Interrupt enabled
Read 0: Interrupt disabled
Read 1: Interrupt enabled

**Figure 25-28. USBSS IRQ_ENABLE_SET Register (IRQENABLER)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | rx_pkt_cmp_1 | tx_pkt_cmp_1 | rx_pkt_cmp_0 | tx_pkt_cmp_0 |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | pd_cmp_flag | rx_mop_ starvation | rx_sop_ starvation |
| R-0h | | | | | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-36. USBSS IRQ_ENABLE_SET Register (IRQENABLER) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-12 | Reserved | Always read as 0. Writes have no effect. |
| 11 | rx_pkt_cmp_1 | Interrupt enable for USB1 Rx CPPI DMA packet completion status |
| 10 | tx_pkt_cmp_1 | Interrupt enable for USB1 Tx CPPI DMA packet completion status |
| 9 | rx_pkt_cmp_0 | Interrupt enable for USB0 Rx CPPI DMA packet completion status |
| 8 | tx_pkt_cmp_0 | Interrupt enable for USB0 Tx CPPI DMA packet completion status |
| 7-3 | Reserved | Always read as 0. Writes have no effect. |
| 2 | pd_cmp_flag | Interrupt enable when the packet is completed and the Packet Descriptor is pushed into the queue manager. |
| 1 | rx_mop_starvation | Interrupt enable when queue manager cannot allocate an Rx buffer in the middle of a packet. |
| 0 | rx_sop_starvation | Interrupt enable when queue manager cannot allocate an Rx buffer at the start of a packet. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.7 USBSS IRQ_ENABLE_CLR Register (IRQCLEARR)

The USBSS IRQCLEARR register allows the USBSS interrupt sources to be masked when writing a 1 to a specific bit. A read of this register returns the USBSS interrupt enabled value.

The USBSS IRQ_ENABLE_CLR register is shown in Figure 25-29 and described in Table 25-37.

General actions per bit:
Write 0: No action
Write 1: Disable interrupt
Read 0: Interrupt disabled
Read 1: Interrupt enabled

**Figure 25-29. USBSS IRQ_ENABLE_CLR Register (IRQCLEARR)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|
| Reserved | | rx_pkt_cmp_1 | tx_pkt_cmp_1 | rx_pkt_cmp_0 | tx_pkt_cmp_0 |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

| 7 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Reserved | | pd_cmp_flag | rx_mop_ starvation | rx_sop_ starvation |
| R-0h | | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-37. USBSS IRQ_ENABLE_CLR Register (IRQCLEARR) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-12 | Reserved | Always read as 0. Writes have no effect. |
| 11 | rx_pkt_cmp_1 | Interrupt enable for USB1 Rx CPPI DMA packet completion status |
| 10 | tx_pkt_cmp_1 | Interrupt enable for USB1 Tx CPPI DMA packet completion status |
| 9 | rx_pkt_cmp_0 | Interrupt enable for USB0 Rx CPPI DMA packet completion status |
| 8 | tx_pkt_cmp_0 | Interrupt enable for USB0 Tx CPPI DMA packet completion status |
| 7-3 | Reserved | Always read as 0. Writes have no effect. |
| 2 | pd_cmp_flag | Interrupt enable when the packet is completed and the Packet Descriptor is pushed into the queue manager. |
| 1 | rx_mop_starvation | Interrupt enable when queue manager cannot allocate an Rx buffer in the middle of a packet. |
| 0 | rx_sop_starvation | Interrupt enable when queue manager cannot allocate an Rx buffer at the start of a packet. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.8 USBSS IRQ_DMA_THRESHOLD_TX0_0 Register (IRQDMATHOLDTX00)

The USBSS IRQDMATHOLDTX00 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX0_0 register is shown in Figure 25-30 and described in Table 25-38.

**Figure 25-30. USBSS IRQ_DMA_THRESHOLD_TX0_0 Register (IRQDMATHOLDTX00)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_tx0_3 | dma_thres_tx0_2 | dma_thres_tx0_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-38. USBSS IRQ_DMA_THRESHOLD_TX0_0 Register (IRQDMATHOLDTX00) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_tx0_3 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 3 |
| 23-16 | dma_thres_tx0_2 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 2 |
| 15-8 | dma_thres_tx0_1 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 1 |
| 7-0 | Reserved | Always read as 0. Writes have no effect. |

### 25.9.1.9 USBSS IRQ_DMA_THRESHOLD_TX0_1 Register (IRQDMATHOLDTX01)

The USBSS IRQDMATHOLDTX01 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX0_1 register is shown in Figure 25-31 and described in Table 25-39.

**Figure 25-31. USBSS IRQ_DMA_THRESHOLD_TX0_1 Register (IRQDMATHOLDTX01)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_tx0_7 | dma_thres_tx0_6 | dma_thres_tx0_5 | dma_thres_tx0_4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-39. USBSS IRQ_DMA_THRESHOLD_TX0_1 Register (IRQDMATHOLDTX01) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_tx0_7 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 7 |
| 23-16 | dma_thres_tx0_6 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 6 |
| 15-8 | dma_thres_tx0_5 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 5 |
| 7-0 | dma_thres_tx0_4 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.10 USBSS IRQ_DMA_THRESHOLD_TX0_2 Register (IRQDMATHOLDTX02)

The USBSS IRQDMATHOLDTX02 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX0_2 register is shown in Figure 25-32 and described in Table 25-40.

**Figure 25-32. USBSS IRQ_DMA_THRESHOLD_TX0_2 Register (IRQDMATHOLDTX02)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_tx0_11 | dma_thres_tx0_10 | dma_thres_tx0_9 | dma_thres_tx0_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-40. USBSS IRQ_DMA_THRESHOLD_TX0_2 Register (IRQDMATHOLDTX02) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_tx0_11 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 11 |
| 23-16 | dma_thres_tx0_10 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 10 |
| 15-8 | dma_thres_tx0_9 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 9 |
| 7-0 | dma_thres_tx0_8 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 8 |

### 25.9.1.11 USBSS IRQ_DMA_THRESHOLD_TX0_3 Register (IRQDMATHOLDTX03)

The USBSS IRQDMATHOLDTX03 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX0_3 register is shown in Figure 25-32 and described in Table 25-40.

**Figure 25-33. USBSS IRQ_DMA_THRESHOLD_TX0_3 Register (IRQDMATHOLDTX03)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_tx0_15 | dma_thres_tx0_14 | dma_thres_tx0_13 | dma_thres_tx0_12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-41. USBSS IRQ_DMA_THRESHOLD_TX0_3 Register (IRQDMATHOLDTX03) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_tx0_15 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 15 |
| 23-16 | dma_thres_tx0_14 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 14 |
| 15-8 | dma_thres_tx0_13 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 13 |
| 7-0 | dma_thres_tx0_12 | DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.12 USBSS IRQ_DMA_THRESHOLD_RX0_0 Register (IRQDMATHOLDRX00)

The USBSS IRQDMATHOLDRX00 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX0_0 register is shown in Figure 25-34 and described in Table 25-42.

**Figure 25-34. USBSS IRQ_DMA_THRESHOLD_RX0_0 Register (IRQDMATHOLDRX00)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_rx0_3 | dma_thres_rx0_2 | dma_thres_rx0_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-42. USBSS IRQ_DMA_THRESHOLD_RX0_0 Register (IRQDMATHOLDRX00) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_rx0_3 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 3 |
| 23-16 | dma_thres_rx0_2 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 2 |
| 15-8 | dma_thres_rx0_1 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 1 |
| 7-0 | Reserved | Always read as 0. Writes have no effect. |

### 25.9.1.13 USBSS IRQ_DMA_THRESHOLD_RX0_1 Register (IRQDMATHOLDRX01)

The USBSS IRQDMATHOLDRX01 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX0_1 register is shown in Figure 25-35 and described in Table 25-43.

**Figure 25-35. USBSS IRQ_DMA_THRESHOLD_RX0_1 Register (IRQDMATHOLDRX01)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_rx0_7 | dma_thres_rx0_6 | dma_thres_rx0_5 | dma_thres_rx0_4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-43. USBSS IRQ_DMA_THRESHOLD_RX0_1 Register (IRQDMATHOLDRX00) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_rx0_7 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 7 |
| 23-16 | dma_thres_rx0_6 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 6 |
| 15-8 | dma_thres_rx0_5 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 5 |
| 7-0 | dma_thres_rx0_4 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.14 USBSS IRQ_DMA_THRESHOLD_RX0_2 Register (IRQDMATHOLDRX02)

The USBSS IRQ_DMA_THRESHOLD_RX0_2 register defines the size of the four DMA thresholds for interrupt pacing. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX0_2 register is shown in Figure 25-36 and described in Table 25-44.

**Figure 25-36. USBSS IRQ_DMA_THRESHOLD_RX0_2 Register (IRQDMATHOLDRX02)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_rx0_11 | dma_thres_rx0_10 | dma_thres_rx0_9 | dma_thres_rx0_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-44. USBSS IRQ_DMA_THRESHOLD_RX0_2 Register (IRQDMATHOLDRX02) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_rx0_11 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 11 |
| 23-16 | dma_thres_rx0_10 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 10 |
| 15-8 | dma_thres_rx0_9 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 9 |
| 7-0 | dma_thres_rx0_8 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 8 |

### 25.9.1.15 USBSS IRQ_DMA_THRESHOLD_RX0_3 Register (IRQDMATHOLDRX03)

The USBSS IRQDMATHOLDRX03 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX0_3 register is shown in Figure 25-37 and described in Table 25-45.

**Figure 25-37. USBSS IRQ_DMA_THRESHOLD_RX0_3 Register (IRQDMATHOLDRX03)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_rx0_15 | dma_thres_rx0_14 | dma_thres_rx0_13 | dma_thres_rx0_12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-45. USBSS IRQ_DMA_THRESHOLD_RX0_3 Register (IRQDMATHOLDRX03) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_rx0_15 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 15 |
| 23-16 | dma_thres_rx0_14 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 14 |
| 15-8 | dma_thres_rx0_13 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 13 |
| 7-0 | dma_thres_rx0_12 | DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.16 USBSS IRQ_DMA_THRESHOLD_TX1_0 Register (IRQDMATHOLDTX10)

The USBSS IRQDMATHOLDTX10 register (IRQDMATHOLDTX10) defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX1_0 register is shown in Figure 25-38 and described in Table 25-46.

**Figure 25-38. USBSS IRQ_DMA_THRESHOLD_TX1_0 Register (IRQDMATHOLDTX10)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_rx0_3 | dma_thres_rx0_2 | dma_thres_rx0_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-46. USBSS IRQ_DMA_THRESHOLD_TX1_0 Register (IRQDMATHOLDTX10) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_tx1_3 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 3 |
| 23-16 | dma_thres_tx1_2 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 2 |
| 15-8 | dma_thres_tx1_1 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 1 |
| 7-0 | Reserved | Always read as 0. Writes have no effect. |

### 25.9.1.17 USBSS IRQ_DMA_THRESHOLD_TX1_1 Register (IRQDMATHOLDTX11)

The USBSS IRQDMATHOLDTX11 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX1_1 register is shown in Figure 25-39 and described in Table 25-47.

**Figure 25-39. USBSS IRQ_DMA_THRESHOLD_TX1_1 Register (IRQDMATHOLDTX11)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_tx1_7 | dma_thres_tx1_6 | dma_thres_tx1_5 | dma_thres_tx1_4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-47. USBSS IRQ_DMA_THRESHOLD_TX1_1 Register (IRQDMATHOLDTX11) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_tx1_7 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 7 |
| 23-16 | dma_thres_tx1_6 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 6 |
| 15-8 | dma_thres_tx1_5 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 5 |
| 7-0 | dma_thres_tx1_4 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.18 USBSS IRQ_DMA_THRESHOLD_TX1_2 Register (IRQDMATHOLDTX12)

The USBSS IRQDMATHOLDTX12 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX1_2 register is shown in Figure 25-40 and described in Table 25-48.

**Figure 25-40. USBSS IRQ_DMA_THRESHOLD_TX1_2 Register (IRQDMATHOLDTX12)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_tx1_11 | dma_thres_tx1_10 | dma_thres_tx1_9 | dma_thres_tx1_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-48. USBSS IRQ_DMA_THRESHOLD_TX1_2 Register (IRQDMATHOLDTX12) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_tx1_11 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 11 |
| 23-16 | dma_thres_tx1_10 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 10 |
| 15-8 | dma_thres_tx1_9 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 9 |
| 7-0 | dma_thres_tx1_8 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 8 |

### 25.9.1.19 USBSS IRQ_DMA_THRESHOLD_TX1_3 Register (IRQDMATHOLDTX13)

The USBSS IRQDMATHOLDTX13 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX1_3 register is shown in Figure 25-41 and described in Table 25-49.

**Figure 25-41. USBSS IRQ_DMA_THRESHOLD_TX1_3 Register (IRQDMATHOLDTX13)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_tx1_15 | dma_thres_tx1_14 | dma_thres_tx1_13 | dma_thres_tx1_12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-49. USBSS IRQ_DMA_THRESHOLD_TX1_3 Register (IRQDMATHOLDTX13) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_tx1_15 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 15 |
| 23-16 | dma_thres_tx1_14 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 14 |
| 15-8 | dma_thres_tx1_13 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 13 |
| 7-0 | dma_thres_tx1_12 | DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.20 USBSS IRQ_DMA_THRESHOLD_RX1_0 Register (IRQDMATHOLDRX10)

The USBSS IRQDMATHOLDRX10 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX1_0 register is shown in Figure 25-42 and described in Table 25-50.

**Figure 25-42. USBSS IRQ_DMA_THRESHOLD_RX1_0 Register (IRQDMATHOLDRX10)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_rx1_3 | dma_thres_rx1_2 | dma_thres_rx1_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-50. USBSS IRQ_DMA_THRESHOLD_RX1_0 Register (IRQDMATHOLDRX10) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_rx1_3 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 3 |
| 23-16 | dma_thres_rx1_2 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 2 |
| 15-8 | dma_thres_rx1_1 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 1 |
| 7-0 | Reserved | Always read as 0. Writes have no effect. |

### 25.9.1.21 USBSS IRQ_DMA_THRESHOLD_RX1_1 Register (IRQDMATHOLDRX11)

The USBSS IRQDMATHOLDRX11 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX1_1 register is shown in Figure 25-43 and described in Table 25-51.

**Figure 25-43. USBSS IRQ_DMA_THRESHOLD_RX1_1 Register (IRQDMATHOLDRX11)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_rx1_7 | dma_thres_rx1_6 | dma_thres_rx1_5 | dma_thres_rx1_4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-51. USBSS IRQ_DMA_THRESHOLD_RX1_1 Register (IRQDMATHOLDRX11) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_rx1_7 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 7 |
| 23-16 | dma_thres_rx1_6 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 6 |
| 15-8 | dma_thres_rx1_5 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 5 |
| 7-0 | dma_thres_rx1_4 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.22 USBSS IRQ_DMA_THRESHOLD_RX1_2 Register (IRQDMATHOLDRX12)

The USBSS IRQDMATHOLDRX12 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX1_2 register is shown in Figure 25-44 and described in Table 25-52.

**Figure 25-44. USBSS IRQ_DMA_THRESHOLD_RX1_2 Register (IRQDMATHOLDRX12)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_rx1_11 | dma_thres_rx1_10 | dma_thres_rx1_9 | dma_thres_rx1_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-52. USBSS IRQ_DMA_THRESHOLD_RX1_2 Register (IRQDMATHOLDRX12) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_rx1_11 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 11 |
| 23-16 | dma_thres_rx1_10 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 10 |
| 15-8 | dma_thres_rx1_9 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 9 |
| 7-0 | dma_thres_rx1_8 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 8 |

### 25.9.1.23 USBSS IRQ_DMA_THRESHOLD_RX1_3 Register (IRQDMATHOLDRX13)

The USBSS IRQDMATHOLDRX13 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX1_3 register is shown in Figure 25-45 and described in Table 25-53.

**Figure 25-45. USBSS IRQ_DMA_THRESHOLD_RX1_3 Register (IRQDMATHOLDRX13)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| dma_thres_rx1_15 | dma_thres_rx1_14 | dma_thres_rx1_13 | dma_thres_rx1_12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-53. USBSS IRQ_DMA_THRESHOLD_RX1_3 Register (IRQDMATHOLDRX13) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | dma_thres_rx1_15 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 15 |
| 23-16 | dma_thres_rx1_14 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 14 |
| 15-8 | dma_thres_rx1_13 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 13 |
| 7-0 | dma_thres_rx1_12 | DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.24 USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0)

The USBSS IRQDMAENABLE0 contains 32 enables. Sixteen enables are assigned to the 16 endpoints associated with the tx_pkt_cmp_0 interrupt and 16 enables are assigned to the 16 endpoints associated with the rx_pkt_cmp_0 interrupt. Endpoint 0 for both the tx and rx does not exist; but it is included for completeness of this register.

If the enable is set; then that specific endpoint can generate an interrupt if the event pending count has exceeded the threshold.

If the enable is cleared; then that specific endpoint cannot generate an interrupt regardless if the event pending count has exceeded the threshold or not.

The USBSS IRQ_DMA_ENABLE_0 Register is shown in Figure 25-46 and described in Table 25-54.

**Figure 25-46. USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| dma_en_rx0_15 | dma_en_rx0_14 | dma_en_rx0_13 | dma_en_rx0_12 | dma_en_rx0_11 | dma_en_rx0_10 | dma_en_rx0_9 | dma_en_rx0_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| dma_en_rx0_7 | dma_en_rx0_6 | dma_en_rx0_5 | dma_en_rx0_4 | dma_en_rx0_3 | dma_en_rx0_2 | dma_en_rx0_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| dma_en_tx0_15 | dma_en_tx0_14 | dma_en_tx0_13 | dma_en_tx0_12 | dma_en_tx0_11 | dma_en_tx0_10 | dma_en_tx0_9 | dma_en_tx0_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| dma_en_tx0_7 | dma_en_tx0_6 | dma_en_tx0_5 | dma_en_tx0_4 | dma_en_tx0_3 | dma_en_tx0_2 | dma_en_tx0_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 25-54. USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | dma_en_rx0_15 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 15 |
| 30 | dma_en_rx0_14 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 14 |
| 29 | dma_en_rx0_13 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 13 |
| 28 | dma_en_rx0_12 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 12 |
| 27 | dma_en_rx0_11 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 11 |
| 26 | dma_en_rx0_10 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 10 |
| 25 | dma_en_rx0_9 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 9 |
| 24 | dma_en_rx0_8 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 8 |
| 23 | dma_en_rx0_7 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 7 |
| 22 | dma_en_rx0_6 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 6 |
| 21 | dma_en_rx0_5 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 5 |
| 20 | dma_en_rx0_4 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 4 |
| 19 | dma_en_rx0_3 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 3 |
| 18 | dma_en_rx0_2 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 2 |
| 17 | dma_en_rx0_1 | DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 1 |
| 16 | Reserved | Always read as 0. Writes have no effect. |
| 15 | dma_en_tx0_15 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 15 |
| 14 | dma_en_tx0_14 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 14 |
| 13 | dma_en_tx0_13 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 13 |
| 12 | dma_en_tx0_12 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 12 |
| 11 | dma_en_tx0_11 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 11 |
| 10 | dma_en_tx0_10 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-54. USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 9 | dma_en_tx0_9 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 9 |
| 8 | dma_en_tx0_8 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 8 |
| 7 | dma_en_tx0_7 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 7 |
| 6 | dma_en_tx0_6 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 6 |
| 5 | dma_en_tx0_5 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 5 |
| 4 | dma_en_tx0_4 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 4 |
| 3 | dma_en_tx0_3 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 3 |
| 2 | dma_en_tx0_2 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 2 |
| 1 | dma_en_tx0_1 | DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 1 |
| 0 | Reserved | Always read as 0. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.25 USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1)

The USBSS IRQDMAENABLE1 register contains 32 enables. Sixteen enables are assigned to the 16 endpoints associated with the tx_pkt_cmp_1 interrupt and 16 enables are assigned to the 16 endpoints associated with the rx_pkt_cmp_1 interrupt. Endpoint 0 for both the tx and rx does not exist; but it is included for completeness of this register.

If the enable is set; then that specific endpoint can generate an interrupt if the event pending count has exceeded the threshold.

If the enable is cleared; then that specific endpoint cannot generate an interrupt regardless if the event pending count has exceeded the threshold or not.

The USBSS IRQ_DMA_ENABLE_0 Register is shown in Figure 25-47 and described in Table 25-55.

**Figure 25-47. USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| dma_en_rx1_15 | dma_en_rx1_14 | dma_en_rx1_13 | dma_en_rx1_12 | dma_en_rx1_11 | dma_en_rx1_10 | dma_en_rx1_9 | dma_en_rx1_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| dma_en_rx1_7 | dma_en_rx1_6 | dma_en_rx1_5 | dma_en_rx1_4 | dma_en_rx1_3 | dma_en_rx1_2 | dma_en_rx1_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| dma_en_tx1_15 | dma_en_tx1_14 | dma_en_tx1_13 | dma_en_tx1_12 | dma_en_tx1_11 | dma_en_tx1_10 | dma_en_tx1_9 | dma_en_tx1_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| dma_en_tx1_7 | dma_en_tx1_6 | dma_en_tx1_5 | dma_en_tx1_4 | dma_en_tx1_3 | dma_en_tx1_2 | dma_en_tx1_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 25-55. USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | dma_en_rx1_15 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 15 |
| 30 | dma_en_rx1_14 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 14 |
| 29 | dma_en_rx1_13 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 13 |
| 28 | dma_en_rx1_12 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 12 |
| 27 | dma_en_rx1_11 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 11 |
| 26 | dma_en_rx1_10 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 10 |
| 25 | dma_en_rx1_9 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 9 |
| 24 | dma_en_rx1_8 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 8 |
| 23 | dma_en_rx1_7 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 7 |
| 22 | dma_en_rx1_6 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 6 |
| 21 | dma_en_rx1_5 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 5 |
| 20 | dma_en_rx1_4 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 4 |
| 19 | dma_en_rx1_3 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 3 |
| 18 | dma_en_rx1_2 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 2 |
| 17 | dma_en_rx1_1 | DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 1 |
| 16 | Reserved | Always read as 0. Writes have no effect. |
| 15 | dma_en_tx1_15 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 15 |
| 14 | dma_en_tx1_14 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 14 |
| 13 | dma_en_tx1_13 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 13 |
| 12 | dma_en_tx1_12 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 12 |
| 11 | dma_en_tx1_11 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 11 |
| 10 | dma_en_tx1_10 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-55. USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 9 | dma_en_tx1_9 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 9 |
| 8 | dma_en_tx1_8 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 8 |
| 7 | dma_en_tx1_7 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 7 |
| 6 | dma_en_tx1_6 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 6 |
| 5 | dma_en_tx1_5 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 5 |
| 4 | dma_en_tx1_4 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 4 |
| 3 | dma_en_tx1_3 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 3 |
| 2 | dma_en_tx1_2 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 2 |
| 1 | dma_en_tx1_1 | DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 1 |
| 0 | Reserved | Always read as 0. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.26 USBSS IRQ_FRAME_THRESHOLD_TX0_0 Register (IRQFRAMETHOLDTX00)

The USBSS IRQFRAMETHOLDTX00 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX0_0 register is shown in Figure 25-48 and described in Table 25-56.

**Figure 25-48. USBSS IRQ_FRAME_THRESHOLD_TX0_0 Register (IRQFRAMETHOLDTX00)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_tx0_3 | frame_thres_tx0_2 | frame_thres_tx0_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-56. USBSS IRQ_FRAME_THRESHOLD_TX0_0 Register (IRQFRAMETHOLDTX00) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_tx0_3 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 3 |
| 23-16 | frame_thres_tx0_2 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 2 |
| 15-8 | frame_thres_tx0_1 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 1 |
| 7-0 | Reserved | Always read as 0. Writes have no effect. |

### 25.9.1.27 USBSS IRQ_FRAME_THRESHOLD_TX0_1 Register (IRQFRAMETHOLDTX01)

The USBSS IRQFRAMETHOLDTX01 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX0_1 register is shown in Figure 25-49 and described in Table 25-57.

**Figure 25-49. USBSS IRQ_FRAME_THRESHOLD_TX0_1 Register (IRQFRAMETHOLDTX01)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_tx0_7 | frame_thres_tx0_6 | frame_thres_tx0_5 | frame_thres_tx0_4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-57. USBSS IRQ_FRAME_THRESHOLD_TX0_1 Register (IRQFRAMETHOLDTX01) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_tx0_7 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 7 |
| 23-16 | frame_thres_tx0_6 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 6 |
| 15-8 | frame_thres_tx0_5 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 5 |
| 7-0 | frame_thres_tx0_4 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.28 USBSS IRQ_FRAME_THRESHOLD_TX0_2 Register (IRQFRAMETHOLDTX02)

The USBSS IRQFRAMETHOLDTX02 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX0_2 register is shown in Figure 25-50 and described in Table 25-58.

**Figure 25-50. USBSS IRQ_FRAME_THRESHOLD_TX0_2 Register (IRQFRAMETHOLDTX02)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_tx0_11 | frame_thres_tx0_10 | frame_thres_tx0_9 | frame_thres_tx0_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-58. USBSS IRQ_FRAME_THRESHOLD_TX0_2 Register (IRQFRAMETHOLDTX02) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_tx0_11 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 11 |
| 23-16 | frame_thres_tx0_10 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 10 |
| 15-8 | frame_thres_tx0_9 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 9 |
| 7-0 | frame_thres_tx0_8 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 8 |

### 25.9.1.29 USBSS IRQ_FRAME_THRESHOLD_TX0_3 Register (IRQFRAMETHOLDTX03)

The USBSS IRQFRAMETHOLDTX03 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX0_3 register is shown in Figure 25-51 and described in Table 25-59.

**Figure 25-51. USBSS IRQ_FRAME_THRESHOLD_TX0_3 Register (IRQFRAMETHOLDTX03)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_tx0_15 | frame_thres_tx0_14 | frame_thres_tx0_13 | frame_thres_tx0_12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-59. USBSS IRQ_FRAME_THRESHOLD_TX0_3 Register (IRQFRAMETHOLDTX03) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_tx0_15 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 15 |
| 23-16 | frame_thres_tx0_14 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 14 |
| 15-8 | frame_thres_tx0_13 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 13 |
| 7-0 | frame_thres_tx0_12 | FRAME threshold value for tx_pkt_cmp_0 for USB0 Endpoint 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.30 USBSS IRQ_FRAME_THRESHOLD_RX0_0 Register (IRQFRAMETHOLDRX00)

The USBSS IRQFRAMETHOLDRX00 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX0_0 register is shown in Figure 25-52 and described in Table 25-60.

**Figure 25-52. USBSS IRQ_FRAME_THRESHOLD_RX0_0 Register (IRQFRAMETHOLDRX00)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_rx0_3 | frame_thres_rx0_2 | frame_thres_rx0_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-60. USBSS IRQ_FRAME_THRESHOLD_RX0_0 Register (IRQFRAMETHOLDRX00) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_rx0_3 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 3 |
| 23-16 | frame_thres_rx0_2 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 2 |
| 15-8 | frame_thres_rx0_1 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 1 |
| 7-0 | Reserved | Always read as 0. Writes have no effect. |

### 25.9.1.31 USBSS IRQ_FRAME_THRESHOLD_RX0_1 Register (IRQFRAMETHOLDRX01)

The USBSS IRQFRAMETHOLDRX01 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX0_1 register is shown in Figure 25-53 and described in Table 25-61.

**Figure 25-53. USBSS IRQ_FRAME_THRESHOLD_RX0_1 Register (IRQFRAMETHOLDRX01)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_rx0_7 | frame_thres_rx0_6 | frame_thres_rx0_5 | frame_thres_rx0_4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-61. USBSS IRQ_FRAME_THRESHOLD_RX0_1 Register (IRQFRAMETHOLDRX01) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_rx0_7 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 7 |
| 23-16 | frame_thres_rx0_6 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 6 |
| 15-8 | frame_thres_rx0_5 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 5 |
| 7-0 | frame_thres_rx0_4 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.32 USBSS IRQ_FRAME_THRESHOLD_RX0_2 Register (IRQFRAMETHOLDRX02)

The USBSS IRQFRAMETHOLDRX02 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX0_2 register is shown in Figure 25-53 and described in Table 25-62.

**Figure 25-54. USBSS IRQ_FRAME_THRESHOLD_RX0_2 Register (IRQFRAMETHOLDRX02)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_rx0_11 | frame_thres_rx0_10 | frame_thres_rx0_9 | frame_thres_rx0_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-62. USBSS IRQ_FRAME_THRESHOLD_RX0_2 Register (IRQFRAMETHOLDRX02) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_rx0_11 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 11 |
| 23-16 | frame_thres_rx0_10 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 10 |
| 15-8 | frame_thres_rx0_9 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 9 |
| 7-0 | frame_thres_rx0_8 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 8 |

### 25.9.1.33 USBSS IRQ_FRAME_THRESHOLD_RX0_3 Register (IRQFRAMETHOLDRX03)

The USBSS IRQFRAMETHOLDRX03 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX0_3 register is shown in Figure 25-55 and described in Table 25-63.

**Figure 25-55. USBSS IRQ_FRAME_THRESHOLD_RX0_3 Register (IRQFRAMETHOLDRX03)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_rx0_15 | frame_thres_rx0_14 | frame_thres_rx0_13 | frame_thres_rx0_12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-63. USBSS IRQ_FRAME_THRESHOLD_RX0_3 Register (IRQFRAMETHOLDRX03) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_rx0_15 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 15 |
| 23-16 | frame_thres_rx0_14 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 14 |
| 15-8 | frame_thres_rx0_13 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 13 |
| 7-0 | frame_thres_rx0_12 | FRAME threshold value for rx_pkt_cmp_0 for USB0 Endpoint 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.34 USBSS IRQ_FRAME_THRESHOLD_TX1_0 Register (IRQFRAMETHOLDTX10)

The USBSS IRQFRAMETHOLDTX10 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX1_0 register is shown in Figure 25-56 and described in Table 25-64.

**Figure 25-56. USBSS IRQ_FRAME_THRESHOLD_TX1_0 Register (IRQFRAMETHOLDTX10)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_tx1_3 | frame_thres_tx1_2 | frame_thres_tx1_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-64. USBSS IRQ_FRAME_THRESHOLD_TX1_0 Register (IRQFRAMETHOLDTX10) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_tx1_3 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 3 |
| 23-16 | frame_thres_tx1_2 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 2 |
| 15-8 | frame_thres_tx1_1 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 1 |
| 7-0 | Reserved | Always read as 0. Writes have no effect. |

### 25.9.1.35 USBSS IRQ_FRAME_THRESHOLD_TX1_1 Register (IRQFRAMETHOLDTX11)

The USBSS IRQFRAMETHOLDTX11 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX1_1 register is shown in Figure 25-57 and described in Table 25-65.

**Figure 25-57. USBSS IRQ_FRAME_THRESHOLD_TX1_1 Register (IRQFRAMETHOLDTX11)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_tx1_7 | frame_thres_tx1_6 | frame_thres_tx1_5 | frame_thres_tx1_4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-65. USBSS IRQ_FRAME_THRESHOLD_TX1_1 Register (IRQFRAMETHOLDTX11) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_tx1_7 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 7 |
| 23-16 | frame_thres_tx1_6 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 6 |
| 15-8 | frame_thres_tx1_5 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 5 |
| 7-0 | frame_thres_tx1_4 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.36 USBSS IRQ_FRAME_THRESHOLD_TX1_2 Register (IRQFRAMETHOLDTX12)

The USBSS IRQFRAMETHOLDTX12 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX1_2 register is shown in Figure 25-58 and described in Table 25-66.

**Figure 25-58. USBSS IRQ_FRAME_THRESHOLD_TX1_2 Register (IRQFRAMETHOLDTX12)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_tx1_11 | frame_thres_tx1_10 | frame_thres_tx1_9 | frame_thres_tx1_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-66. USBSS IRQ_FRAME_THRESHOLD_TX1_2 Register (IRQFRAMETHOLDTX12) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_tx1_11 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 11 |
| 23-16 | frame_thres_tx1_10 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 10 |
| 15-8 | frame_thres_tx1_9 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 9 |
| 7-0 | frame_thres_tx1_8 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 8 |

### 25.9.1.37 USBSS IRQ_FRAME_THRESHOLD_TX1_3 Register (IRQFRAMETHOLDTX13)

The USBSS IRQFRAMETHOLDTX13 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX1_3 register is shown in Figure 25-59 and described in Table 25-67.

**Figure 25-59. USBSS IRQ_FRAME_THRESHOLD_TX1_3 Register (IRQFRAMETHOLDTX13)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_tx1_15 | frame_thres_tx1_14 | frame_thres_tx1_13 | frame_thres_tx1_12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-67. USBSS IRQ_FRAME_THRESHOLD_TX1_3 Register (IRQFRAMETHOLDTX13) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_tx1_15 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 15 |
| 23-16 | frame_thres_tx1_14 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 14 |
| 15-8 | frame_thres_tx1_13 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 13 |
| 7-0 | frame_thres_tx1_12 | FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.38 USBSS IRQ_FRAME_THRESHOLD_RX1_0 Register (IRQFRAMETHOLDRX10)

The USBSS IRQFRAMETHOLDRX10 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX1_0 register is shown in Figure 25-60 and described in Table 25-68.

**Figure 25-60. USBSS IRQ_FRAME_THRESHOLD_RX1_0 Register (IRQFRAMETHOLDRX10)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_rx1_3 | frame_thres_rx1_2 | frame_thres_rx1_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-68. USBSS IRQ_FRAME_THRESHOLD_RX1_0 Register (IRQFRAMETHOLDRX10) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_rx1_3 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 3 |
| 23-16 | frame_thres_rx1_2 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 2 |
| 15-8 | frame_thres_rx1_1 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 1 |
| 7-0 | Reserved | Always read as 0. Writes have no effect. |

### 25.9.1.39 USBSS IRQ_FRAME_THRESHOLD_RX1_1 Register (IRQFRAMETHOLDRX11)

The USBSS IRQFRAMETHOLDRX11 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX1_1 register is shown in Figure 25-61 and described in Table 25-69.

**Figure 25-61. USBSS IRQ_FRAME_THRESHOLD_RX1_1 Register (IRQFRAMETHOLDRX11)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_rx1_7 | frame_thres_rx1_6 | frame_thres_rx1_5 | frame_thres_rx1_4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-69. USBSS IRQ_FRAME_THRESHOLD_RX1_1 Register (IRQFRAMETHOLDRX11) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_rx1_7 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 7 |
| 23-16 | frame_thres_rx1_6 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 6 |
| 15-8 | frame_thres_rx1_5 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 5 |
| 7-0 | frame_thres_rx1_4 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 4 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.40 USBSS IRQ_FRAME_THRESHOLD_RX1_2 Register (IRQFRAMETHOLDRX12)

The USBSS IRQFRAMETHOLDRX12 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX1_2 register is shown in Figure 25-62 and described in Table 25-70.

**Figure 25-62. USBSS IRQ_FRAME_THRESHOLD_RX1_2 Register (IRQFRAMETHOLDRX12)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_rx1_11 | frame_thres_rx1_10 | frame_thres_rx1_9 | frame_thres_rx1_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-70. USBSS IRQ_FRAME_THRESHOLD_RX1_2 Register (IRQFRAMETHOLDRX12) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_rx1_11 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 11 |
| 23-16 | frame_thres_rx1_10 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 10 |
| 15-8 | frame_thres_rx1_9 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 9 |
| 7-0 | frame_thres_rx1_8 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 8 |

### 25.9.1.41 USBSS IRQ_FRAME_THRESHOLD_RX1_3 Register (IRQFRAMETHOLDRX13)

The USBSS IRQFRAMETHOLDRX13 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX1_3 register is shown in Figure 25-63 and described in Table 25-71.

**Figure 25-63. USBSS IRQ_FRAME_THRESHOLD_RX1_3 Register (IRQFRAMETHOLDRX13)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| frame_thres_rx1_15 | frame_thres_rx1_14 | frame_thres_rx1_13 | frame_thres_rx1_12 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-71. USBSS IRQ_FRAME_THRESHOLD_RX1_3 Register (IRQFRAMETHOLDRX13) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | frame_thres_rx1_15 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 15 |
| 23-16 | frame_thres_rx1_14 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 14 |
| 15-8 | frame_thres_rx1_13 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 13 |
| 7-0 | frame_thres_rx1_12 | FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 12 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.42 USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0)

The USBSS IRQFRAMEENABLE0 register contains 32 enables for USB0. Sixteen enables are assigned to the 16 endpoints associated with the tx_pkt_cmp_0 interrupt and 16 enables are assigned to the 16 endpoints associated with the rx_pkt_cmp_0 interrupt. Endpoint 0 for both the tx and rx does not exist; but it is included for completeness of this register.

If the enable is set; then that specific endpoint can generate an interrupt if the frame count has exceeded the threshold. If the enable is not set; then no hardware interrupt will be generated for that specific endpoint. It is still possible to generate the interrupt via software.

If the enable is cleared; then that specific endpoint cannot generate an interrupt regardless if the frame count has exceeded the threshold or not.

The USBSS IRQ_FRAME_ENABLE_0 register is shown in Figure 25-64 and described in Table 25-72.

**Figure 25-64. USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| frame_en_rx0_15 | frame_en_rx0_14 | frame_en_rx0_13 | frame_en_rx0_12 | frame_en_rx0_11 | frame_en_rx0_10 | frame_en_rx0_9 | frame_en_rx0_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| frame_en_rx0_7 | frame_en_rx0_6 | frame_en_rx0_5 | frame_en_rx0_4 | frame_en_rx0_3 | frame_en_rx0_2 | frame_en_rx0_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| frame_en_tx0_15 | frame_en_tx0_14 | frame_en_tx0_13 | frame_en_tx0_12 | frame_en_tx0_11 | frame_en_tx0_10 | frame_en_tx0_9 | frame_en_tx0_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| frame_en_tx0_7 | frame_en_tx0_6 | frame_en_tx0_5 | frame_en_tx0_4 | frame_en_tx0_3 | frame_en_tx0_2 | frame_en_tx0_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 25-72. USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | frame_en_rx0_15 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 15 |
| 30 | frame_en_rx0_14 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 14 |
| 29 | frame_en_rx0_13 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 13 |
| 28 | frame_en_rx0_12 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 12 |
| 27 | frame_en_rx0_11 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 11 |
| 26 | frame_en_rx0_10 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 10 |
| 25 | frame_en_rx0_9 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 9 |
| 24 | frame_en_rx0_8 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 8 |
| 23 | frame_en_rx0_7 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 7 |
| 22 | frame_en_rx0_6 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 6 |
| 21 | frame_en_rx0_5 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 5 |
| 20 | frame_en_rx0_4 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 4 |
| 19 | frame_en_rx0_3 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 3 |
| 18 | frame_en_rx0_2 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 2 |
| 17 | frame_en_rx0_1 | FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 1 |
| 16 | Reserved | Always read as 0. Writes have no effect. |
| 15 | frame_en_tx0_15 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 15 |
| 14 | frame_en_tx0_14 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 14 |
| 13 | frame_en_tx0_13 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 13 |
| 12 | frame_en_tx0_12 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 12 |
| 11 | frame_en_tx0_11 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 11 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-72. USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 10 | frame_en_tx0_10 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 10 |
| 9 | frame_en_tx0_9 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 9 |
| 8 | frame_en_tx0_8 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 8 |
| 7 | frame_en_tx0_7 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 7 |
| 6 | frame_en_tx0_6 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 6 |
| 5 | frame_en_tx0_5 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 5 |
| 4 | frame_en_tx0_4 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 4 |
| 3 | frame_en_tx0_3 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 3 |
| 2 | frame_en_tx0_2 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 2 |
| 1 | frame_en_tx0_1 | FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 1 |
| 0 | Reserved | Always read as 0. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.1.43 USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1)

The USBSS IRQFRAMEENABLE1 register contains 30 enables for USB1. Fifteen enables are assigned to the 15 endpoints associated with the tx_pkt_cmp_1 interrupt and 15 enables are assigned to the 15 endpoints associated with the rx_pkt_cmp_1 interrupt. Endpoint 0 for both the tx and rx does not exist; but it is included for completeness of this register.

If the enable is set; then that specific endpoint can generate an interrupt if the frame count has exceeded the threshold. If the enable is not set; then no hardware interrupt will be generated for that specific endpoint. It is still possible to generate the interrupt via software.

If the enable is cleared; then that specific endpoint cannot generate an interrupt regardless if the frame count has exceeded the threshold or not. The USBSS IRQ_FRAME_ENABLE_1 register is shown in Figure 25-65 and described in Table 25-73.

**Figure 25-65. USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| frame_en_rx1_15 | frame_en_rx1_14 | frame_en_rx1_13 | frame_en_rx1_12 | frame_en_rx1_11 | frame_en_rx1_10 | frame_en_rx1_9 | frame_en_rx1_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| frame_en_rx1_7 | frame_en_rx1_6 | frame_en_rx1_5 | frame_en_rx1_4 | frame_en_rx1_3 | frame_en_rx1_2 | frame_en_rx1_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| frame_en_tx1_15 | frame_en_tx1_14 | frame_en_tx1_13 | frame_en_tx1_12 | frame_en_tx1_11 | frame_en_tx1_10 | frame_en_tx1_9 | frame_en_tx1_8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| frame_en_tx1_7 | frame_en_tx1_6 | frame_en_tx1_5 | frame_en_tx1_4 | frame_en_tx1_3 | frame_en_tx1_2 | frame_en_tx1_1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 25-73. USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | frame_en_rx1_15 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 15 |
| 30 | frame_en_rx1_14 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 14 |
| 29 | frame_en_rx1_13 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 13 |
| 28 | frame_en_rx1_12 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 12 |
| 27 | frame_en_rx1_11 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 11 |
| 26 | frame_en_rx1_10 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 10 |
| 25 | frame_en_rx1_9 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 9 |
| 24 | frame_en_rx1_8 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 8 |
| 23 | frame_en_rx1_7 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 7 |
| 22 | frame_en_rx1_6 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 6 |
| 21 | frame_en_rx1_5 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 5 |
| 20 | frame_en_rx1_4 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 4 |
| 19 | frame_en_rx1_3 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 3 |
| 18 | frame_en_rx1_2 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 2 |
| 17 | frame_en_rx1_1 | FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 1 |
| 16 | Reserved | Always read as 0. Writes have no effect. |
| 15 | frame_en_tx1_15 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 15 |
| 14 | frame_en_tx1_14 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 14 |
| 13 | frame_en_tx1_13 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 13 |
| 12 | frame_en_tx1_12 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 12 |
| 11 | frame_en_tx1_11 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 11 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-73. USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 10 | frame_en_tx1_10 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 10 |
| 9 | frame_en_tx1_9 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 9 |
| 8 | frame_en_tx1_8 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 8 |
| 7 | frame_en_tx1_7 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 7 |
| 6 | frame_en_tx1_6 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 6 |
| 5 | frame_en_tx1_5 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 5 |
| 4 | frame_en_tx1_4 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 4 |
| 3 | frame_en_tx1_3 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 3 |
| 2 | frame_en_tx1_2 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 2 |
| 1 | frame_en_tx1_1 | FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 1 |
| 0 | Reserved | Always read as 0. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

## 25.9.2 USB0 and USB1 Controller Registers

The USB0 and USB1 register descriptions are included in this section.

### 25.9.2.1 USB0 Controller Registers

Table 25-74 lists the registers for the USB0 Controller submodule.

**Table 25-74. USB0 Controller Registers**

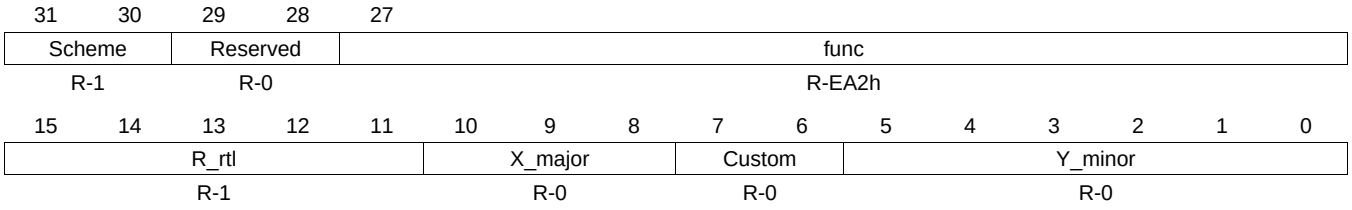
| Address Offset | Acronym | USB0 Controller Register |
|---|---|---|
| 1000h | USB0REV | USB0 REVISION |
| 1014h | USB0CTRL | USB0 Control Register |
| 1018h | USB0STAT | USB0 Status Register |
| 1020h | USB0IRQMSTAT | USB0 IRQ_MERGED_STATUS |
| 1024h | USB0IRQEOI | USB0 IRQ_EOI |
| 1028h | USB0IRQSTATRAW0 | USB0 IRQ_STATUS_RAW_0 |
| 102Ch | USB0IRQSTATRAW1 | USB0 IRQ_STATUS_RAW_1 |
| 1030h | USB0IRQSTAT0 | USB0 IRQ_STATUS_0 |
| 1034h | USB0IRQSTAT1 | USB0 IRQ_STATUS_1 |
| 1038h | USB0IRQENABLESET0 | USB0 IRQ_ENABLE_SET_0 |
| 103Ch | USB0IRQENABLESET1 | USB0 IRQ_ENABLE_SET_1 |
| 1040h | USB0IRQENABLECLR0 | USB0 IRQ_ENABLE_CLR_0 |
| 1044h | USB0IRQENABLECLR1 | USB0 IRQ_ENABLE_CLR_1 |
| 1070h | USB0TXMODE | USB0 Tx Mode Register |
| 1074h | USB0RXMODE | USB0 Rx Mode Register |
| 1080h | USB0GENRNDISEP1 | USB0 Generic RNDIS Size EP1 |
| 1084h | USB0GENRNDISEP2 | USB0 Generic RNDIS Size EP2 |
| 1088h | USB0GENRNDISEP3 | USB0 Generic RNDIS Size EP3 |
| 108Ch | USB0GENRNDISEP4 | USB0 Generic RNDIS Size EP4 |
| 1090h | USB0GENRNDISEP5 | USB0 Generic RNDIS Size EP5 |
| 1094h | USB0GENRNDISEP6 | USB0 Generic RNDIS Size EP6 |
| 1098h | USB0GENRNDISEP7 | USB0 Generic RNDIS Size EP7 |
| 109Ch | USB0GENRNDISEP8 | USB0 Generic RNDIS Size EP8 |
| 10A0h | USB0GENRNDISEP9 | USB0 Generic RNDIS Size EP9 |
| 10A4h | USB0GENRNDISEP10 | USB0 Generic RNDIS Size EP10 |
| 10A8h | USB0GENRNDISEP11 | USB0 Generic RNDIS Size EP11 |
| 10ACh | USB0GENRNDISEP12 | USB0 Generic RNDIS Size EP12 |
| 10B0h | USB0GENRNDISEP13 | USB0 Generic RNDIS Size EP13 |
| 10B4h | USB0GENRNDISEP14 | USB0 Generic RNDIS Size EP14 |
| 10B8h | USB0GENRNDISEP15 | USB0 Generic RNDIS Size EP15 |
| 10D0h | USB0AUTOREQ | USB0 Auto Req Register |
| 10D4h | USB0SRPFIXTIME | USB0 SRP Fix Time |
| 10D8h | USB0TDOWN | USB0 Teardown Register |
| 10DCh | | USB0 Threshold XDMA Idle Register |
| 10E0h | USB0UTMI | USB0 PHY UTMI Register |
| 10E4h | USB0UTMILB | USB0 MGC UTMI Loopback Register |
| 10E8h | USB0MODE | USB0 Mode Register |
| 1400h–1468Ch | - | USB0 Mentor Core Registers/FIFOs |
| 146Ch | USB0HWVERS | USB0 Mentor Core HWVERS Register (see Section 25.9.7.2.6) |
| 0x1470 – 0x159C | - | USB0 Mentor Core Registers/FIFOs |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.1 USB0 Revision Register (USB0REV)

The USB0 revision register (USB0REV) register contains the major and minor revisions for the USB0 module. The USB0 revision register is shown in Figure 25-66 and described in Table 25-75.

**Figure 25-66. USB0 Revision Register (USB0REV)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-75. USB0 Revision Register (USB0REV) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-30 | Scheme | | Used to distinguish between old scheme and current |
| | | 1 | Highlander 0.8 |
| 29-28 | Reserved | 0 | Reserved |
| 27-16 | func | EA2h | Function indicates a software compatible module family. |
| 15-11 | R_rtl | 1h | RTL revision. Will vary depending on release. |
| 10-8 | X_major | 0h | Major revision |
| 7-6 | Custom | 0h | Custom revision |
| 5-0 | Y_minor | 0h | Minor revision |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.2 USB0 Control Register (USB0CTRL)

The USB0 control register (USB0CTRL) allows the CPU to control various aspects of the module. If uint is set high, then the Mentor controller generic interrupt for USB[9] will be generated (if enabled). This requires software to read the Mentor controller's registers to determine which interrupt USB[0] to USB[7] occurred. If uint is set low, then the usb20otg_f module will automatically read the Mentor controller's registers and set the appropriate interrupt USB[0] to USB[7] (if enabled). The generic interrupt for USB[9] will not be generated.

The USB0 control register is shown in Figure 25-67 and described in Table 25-76.

**Figure 25-67. USB0 Control Register (USB0CTRL)**

| 31 | 30 | 29 | 16 |
|---|---|---|---|
| dis_deb | dis_srp | Reserved | |
| R/W-0h | R/W-0h | R-0h | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | soft reset isolation | rndis | uint | Reserved | clkfack | soft reset |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W1-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-76. USB0 Control Register (USB0CTRL) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31 | dis_deb | | Disable the VBUS debouncer circuit fix |
| 30 | dis_srp | | Disable the OTG Session Request Protocol (SRP) AVALID circuit fix. When enabled (=0) this allows additional time for the VBUS signal to be measured against the VBUS thresholds. The time is specified in the USB0 SRP Fix Time Register. |
| 29-6 | Reserved | | Always read as 0. Writes have no effect. |
| 5 | soft reset isolation | | Soft reset isolation. When high this bit forces all USB0 signals that connect to the USBSS to zeros during a soft reset via bit 0 of this register. This bit should be set high prior to setting bit 0 and cleared after bit 0 is cleared. |
| 4 | rndis | | Global RNDIS mode enable for all endpoints |
| 3 | uint | | USB non-Highlander interrupt enable |
| | | 1 | Non-Highlander |
| | | 0 | Highlander |
| 2 | Reserved | | Always read as 0. Writes have no effect. |
| 1 | clkfack | | Clock stop fast ack enable |
| 0 | soft reset | | Software reset of USB0 |
| | | Write 0 | No action |
| | | Write 1 | Initiate software reset |
| | | Read 0 | Reset done, no action |
| | | Read 1 | Reset ongoing |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.3 USB0 Status Register (USB0STAT)

The USB0 status register (USB0STAT) allows the CPU to check the voltage state level of USB0_DRVVBUS pin. The USB0 status register is shown in Figure 25-68 and described in Table 25-77.

**Figure 25-68. USB0 Status Register (USB0STAT)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | drvvbus |
| R-0h | | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-77. USB0 Status Register (USB0STAT) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-1 | Reserved | Always read as 0. Writes have no effect. |
| 0 | drvvbus | Current USB0_DRVVBUS value |

#### 25.9.2.1.4 USB0 IRQ_MERGED_STATUS Register (USB0IRQMSTAT)

The USB0 IRQ_MERGED_STATUS register (USB0IRQMSTAT) register contains a merged status for all the events for USB0. This register optimizes software accesses in a module where a large number of events are associated to one IRQ line. The merged status is read-only, and does not need to be actively cleared like the IRQ status. It will clear automatically.

The USB0 IRQ_MERGED_STATUS register is shown in Figure 25-69 and described in Table 25-78.

**Figure 25-69. USB0 IRQ_MERGED_STATUS Register (USB0IRQMSTAT)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | Bank1 | Bank1 |
| R-0h | | R-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-78. USB0 IRQ_MERGED_STATUS Register (USB0IRQMSTAT) Field Descriptions**

| Bits | Field | Unit | Description |
|---|---|---|---|
| 31-2 | Reserved | | Always read as 0. Writes have no effect. |
| 1 | Bank1 | 0 | No events pending from IRQ_STATUS_1 |
| | | 1 | At least one event is pending from IRQ_STATUS_1 |
| 0 | Bank0 | 0 | No events pending from IRQ_STATUS_0 |
| | | 1 | At least one event is pending from IRQ_STATUS_0 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.5 USB0 IRQ_EOI Register (USB0IRQEOI)

The USB0 IRQ_EOI register (USB0IRQEOI) allows the CPU to acknowledge completion of an interrupt by writing to the EOI. An eoi_write signal will be generated and another interrupt will be triggered if interrupt sources remain.

This register will be reset one cycle after it has been written to.

The USB0 IRQ_EOI register is shown in Figure 25-70 and described in Table 25-79.

**Figure 25-70. USB0 IRQ_EOI Register (USB0IRQEOI)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | EOI for USB0 |
| R-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-79. USB0 IRQ_EOI Register (USB0IRQEOI) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-1 | Reserved | Always read as 0. Writes have no effect. |
| 0 | EOI for USB0 | EOI for USB0 interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.6 USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0)

The USB0 IRQ_STATUS_RAW_0 register (USB0IRQSTATRAW0) allows the USB0 interrupt sources to be manually triggered when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Set event
Read 0: No event pending
Read 1: Event pending

The USB0 IRQ_STATUS_RAW_0 register is shown in Figure 25-71 and described in Table 25-80.

**Figure 25-71. USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RX EP 15 | RX EP 14 | RX EP 13 | RX EP 12 | RX EP 11 | RX EP 10 | RX EP 9 | RX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RX EP 7 | RX EP 6 | RX EP 5 | RX EP 4 | RX EP 3 | RX EP 2 | RX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TX EP 15 | TX EP 14 | TX EP 13 | TX EP 12 | TX EP 11 | TX EP 10 | TX EP 9 | TX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX EP 7 | TX EP 6 | TX EP 5 | TX EP 4 | TX EP 3 | TX EP 2 | TX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-80. USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | RX EP 15 | Interrupt status for RX endpoint 15 |
| 30 | RX EP 14 | Interrupt status for RX endpoint 14 |
| 29 | RX EP 13 | Interrupt status for RX endpoint 13 |
| 28 | RX EP 12 | Interrupt status for RX endpoint 12 |
| 27 | RX EP 11 | Interrupt status for RX endpoint 11 |
| 26 | RX EP 10 | Interrupt status for RX endpoint 10 |
| 25 | RX EP 9 | Interrupt status for RX endpoint 9 |
| 24 | RX EP 8 | Interrupt status for RX endpoint 8 |
| 23 | RX EP 7 | Interrupt status for RX endpoint 7 |
| 22 | RX EP 6 | Interrupt status for RX endpoint 6 |
| 21 | RX EP 5 | Interrupt status for RX endpoint 5 |
| 20 | RX EP 4 | Interrupt status for RX endpoint 4 |
| 19 | RX EP 3 | Interrupt status for RX endpoint 3 |
| 18 | RX EP 2 | Interrupt status for RX endpoint 2 |
| 17 | RX EP 1 | Interrupt status for RX endpoint 1 |
| 16 | Reserved | Always read 0. Writes have no effect. |
| 15 | TX EP 15 | Interrupt status for TX endpoint 15 |
| 14 | TX EP 14 | Interrupt status for TX endpoint 14 |
| 13 | TX EP 13 | Interrupt status for TX endpoint 13 |
| 12 | TX EP 12 | Interrupt status for TX endpoint 12 |
| 11 | TX EP 11 | Interrupt status for TX endpoint 11 |
| 10 | TX EP 10 | Interrupt status for TX endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-80. USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 9 | TX EP 9 | Interrupt status for TX endpoint 9 |
| 8 | TX EP 8 | Interrupt status for TX endpoint 8 |
| 7 | TX EP 7 | Interrupt status for TX endpoint 7 |
| 6 | TX EP 6 | Interrupt status for TX endpoint 6 |
| 5 | TX EP 5 | Interrupt status for TX endpoint 5 |
| 4 | TX EP 4 | Interrupt status for TX endpoint 4 |
| 3 | TX EP 3 | Interrupt status for TX endpoint 3 |
| 2 | TX EP 2 | Interrupt status for TX endpoint 2 |
| 1 | TX EP 1 | Interrupt status for TX endpoint 1 |
| 0 | TX EP 0 | Interrupt status for TX endpoint 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.7 USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1)

The USB0 IRQ_STATUS_RAW_1 register (USB0IRQSTATRAW1) allows the USB0 interrupt sources to be manually triggered when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Set event
Read 0: No event pending
Read 1: Event pending

The USB0 IRQ_STATUS_RAW_1 register is shown in Figure 25-72 and described in Table 25-81.

**Figure 25-72. USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TX FIFO 15 | TX FIFO 14 | TX FIFO 13 | TX FIFO 12 | TX FIFO 11 | TX FIFO 10 | TX FIFO 9 | TX FIFO 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TX FIFO 7 | TX FIFO 6 | TX FIFO 5 | TX FIFO 4 | TX FIFO 3 | TX FIFO 2 | TX FIFO 1 | TX FIFO 0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | | | | | 10 | 9 | 8 |
| Reserved | | | | | | USB[9] | USB[8] |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USB[7] | USB[6] | USB[5] | USB[4] | USB[3] | USB[2] | USB[1] | USB[0] |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-81. USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | TX FIFO 15 | Interrupt status for TX FIFO endpoint 15 |
| 30 | TX FIFO 14 | Interrupt status for TX FIFO endpoint 14 |
| 29 | TX FIFO 13 | Interrupt status for TX FIFO endpoint 13 |
| 28 | TX FIFO 12 | Interrupt status for TX FIFO endpoint 12 |
| 27 | TX FIFO 11 | Interrupt status for TX FIFO endpoint 11 |
| 26 | TX FIFO 10 | Interrupt status for TX FIFO endpoint 10 |
| 25 | TX FIFO 9 | Interrupt status for TX FIFO endpoint 9 |
| 24 | TX FIFO 8 | Interrupt status for TX FIFO endpoint 8 |
| 23 | TX FIFO 7 | Interrupt status for TX FIFO endpoint 7 |
| 22 | TX FIFO 6 | Interrupt status for TX FIFO endpoint 6 |
| 21 | TX FIFO 5 | Interrupt status for TX FIFO endpoint 5 |
| 20 | TX FIFO 4 | Interrupt status for TX FIFO endpoint 4 |
| 19 | TX FIFO 3 | Interrupt status for TX FIFO endpoint 3 |
| 18 | TX FIFO 2 | Interrupt status for TX FIFO endpoint 2 |
| 17 | TX FIFO 1 | Interrupt status for TX FIFO endpoint 1 |
| 16 | TX FIFO 0 | Interrupt status for TX FIFO endpoint 0 |
| 15-10 | Reserved | Always read 0. Writes have no effect. |
| 9 | USB[9] | Interrupt status for Mentor controller USB_INT generic interrupt |
| 8 | USB[8] | Interrupt status for DRVVBUS level change |
| 7 | USB[7] | Interrupt status for VBUS < VBUS valid threshold |
| 6 | USB[6] | Interrupt status for SRP detected |
| 5 | USB[5] | Interrupt status for device disconnected (host mode) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-81. USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 4 | USB[4] | Interrupt status for device connected (host mode) |
| 3 | USB[3] | Interrupt status for SOF started |
| 2 | USB[2] | Interrupt status for Reset signaling detected (peripheral mode)<br>Babble detected (host mode) |
| 1 | USB[1] | Interrupt status for Resume signaling detected |
| 0 | USB[0] | Interrupt status for Suspend signaling detected |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.8 USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0)

The USB0 IRQ_STATUS_0 register (USB0IRQSTAT0) allows the USB0 interrupt sources to be manually cleared when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Clear event
Read 0: No event pending
Read 1: Event pending

The USB0 IRQ_STATUS_0 register is shown in Figure 25-73 and described in Table 25-82.

**Figure 25-73. USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RX EP 15 | RX EP 14 | RX EP 13 | RX EP 12 | RX EP 11 | RX EP 10 | RX EP 9 | RX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RX EP 7 | RX EP 6 | RX EP 5 | RX EP 4 | RX EP 3 | RX EP 2 | RX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TX EP 15 | TX EP 14 | TX EP 13 | TX EP 12 | TX EP 11 | TX EP 10 | TX EP 9 | TX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX EP 7 | TX EP 6 | TX EP 5 | TX EP 4 | TX EP 3 | TX EP 2 | TX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-82. USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | RX EP 15 | Interrupt status for RX endpoint 15 |
| 30 | RX EP 14 | Interrupt status for RX endpoint 14 |
| 29 | RX EP 13 | Interrupt status for RX endpoint 13 |
| 28 | RX EP 12 | Interrupt status for RX endpoint 12 |
| 27 | RX EP 11 | Interrupt status for RX endpoint 11 |
| 26 | RX EP 10 | Interrupt status for RX endpoint 10 |
| 25 | RX EP 9 | Interrupt status for RX endpoint 9 |
| 24 | RX EP 8 | Interrupt status for RX endpoint 8 |
| 23 | RX EP 7 | Interrupt status for RX endpoint 7 |
| 22 | RX EP 6 | Interrupt status for RX endpoint 6 |
| 21 | RX EP 5 | Interrupt status for RX endpoint 5 |
| 20 | RX EP 4 | Interrupt status for RX endpoint 4 |
| 19 | RX EP 3 | Interrupt status for RX endpoint 3 |
| 18 | RX EP 2 | Interrupt status for RX endpoint 2 |
| 17 | RX EP 1 | Interrupt status for RX endpoint 1 |
| 16 | Reserved | Always read 0. Writes have no effect. |
| 15 | TX EP 15 | Interrupt status for TX endpoint 15 |
| 14 | TX EP 14 | Interrupt status for TX endpoint 14 |
| 13 | TX EP 13 | Interrupt status for TX endpoint 13 |
| 12 | TX EP 12 | Interrupt status for TX endpoint 12 |
| 11 | TX EP 11 | Interrupt status for TX endpoint 11 |
| 10 | TX EP 10 | Interrupt status for TX endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-82. USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 9 | TX EP 9 | Interrupt status for TX endpoint 9 |
| 8 | TX EP 8 | Interrupt status for TX endpoint 8 |
| 7 | TX EP 7 | Interrupt status for TX endpoint 7 |
| 6 | TX EP 6 | Interrupt status for TX endpoint 6 |
| 5 | TX EP 5 | Interrupt status for TX endpoint 5 |
| 4 | TX EP 4 | Interrupt status for TX endpoint 4 |
| 3 | TX EP 3 | Interrupt status for TX endpoint 3 |
| 2 | TX EP 2 | Interrupt status for TX endpoint 2 |
| 1 | TX EP 1 | Interrupt status for TX endpoint 1 |
| 0 | TX EP 0 | Interrupt status for TX endpoint 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.9 USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1)

The USB0 IRQ_STATUS_1 register (USB0IRQSTAT1) allows the USB0 interrupt sources to be manually cleared when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Clear event
Read 0: No event pending
Read 1: Event pending

The USB0 IRQ_STATUS_1 register is shown in Figure 25-74 and described in Table 25-83.

**Figure 25-74. USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TX FIFO 15 | TX FIFO 14 | TX FIFO 13 | TX FIFO 12 | TX FIFO 11 | TX FIFO 10 | TX FIFO 9 | TX FIFO 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TX FIFO 7 | TX FIFO 6 | TX FIFO 5 | TX FIFO 4 | TX FIFO 3 | TX FIFO 2 | TX FIFO 1 | TX FIFO 0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | | | | | 10 | 9 | 8 |
| Reserved | | | | | | USB[9] | USB[8] |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USB[7] | USB[6] | USB[5] | USB[4] | USB[3] | USB[2] | USB[1] | USB[0] |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-83. USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | TX FIFO 15 | Interrupt status for TX FIFO endpoint 15 |
| 30 | TX FIFO 14 | Interrupt status for TX FIFO endpoint 14 |
| 29 | TX FIFO 13 | Interrupt status for TX FIFO endpoint 13 |
| 28 | TX FIFO 12 | Interrupt status for TX FIFO endpoint 12 |
| 27 | TX FIFO 11 | Interrupt status for TX FIFO endpoint 11 |
| 26 | TX FIFO 10 | Interrupt status for TX FIFO endpoint 10 |
| 25 | TX FIFO 9 | Interrupt status for TX FIFO endpoint 9 |
| 24 | TX FIFO 8 | Interrupt status for TX FIFO endpoint 8 |
| 23 | TX FIFO 7 | Interrupt status for TX FIFO endpoint 7 |
| 22 | TX FIFO 6 | Interrupt status for TX FIFO endpoint 6 |
| 21 | TX FIFO 5 | Interrupt status for TX FIFO endpoint 5 |
| 20 | TX FIFO 4 | Interrupt status for TX FIFO endpoint 4 |
| 19 | TX FIFO 3 | Interrupt status for TX FIFO endpoint 3 |
| 18 | TX FIFO 2 | Interrupt status for TX FIFO endpoint 2 |
| 17 | TX FIFO 1 | Interrupt status for TX FIFO endpoint 1 |
| 16 | TX FIFO 0 | Interrupt status for TX FIFO endpoint 0 |
| 15-10 | Reserved | Always read 0. Writes have no effect. |
| 9 | USB[9] | Interrupt status for Mentor controller USB_INT generic interrupt |
| 8 | USB[8] | Interrupt status for DRVVBUS level change |
| 7 | USB[7] | Interrupt status for VBUS < VBUS valid threshold |
| 6 | USB[6] | Interrupt status for SRP detected |
| 5 | USB[5] | Interrupt status for device disconnected (host mode) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-83. USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 4 | USB[4] | Interrupt status for device connected (host mode) |
| 3 | USB[3] | Interrupt status for SOF started |
| 2 | USB[2] | Interrupt status for Reset signaling detected (peripheral mode)<br>Babble detected (host mode) |
| 1 | USB[1] | Interrupt status for Resume signaling detected |
| 0 | USB[0] | Interrupt status for Suspend signaling detected |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.10 USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0)

The USB0 IRQ_ENABLE_SET_0 register (USB0IRQENABLESET0) allows the USB0 interrupt sources to be manually enabled when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt enabled value. General actions per bit:
Write 0: No action
Write 1: Interrupt enabled
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB0 IRQ_ENABLE_SET_0 register is shown in Figure 25-75 and described in Table 25-84.

**Figure 25-75. USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RX EP 15 | RX EP 14 | RX EP 13 | RX EP 12 | RX EP 11 | RX EP 10 | RX EP 9 | RX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RX EP 7 | RX EP 6 | RX EP 5 | RX EP 4 | RX EP 3 | RX EP 2 | RX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TX EP 15 | TX EP 14 | TX EP 13 | TX EP 12 | TX EP 11 | TX EP 10 | TX EP 9 | TX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX EP 7 | TX EP 6 | TX EP 5 | TX EP 4 | TX EP 3 | TX EP 2 | TX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-84. USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | RX EP 15 | Interrupt enable for RX endpoint 15 |
| 30 | RX EP 14 | Interrupt enable for RX endpoint 14 |
| 29 | RX EP 13 | Interrupt enable for RX endpoint 13 |
| 28 | RX EP 12 | Interrupt enable for RX endpoint 12 |
| 27 | RX EP 11 | Interrupt enable for RX endpoint 11 |
| 26 | RX EP 10 | Interrupt enable for RX endpoint 10 |
| 25 | RX EP 9 | Interrupt enable for RX endpoint 9 |
| 24 | RX EP 8 | Interrupt enable for RX endpoint 8 |
| 23 | RX EP 7 | Interrupt enable for RX endpoint 7 |
| 22 | RX EP 6 | Interrupt enable for RX endpoint 6 |
| 21 | RX EP 5 | Interrupt enable for RX endpoint 5 |
| 20 | RX EP 4 | Interrupt enable for RX endpoint 4 |
| 19 | RX EP 3 | Interrupt enable for RX endpoint 3 |
| 18 | RX EP 2 | Interrupt enable for RX endpoint 2 |
| 17 | RX EP 1 | Interrupt enable for RX endpoint 1 |
| 16 | Reserved | Always read 0. Writes have no effect. |
| 15 | TX EP 15 | Interrupt enable for TX endpoint 15 |
| 14 | TX EP 14 | Interrupt enable for TX endpoint 14 |
| 13 | TX EP 13 | Interrupt enable for TX endpoint 13 |
| 12 | TX EP 12 | Interrupt enable for TX endpoint 12 |
| 11 | TX EP 11 | Interrupt enable for TX endpoint 11 |
| 10 | TX EP 10 | Interrupt enable for TX endpoint 10 |
| 9 | TX EP 9 | Interrupt enable for TX endpoint 9 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-84. USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 8 | TX EP 8 | Interrupt enable for TX endpoint 8 |
| 7 | TX EP 7 | Interrupt enable for TX endpoint 7 |
| 6 | TX EP 6 | Interrupt enable for TX endpoint 6 |
| 5 | TX EP 5 | Interrupt enable for TX endpoint 5 |
| 4 | TX EP 4 | Interrupt enable for TX endpoint 4 |
| 3 | TX EP 3 | Interrupt enable for TX endpoint 3 |
| 2 | TX EP 2 | Interrupt enable for TX endpoint 2 |
| 1 | TX EP 1 | Interrupt enable for TX endpoint 1 |
| 0 | TX EP 0 | Interrupt enable for TX endpoint 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.11 USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1)

The USB0 IRQ_ENABLE_SET_1 register (USB0IRQENABLESET1) allows the USB0 interrupt sources to be manually enabled when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt enabled value. General actions per bit:
Write 0: No action
Write 1: Interrupt enabled
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB0 IRQ_ENABLE_SET_1 register is shown in Figure 25-76 and described in Table 25-85.

**Figure 25-76. USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TX FIFO 15 | TX FIFO 14 | TX FIFO 13 | TX FIFO 12 | TX FIFO 11 | TX FIFO 10 | TX FIFO 9 | TX FIFO 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TX FIFO 7 | TX FIFO 6 | TX FIFO 5 | TX FIFO 4 | TX FIFO 3 | TX FIFO 2 | TX FIFO 1 | TX FIFO 0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | | | | | 10 | 9 | 8 |
| Reserved | | | | | | USB[9] | USB[8] |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USB[7] | USB[6] | USB[5] | USB[4] | USB[3] | USB[2] | USB[1] | USB[0] |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-85. USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | TX FIFO 15 | Interrupt enable for TX FIFO endpoint 15 |
| 30 | TX FIFO 14 | Interrupt enable for TX FIFO endpoint 14 |
| 29 | TX FIFO 13 | Interrupt enable for TX FIFO endpoint 13 |
| 28 | TX FIFO 12 | Interrupt enable for TX FIFO endpoint 12 |
| 27 | TX FIFO 11 | Interrupt enable for TX FIFO endpoint 11 |
| 26 | TX FIFO 10 | Interrupt enable for TX FIFO endpoint 10 |
| 25 | TX FIFO 9 | Interrupt enable for TX FIFO endpoint 9 |
| 24 | TX FIFO 8 | Interrupt enable for TX FIFO endpoint 8 |
| 23 | TX FIFO 7 | Interrupt enable for TX FIFO endpoint 7 |
| 22 | TX FIFO 6 | Interrupt enable for TX FIFO endpoint 6 |
| 21 | TX FIFO 5 | Interrupt enable for TX FIFO endpoint 5 |
| 20 | TX FIFO 4 | Interrupt enable for TX FIFO endpoint 4 |
| 19 | TX FIFO 3 | Interrupt enable for TX FIFO endpoint 3 |
| 18 | TX FIFO 2 | Interrupt enable for TX FIFO endpoint 2 |
| 17 | TX FIFO 1 | Interrupt enable for TX FIFO endpoint 1 |
| 16 | TX FIFO 0 | Interrupt enable for TX FIFO endpoint 0 |
| 15-10 | Reserved | Always read 0. Writes have no effect. |
| 9 | USB[9] | Interrupt enable for Mentor controller USB_INT generic interrupt |
| 8 | USB[8] | Interrupt enable for DRVVBUS level change |
| 7 | USB[7] | Interrupt enable for VBUS < VBUS valid threshold |
| 6 | USB[6] | Interrupt enable for SRP detected |
| 5 | USB[5] | Interrupt enable for device disconnected (host mode) |
| 4 | USB[4] | Interrupt enable for device connected (host mode) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-85. USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 3 | USB[3] | Interrupt enable for SOF started |
| 2 | USB[2] | Interrupt enable for Reset signaling detected (peripheral mode)<br>Babble detected (host mode) |
| 1 | USB[1] | Interrupt enable for Resume signaling detected |
| 0 | USB[0] | Interrupt enable for Suspend signaling detected |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.12 USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0)

The USB0 IRQ_ENABLE_CLR_0 register (USB0IRQENABLECLR0) allows the USB0 interrupt sources to be manually disabled when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt enabled value. General actions per bit:
Write 0: No action
Write 1: Clear interrupt

The USB0 IRQ_ENABLE_CLR_0 register is shown in Figure 25-77 and described in Table 25-86.

**Figure 25-77. USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RX EP 15 | RX EP 14 | RX EP 13 | RX EP 12 | RX EP 11 | RX EP 10 | RX EP 9 | RX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RX EP 7 | RX EP 6 | RX EP 5 | RX EP 4 | RX EP 3 | RX EP 2 | RX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TX EP 15 | TX EP 14 | TX EP 13 | TX EP 12 | TX EP 11 | TX EP 10 | TX EP 9 | TX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX EP 7 | TX EP 6 | TX EP 5 | TX EP 4 | TX EP 3 | TX EP 2 | TX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-86. USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | RX EP 15 | Clear interrupt for RX endpoint 15 |
| 30 | RX EP 14 | Clear interrupt for RX endpoint 14 |
| 29 | RX EP 13 | Clear interrupt for RX endpoint 13 |
| 28 | RX EP 12 | Clear interrupt for RX endpoint 12 |
| 27 | RX EP 11 | Clear interrupt for RX endpoint 11 |
| 26 | RX EP 10 | Clear interrupt for RX endpoint 10 |
| 25 | RX EP 9 | Clear interrupt for RX endpoint 9 |
| 24 | RX EP 8 | Clear interrupt for RX endpoint 8 |
| 23 | RX EP 7 | Clear interrupt for RX endpoint 7 |
| 22 | RX EP 6 | Clear interrupt for RX endpoint 6 |
| 21 | RX EP 5 | Clear interrupt for RX endpoint 5 |
| 20 | RX EP 4 | Clear interrupt for RX endpoint 4 |
| 19 | RX EP 3 | Clear interrupt for RX endpoint 3 |
| 18 | RX EP 2 | Clear interrupt for RX endpoint 2 |
| 17 | RX EP 1 | Clear interrupt for RX endpoint 1 |
| 16 | Reserved | Always read 0. Writes have no effect. |
| 15 | TX EP 15 | Clear interrupt for TX endpoint 15 |
| 14 | TX EP 14 | Clear interrupt for TX endpoint 14 |
| 13 | TX EP 13 | Clear interrupt for TX endpoint 13 |
| 12 | TX EP 12 | Clear interrupt for TX endpoint 12 |
| 11 | TX EP 11 | Clear interrupt for TX endpoint 11 |
| 10 | TX EP 10 | Clear interrupt for TX endpoint 10 |
| 9 | TX EP 9 | Clear interrupt for TX endpoint 9 |
| 8 | TX EP 8 | Clear interrupt for TX endpoint 8 |
| 7 | TX EP 7 | Clear interrupt for TX endpoint 7 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-86. USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 6 | TX EP 6 | Clear interrupt for TX endpoint 6 |
| 5 | TX EP 5 | Clear interrupt for TX endpoint 5 |
| 4 | TX EP 4 | Clear interrupt for TX endpoint 4 |
| 3 | TX EP 3 | Clear interrupt for TX endpoint 3 |
| 2 | TX EP 2 | Clear interrupt for TX endpoint 2 |
| 1 | TX EP 1 | Clear interrupt for TX endpoint 1 |
| 0 | Reserved | Always read 0. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.13 USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1)

The USB0 IRQ_ENABLE_CLR_1 register (USB0IRQENABLECLR1) allows the USB0 interrupt sources to be manually disabled when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt enabled value. General actions per bit:
Write 0: No action
Write 1: Clear interrupt

The USB0 IRQ_ENABLE_CLR_1 register is shown in Figure 25-78 and described in Table 25-87.

**Figure 25-78. USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TX FIFO 15 | TX FIFO 14 | TX FIFO 13 | TX FIFO 12 | TX FIFO 11 | TX FIFO 10 | TX FIFO 9 | TX FIFO 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TX FIFO 7 | TX FIFO 6 | TX FIFO 5 | TX FIFO 4 | TX FIFO 3 | TX FIFO 2 | TX FIFO 1 | TX FIFO 0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | | | | | 10 | 9 | 8 |
| Reserved | | | | | | USB[9] | USB[8] |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USB[7] | USB[6] | USB[5] | USB[4] | USB[3] | USB[2] | USB[1] | USB[0] |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-87. USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | TX FIFO 15 | Clear interrupt for TX FIFO endpoint 15 |
| 30 | TX FIFO 14 | Clear interrupt for TX FIFO endpoint 14 |
| 29 | TX FIFO 13 | Clear interrupt for TX FIFO endpoint 13 |
| 28 | TX FIFO 12 | Clear interrupt for TX FIFO endpoint 12 |
| 27 | TX FIFO 11 | Clear interrupt for TX FIFO endpoint 11 |
| 26 | TX FIFO 10 | Clear interrupt for TX FIFO endpoint 10 |
| 25 | TX FIFO 9 | Clear interrupt for TX FIFO endpoint 9 |
| 24 | TX FIFO 8 | Clear interrupt for TX FIFO endpoint 8 |
| 23 | TX FIFO 7 | Clear interrupt for TX FIFO endpoint 7 |
| 22 | TX FIFO 6 | Clear interrupt for TX FIFO endpoint 6 |
| 21 | TX FIFO 5 | Clear interrupt for TX FIFO endpoint 5 |
| 20 | TX FIFO 4 | Clear interrupt for TX FIFO endpoint 4 |
| 19 | TX FIFO 3 | Clear interrupt for TX FIFO endpoint 3 |
| 18 | TX FIFO 2 | Clear interrupt for TX FIFO endpoint 2 |
| 17 | TX FIFO 1 | Clear interrupt for TX FIFO endpoint 1 |
| 16 | TX FIFO 0 | Clear interrupt for TX FIFO endpoint 0 |
| 15-10 | Reserved | Always read 0. Writes have no effect. |
| 9 | USB[9] | Clear interrupt for Mentor controller USB_INT generic interrupt |
| 8 | USB[8] | Clear interrupt for DRVVBUS level change |
| 7 | USB[7] | Clear interrupt for VBUS < VBUS valid threshold |
| 6 | USB[6] | Clear interrupt for SRP detected |
| 5 | USB[5] | Clear interrupt for device disconnected (host mode) |
| 4 | USB[4] | Clear interrupt for device connected (host mode) |
| 3 | USB[3] | Clear interrupt for SOF started |
| 2 | USB[2] | Clear interrupt for Reset signaling detected (peripheral mode) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-87. USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| | | Babble detected (host mode) |
| 1 | USB[1] | Clear interrupt for Resume signaling detected |
| 0 | USB[0] | Clear interrupt for Suspend signaling detected |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.14 USB0 Tx Mode Register (USB0TXMODE)

The USB0 Tx mode register (USB0TXMODE) register allows the CPU to individually enable RNDIS/Generic/CDC modes for each endpoint. Using the global RNDIS enable in the Control Register overrides this register and enables RNDIS mode for all endpoints.

The USB0 Tx mode register is shown in Figure 25-79 and described in Table 25-88.

**Figure 25-79. USB0 Tx Mode Register (USB0TXMODE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | Tx15_mode | | Tx14_mode | | Tx13_mode | | Tx12_mode | | Tx11_mode | | Tx10_mode | | Tx9_mode | |
| R-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Tx8_mode | | Tx7_mode | | Tx6_mode | | Tx5_mode | | Tx4_mode | | Tx3_mode | | Tx2_mode | | Tx1_mode | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-88. USB0 Tx Mode Register (USB0TXMODE) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Always read as 0. Writes have no effect. |
| 29-28 | Tx15_mode | | |
| | | 00 | Transparent Mode on TX endpoint 15 |
| | | 01 | RNDIS MODE on TX endpoint 15 |
| | | 10 | CDC Mode on TX endpoint 15 |
| | | 11 | Generic RNDIS Mode on TX endpoint 15 |
| 27-26 | Tx14_mode | | |
| | | 00 | Transparent Mode on TX endpoint 14 |
| | | 01 | RNDIS MODE on TX endpoint 14 |
| | | 10 | CDC Mode on TX endpoint 14 |
| | | 11 | Generic RNDIS Mode on TX endpoint 14 |
| 25-24 | Tx13_mode | | |
| | | 00 | Transparent Mode on TX endpoint 13 |
| | | 01 | RNDIS MODE on TX endpoint 13 |
| | | 10 | CDC Mode on TX endpoint 13 |
| | | 11 | Generic RNDIS Mode on TX endpoint 13 |
| 23-22 | Tx12_mode | | |
| | | 00 | Transparent Mode on TX endpoint 12 |
| | | 01 | RNDIS MODE on TX endpoint 12 |
| | | 10 | CDC Mode on TX endpoint 12 |
| | | 11 | Generic RNDIS Mode on TX endpoint 12 |
| 21-20 | Tx11_mode | | |
| | | 00 | Transparent Mode on TX endpoint 11 |
| | | 01 | RNDIS MODE on TX endpoint 11 |
| | | 10 | CDC Mode on TX endpoint 11 |
| | | 11 | Generic RNDIS Mode on TX endpoint 11 |
| 19-18 | Tx10_mode | | |
| | | 00 | Transparent Mode on TX endpoint 10 |
| | | 01 | RNDIS MODE on TX endpoint 10 |
| | | 10 | CDC Mode on TX endpoint 10 |
| | | 11 | Generic RNDIS Mode on TX endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-88. USB0 Tx Mode Register (USB0TXMODE) Field Descriptions (continued)**

| Bits | Field | Value | Description |
|---|---|---|---|
| 17-16 | Tx9_mode | | |
| | | 00 | Transparent Mode on TX endpoint 9 |
| | | 01 | RNDIS MODE on TX endpoint 9 |
| | | 10 | CDC Mode on TX endpoint 9 |
| | | 11 | Generic RNDIS Mode on TX endpoint 9 |
| 15-14 | Tx8_mode | | |
| | | 00 | Transparent Mode on TX endpoint 8 |
| | | 01 | RNDIS MODE on TX endpoint 8 |
| | | 10 | CDC Mode on TX endpoint 8 |
| | | 11 | Generic RNDIS Mode on TX endpoint 8 |
| 13-12 | Tx7_mode | | |
| | | 00 | Transparent Mode on TX endpoint 7 |
| | | 01 | RNDIS MODE on TX endpoint 7 |
| | | 10 | CDC Mode on TX endpoint 7 |
| | | 11 | Generic RNDIS Mode on TX endpoint 7 |
| 11-10 | Tx6_mode | | |
| | | 00 | Transparent Mode on TX endpoint 6 |
| | | 01 | RNDIS MODE on TX endpoint 6 |
| | | 10 | CDC Mode on TX endpoint 6 |
| | | 11 | Generic RNDIS Mode on TX endpoint 6 |
| 9-8 | Tx5_mode | | |
| | | 00 | Transparent Mode on TX endpoint 5 |
| | | 01 | RNDIS MODE on TX endpoint 5 |
| | | 10 | CDC Mode on TX endpoint 5 |
| | | 11 | Generic RNDIS Mode on TX endpoint 5 |
| 7-6 | Tx4_mode | | |
| | | 00 | Transparent Mode on TX endpoint 4 |
| | | 01 | RNDIS MODE on TX endpoint 4 |
| | | 10 | CDC Mode on TX endpoint 4 |
| | | 11 | Generic RNDIS Mode on TX endpoint 4 |
| 5-4 | Tx3_mode | | |
| | | 00 | Transparent Mode on TX endpoint 3 |
| | | 01 | RNDIS MODE on TX endpoint 3 |
| | | 10 | CDC Mode on TX endpoint 3 |
| | | 11 | Generic RNDIS Mode on TX endpoint 3 |
| 3-2 | Tx2_mode | | |
| | | 00 | Transparent Mode on TX endpoint 2 |
| | | 01 | RNDIS MODE on TX endpoint 2 |
| | | 10 | CDC Mode on TX endpoint 2 |
| | | 11 | Generic RNDIS Mode on TX endpoint 2 |
| 1-0 | Tx1_mode | | |
| | | 00 | Transparent Mode on TX endpoint 1 |
| | | 01 | RNDIS MODE on TX endpoint 1 |
| | | 10 | CDC Mode on TX endpoint 1 |
| | | 11 | Generic RNDIS Mode on TX endpoint 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.15 USB0 Rx Mode Register (USB0RXMODE)

The USB0 Rx mode register (USB0RXMODE) register allows the CPU to individually enable RNDIS/Generic/CDC modes for each endpoint. Using the global RNDIS enable in the control register overrides this register and enable RNDIS mode for all endpoints.

The USB0 Rx mode register is shown in Figure 25-80 and described in Table 25-89.

**Figure 25-80. USB0 Rx Mode Register (USB0RXMODE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | Rx15_mode | | Rx14_mode | | Rx13_mode | | Rx12_mode | | Rx11_mode | | Rx10_mode | | Rx9_mode | |
| R-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rx8_mode | | Rx7_mode | | Rx6_mode | | Rx5_mode | | Rx4_mode | | Rx3_mode | | Rx2_mode | | Rx1_mode | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-89. USB0 Rx Mode Register (USB0RXMODE) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Always read as 0. Writes have no effect. |
| 29-28 | Rx15_mode | | |
| | | 00 | Transparent Mode on RX endpoint 15 |
| | | 01 | RNDIS MODE on RX endpoint 15 |
| | | 10 | CDC Mode on RX endpoint 15 |
| | | 11 | Generic RNDIS Mode on RX endpoint 15 |
| 27-26 | Rx14_mode | | |
| | | 00 | Transparent Mode on RX endpoint 14 |
| | | 01 | RNDIS MODE on RX endpoint 14 |
| | | 10 | CDC Mode on RX endpoint 14 |
| | | 11 | Generic RNDIS Mode on RX endpoint 14 |
| 25-24 | Rx13_mode | | |
| | | 00 | Transparent Mode on RX endpoint 13 |
| | | 01 | RNDIS MODE on RX endpoint 13 |
| | | 10 | CDC Mode on RX endpoint 13 |
| | | 11 | Generic RNDIS Mode on RX endpoint 13 |
| 23-22 | Rx12_mode | | |
| | | 00 | Transparent Mode on RX endpoint 12 |
| | | 01 | RNDIS MODE on RX endpoint 12 |
| | | 10 | CDC Mode on RX endpoint 12 |
| | | 11 | Generic RNDIS Mode on RX endpoint 12 |
| 21-20 | Rx11_mode | | |
| | | 00 | Transparent Mode on RX endpoint 11 |
| | | 01 | RNDIS MODE on RX endpoint 11 |
| | | 10 | CDC Mode on RX endpoint 11 |
| | | 11 | Generic RNDIS Mode on RX endpoint 11 |
| 19-18 | Rx10_mode | | |
| | | 00 | Transparent Mode on RX endpoint 10 |
| | | 01 | RNDIS MODE on RX endpoint 10 |
| | | 10 | CDC Mode on RX endpoint 10 |
| | | 11 | Generic RNDIS Mode on RX endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-89. USB0 Rx Mode Register (USB0RXMODE) Field Descriptions (continued)**

| Bits | Field | Value | Description |
|---|---|---|---|
| 17-16 | Rx9_mode | | |
| | | 00 | Transparent Mode on RX endpoint 9 |
| | | 01 | RNDIS MODE on RX endpoint 9 |
| | | 10 | CDC Mode on RX endpoint 9 |
| | | 11 | Generic RNDIS Mode on RX endpoint 9 |
| 15-14 | Rx8_mode | | |
| | | 00 | Transparent Mode on RX endpoint 8 |
| | | 01 | RNDIS MODE on RX endpoint 8 |
| | | 10 | CDC Mode on RX endpoint 8 |
| | | 11 | Generic RNDIS Mode on RX endpoint 8 |
| 13-12 | Rx7_mode | | |
| | | 00 | Transparent Mode on RX endpoint 7 |
| | | 01 | RNDIS MODE on RX endpoint 7 |
| | | 10 | CDC Mode on RX endpoint 7 |
| | | 11 | Generic RNDIS Mode on RX endpoint 7 |
| 11-10 | Rx6_mode | | |
| | | 00 | Transparent Mode on RX endpoint 6 |
| | | 01 | RNDIS MODE on RX endpoint 6 |
| | | 10 | CDC Mode on RX endpoint 6 |
| | | 11 | Generic RNDIS Mode on RX endpoint 6 |
| 9-8 | Rx5_mode | | |
| | | 00 | Transparent Mode on RX endpoint 5 |
| | | 01 | RNDIS MODE on RX endpoint 5 |
| | | 10 | CDC Mode on RX endpoint 5 |
| | | 11 | Generic RNDIS Mode on RX endpoint 5 |
| 7-6 | Rx4_mode | | |
| | | 00 | Transparent Mode on RX endpoint 4 |
| | | 01 | RNDIS MODE on RX endpoint 4 |
| | | 10 | CDC Mode on RX endpoint 4 |
| | | 11 | Generic RNDIS Mode on RX endpoint 4 |
| 5-4 | Rx3_mode | | |
| | | 00 | Transparent Mode on RX endpoint 3 |
| | | 01 | RNDIS MODE on RX endpoint 3 |
| | | 10 | CDC Mode on RX endpoint 3 |
| | | 11 | Generic RNDIS Mode on RX endpoint 3 |
| 3-2 | Rx2_mode | | |
| | | 00 | Transparent Mode on RX endpoint 2 |
| | | 01 | RNDIS MODE on RX endpoint 2 |
| | | 10 | CDC Mode on RX endpoint 2 |
| | | 11 | Generic RNDIS Mode on RX endpoint 2 |
| 1-0 | Rx1_mode | | |
| | | 00 | Transparent Mode on RX endpoint 1 |
| | | 01 | RNDIS MODE on RX endpoint 1 |
| | | 10 | CDC Mode on RX endpoint 1 |
| | | 11 | Generic RNDIS Mode on RX endpoint 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.16 USB0 Generic RNDIS EPn Size Register (USB0GENRNDISEPn)

The USB0 generic RNDIS EPn size register (USB0GENRNDISEPn) is programmed with a RNDIS packet size in bytes. When EP(n) is in Generic RNDIS mode, the received USB packets will be collected into a single CPPI packet that is completed when the number of bytes equal to the value of this register has been received, or a “short” packet is received. Note that this register must be programmed with a value that is an integer multiple of the endpoint size. The maximum value this register can be programmed with is 0x10000, or 65536. N can range from 1 to 15.

The USB0 generic RNDIS EPn size register is shown in Figure 25-81 and described in Table 25-90.

**Figure 25-81. USB0 Generic RNDIS EPn Size Register (USB0GENRNDISEPn)**

| 31 | 17 | 16 | 0 |
|---|---|---|---|
| Reserved | | Ep(n)_size | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-90. USB0 Generic RNDIS EPn Size Register (USB0GENRNDISEPn) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-17 | Reserved | Always read as 0. Writes have no effect. |
| 16-0 | Ep(n)_size | Generic RNDIS packet size. n=1-15 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.17 USB0 Auto Req Register (USB0AUTOREQ)

The USB0 auto req register (USB0AUTOREQ) allows the CPU to enable an automatic IN token request generation for host mode RX operation per each RX endpoint. This feature has the DMA set the ReqPkt bit in the RXCSR when it clears the RxPktRdy bit after reading out a packet. The ReqPkt bit is used by the core to generate an IN token to receive data. By using this feature, the host can automatically generate an IN token after the DMA finishes receiving data and empties an endpoint buffer, thus receiving the next data packet as soon as possible from the connected device. Without this feature, the CPU will have to manually set the ReqPkt bit for every USB packet.

There are two modes that Auto Req can function: always or all except an EOP. The always mode will set the ReqPkt bit after every USB packet the DMA receives thus generating a new IN token after each USB packet. The EOP mode will set the ReqPkt bit after every USB packet that isn't an EOP (end of packet) in the CPPI descriptor. For RNDIS, CDC, and Generic RNDIS modes the auto req will stop when the EOP is received (either via a short packet for RNDIS,CDC, and Generic RNDIS or the count is reached for Generic RNDIS), making it useful for starting a large RNDIS packet and having it auto generate IN tokens until the end of the RNDIS packet. For transparent mode every USB packet is an EOP CPPI packet so the auto req never functions and acts like auto req is disabled.

The USB0 auto req register is shown in Figure 25-82 and described in Table 25-91.

**Figure 25-82. USB0 Auto Req Register (USB0AUTOREQ)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | Rx15_autoreq | | Rx14_autoreq | | Rx13_autoreq | | Rx12_autoreq | | Rx11_autoreq | | Rx10_autoreq | | Rx9_autoreq | |
| R-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rx8_autoreq | | Rx7_autoreq | | Rx6_autoreq | | Rx5_autoreq | | Rx4_autoreq | | Rx3_autoreq | | Rx2_autoreq | | Rx1_autoreq | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-91. USB0 Auto Req Register (USB0AUTOREQ) Field Descriptions**

| Bits | Field | Values | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Always read as 0. Writes have no effect. |
| 29-28 | Rx15_autoreq | | RX endpoint 15 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 27-26 | Rx14_autoreq | | RX endpoint 14 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 25-24 | Rx13_autoreq | | RX endpoint 13 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 23-22 | Rx12_autoreq | | RX endpoint 12 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-91. USB0 Auto Req Register (USB0AUTOREQ) Field Descriptions (continued)**

| Bits | Field | Values | Description |
|---|---|---|---|
| 21-20 | Rx11_autoreq | | RX endpoint 11 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 19-18 | Rx10_autoreq | | RX endpoint 10 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 17-16 | Rx9_autoreq | | RX endpoint 9 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 15-14 | Rx8_autoreq | | RX endpoint 8 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 13-12 | Rx7_autoreq | | RX endpoint 7 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 11-10 | Rx6_autoreq | | RX endpoint 6 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 9-8 | Rx5_autoreq | | RX endpoint 5 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 7-6 | Rx4_autoreq | | RX endpoint 4 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 5-4 | Rx3_autoreq | | RX endpoint 3 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-91. USB0 Auto Req Register (USB0AUTOREQ) Field Descriptions (continued)**

| Bits | Field | Values | Description |
|---|---|---|---|
| 3-2 | Rx2_autoreq | | RX endpoint 2 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 1-0 | Rx1_autoreq | | RX endpoint 1 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.18 USB0 SRP Fix Time Register (USB0SRPFIXTIME)

The USB0 SRP fix time register (USB0SRPFIXTIME) allows the CPU to configure the maximum amount of time the SRP fix logic blocks the AVAID from the PHY to the OTG core. This time allows the VBUS signal the ability to get below the thresholds and therefore remove the chance of voltage bounces which could give false threshold measurements.

The USB0 SRP fix time register is shown in Figure 25-83 and described in Table 25-92.

**Figure 25-83. USB0 SRP Fix Time Register (USB0SRPFIXTIME)**

| 31 | 0 |
|---|---|
| srpfixtime | |
| R/W-0280de80h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-92. USB0 SRP Fix Time Register (USB0SRPFIXTIME) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-0 | srpfixtime | SRP Fix maximum time in 60 MHz cycles. Default is 700 ms. |

#### 25.9.2.1.19 USB0 Teardown Register (USB0TDOWN)

The USB0 teardown register (USB0TDOWN) controls the tearing down of rx and tx fifos in the USB controller. When a '1' is written to a valid bit in this register, the CPPI FIFO pointers for that endpoint are cleared, and the register automatically clears itself after 1 clock cycle. This register must be used in conjunction with the CPPI DMA Teardown mechanism. The host should also write the FlushFIFO bits in the TXCSR and RXCSR Mentor USB Controller registers to ensure a complete teardown of the endpoint. See the Mentor specification for details.

The USB0 teardown register is shown in Figure 25-84 and described in Table 25-93.

**Figure 25-84. USB0 Teardown Register (USB0TDOWN)**

| 31 | 17 | 16 | 15 | 0 |
|---|---|---|---|---|
| tx_tdown | | Rsvd | rx_tdown | Rsvd |
| R/W-0h | | R-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-93. USB0 Teardown Register (USB0TDOWN) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-17 | tx_tdown | Tx Endpoint Teardown. Write '1' to corresponding bit to set. Read as '0'.<br>Bit 31 = Endpoint 15<br>...<br>Bit 17 = Endpoint 1 |
| 16 | Reserved | Always read as 0. Writes have no effect. |
| 15-1 | rx_tdown | RX Endpoint Teardown. Write '1' to corresponding bit to set. Read as '0'.<br>Bit 15 = Endpoint 15<br>...<br>Bit 1 = Endpoint 1 |
| 0 | Reserved | Always read as 0. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.20 USB0 Threshold XDMA Idle Register

The USB0 Threshold XDMA Idle register provides a cycle threshold number that is programmable. This cycle threshold and cycle count is used by the XDMA state machine when the EP_STATE machine is in the IDLE mode. When the threshold is reached, the XDMA leaves the G_IDLE state and reads the TXCSR registers within the Mentor controller and possibly generates an interrupt for TX_FIFO[15:0].

See the XDMA section in the Integration Specification for more details about the various state machines.

The USB0 threshold XDMA idle register is shown in Figure 25-85 and described in Table 25-94.

**Figure 25-85. USB0 Threshold XDMA Idle Register**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | thres_xdma_idle | |
| R-0h | | R/W-18h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-94. USB0 Threshold XDMA Idle Register Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-8 | Reserved | Always read as 0. Writes have no effect. |
| 7-0 | thres_xdma_idle | Threshold XDMA Idle cycle parameter. Valid values are 0 to 255. Default is 24 cycles. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.21 USB0 PHY UTMI Register (USB0UTMI)

The USB0 PHY UTMI register (USB0UTMI) controls various PHY UTMI signals. The UTMI interface is located between the PHY module and the Mentor Graphics Controller. These signals are inputs to the PHY but are not outputs from the Mentor Graphics Controller.

The USB0 PHY UTMI register is shown in Figure 25-86 and described in Table 25-95.

**Figure 25-86. USB0 PHY UTMI Register (USB0UTMI)**

| 31 | | | | | | | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| txbitstuffen | Reserved | Reserved | Reserved | Reserved | txenablen | Reserved | Reserved |
| R/W-0h | R/W-0h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | | | | | | | 8 |
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 7 | | | | 3 | 2 | 1 | 0 |
| Reserved | | | | | Reserved | Reserved | Reserved |
| R-0h | | | | | R/W-0h | R/W-1h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-95. USB0 PHY UTMI Register (USB0UTMI) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | Reserved | Always read as 0. Writes have no effect. |
| 23 | txbitstuffen | PHY UTMI input for signal txbitstuffen |
| 22 | Reserved | Reserved, this bit has no function. |
| 21 | Reserved | Reserved, this bit has no function. |
| 20 | Reserved | Reserved, this bit has no function. |
| 19 | Reserved | Reserved, this bit has no function. |
| 18 | txenablen | PHY UTMI input for signal txenablen |
| 17 | Reserved | Reserved, this bit has no function. |
| 16 | Reserved | Reserved, this bit has no function. |
| 15-8 | Reserved | Reserved, this bit has no function. |
| 7-3 | Reserved | Reserved, this bit has no function. |
| 2 | Reserved | Reserved, this bit has no function. |
| 1 | Reserved | Reserved, this bit has no function. |
| 0 | Reserved | Reserved, this bit has no function. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.22 USB0 MGC UTMI Loopback Register (USB0UTMILB)

The USB0 MGC UTMI loopback register (USB0UTMILB) contains most of the input and output UTMI signals for the Mentor Graphics Controller. The UTMI interface is located between the PHY module and the Mentor Graphics Controller.

In the loopback mode test register bits 11 to 0 control various UTMI signals that are input to the Mentor Graphics Controller.

In the loopback mode test register bits 28 to 16 observe various UTMI signals that are output from the Mentor Graphics Controller.

The USB0 MGC UTMI loopback register is shown in Figure 25-87 and described in Table 25-96.

**Figure 25-87. USB0 MGC UTMI Loopback Register (USB0UTMILB)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | suspendm | suspendm | | txvalid | xcvrsel |
| R-0h | | | R-0h | R-0h | | R-0h | R-0h |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| xcvrsel | termsel | drvvbus | chrgvbus | dischrgvbus | dppulldown | dmpulldown | idpullup |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| Reserved | | | | iddig | hostdiscon | sessend | avalid |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| vbusvalid | rxerror | Reserved | | linestate | | Reserved | |
| R/W-0h | R/W-0h | R-0h | | R/W-1h | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-96. USB0 MGC UTMI Loopback Register (USB0UTMILB) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-29 | Reserved | Always read as 0. Writes have no effect. |
| 28 | suspendm | Loopback test observed value for suspendm |
| 27-26 | opmode | Loopback test observed value for opmode |
| 25 | txvalid | Loopback test observed value for txvalid |
| 24-23 | xcvrsel | Loopback test observed value for xcvrsel |
| 22 | termsel | Loopback test observed value for termsel |
| 21 | drvvbus | Loopback test observed value for drvvbus |
| 20 | chrgvbus | Loopback test observed value for chrgvbus |
| 19 | dischrgvbus | Loopback test observed value for dischrgvbus |
| 18 | dppulldown | Loopback test observed value for dppulldown |
| 17 | dmpulldown | Loopback test observed value for dmpulldown |
| 16 | idpullup | Loopback test observed value for idpullup |
| 15-12 | Reserved | Always read as 0. Writes have no effect. |
| 11 | iddig | Loopback test value for iddig |
| 10 | hostdiscon | Loopback test value for hostdiscon |
| 9 | sessend | Loopback test value for sessend |
| 8 | avalid | Loopback test value for avalid |
| 7 | vbusvalid | Loopback test value for vbusvalid |
| 6 | rxerror | Loopback test value for rxerror |
| 5-4 | Reserved | Always read as 0. Writes have no effect. |
| 3-2 | linestate | Loopback test value for linestate |
| 1-0 | Reserved | Always read as 0. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.1.23 USB0 Mode Register (USB0MODE)

The USB mode (USB0MODE) register allows the user a flexibility of controlling the role that the controller assumes. Via the setting of the USB0MODE, the user has the capability of allowing the control to be coming from a register field (USB0MODE[Bit8=iddig]) or from the USB_ID pin which is indirectly controlled by the cable end. if USB0MODE=0, the state of the USB_ID pin controls the role the controller assumes.

The loopback bit enables the loopback test. This test allows MGC0 to be connected to MGC1. It is important to set both loopback bits in both USB0/1 Mode registers. The USB0 MGC UTMI loopback register contains the various UTMI signals to be controlled and observed during loopback test.

The phy_test bit enables the phy_test mode. This test mode is intended to allow additional control of the UTMI inputs to the PHY. Currently, these inputs are drvvbus, dppulldown, dmpulldown, and idpullup. When phy_test is high, then the pin inputs for these signals control the inputs to the PHY instead of the Mentor controller outputs. The phy_test mode is not active with loopback mode is active.

When phy_test is active than the PHY inputs datainh is equal to datain. And txvalidh is equal to txvalid.

The USB0 mode register is shown in Figure 25-88 and described in Table 25-97.

**Figure 25-88. USB0 Mode Register (USB0MODE)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 9 | 8 | 7 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | iddig | Reserved | | phy_test | loopback |
| R-0h | | R/W-1h | R-0h | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-97. USB0 Mode Register (USB0MODE) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | | Always read as 0. Writes have no effect. |
| 8 | iddig | | MGC input value for iddig |
| | | 0 | A type |
| | | 1 | B type |
| 7-2 | Reserved | | Always read as 0. Writes have no effect. |
| 1 | phy_test | | PHY test |
| | | 0 | Normal mode |
| | | 1 | PHY test mode |
| 0 | loopback | | Loopback test mode |
| | | 0 | Normal mode |
| | | 1 | Loopback test mode |

#### 25.9.2.1.24 USB0 Mentor Core Registers

A description of the Mentor core registers is available in Section 25.9.6. Two instantiations of the USB core exists, USB0 and USB1. These registers reside back to back within USB0 space. The core registers that are used by USB0 subsystem are defined within offsets 1400h-159Ch, while the core registers that are used by USB1 subsystem are defined within offset 1C00h-1D9Ch.


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.2.2 USB1 Controller Registers

Table 25-98 lists the registers for the USB1 controller submodule.

**Table 25-98. USB1 Controller Registers**

| Address Offset | Acronym | USB1 Controller Register |
|---|---|---|
| 1800h | USB1REV | USB1 Revision Register |
| 1814h | USB1CTRL | USB1 Control Register |
| 1818h | USB1STAT | USB1 Status Register |
| 1820h | USB1IRQMSTAT | USB1 IRQ_MERGED_STATUS |
| 1824h | USB1IRQEOI | USB1 IRQ_EOI |
| 1828h | USB1IRQSTATRAW0 | USB1 IRQ_STATUS_RAW_0 |
| 182Ch | USB1IRQSTATRAW1 | USB1 IRQ_STATUS_RAW_1 |
| 1830h | USB1IRQSTAT0 | USB1 IRQ_STATUS_0 |
| 1834h | USB1IRQSTAT1 | USB1 IRQ_STATUS_1 |
| 1838h | USB1IRQENABLESET0 | USB1 IRQ_ENABLE_SET_0 |
| 183Ch | USB1IRQENABLESET1 | USB1 IRQ_ENABLE_SET_1 |
| 1840h | USB1IRQENABLECLR0 | USB1 IRQ_ENABLE_CLR_0 |
| 1844h | USB1IRQENABLECLR1 | USB1 IRQ_ENABLE_CLR_1 |
| 1870h | USB1TXMODE | USB1 Tx Mode Register |
| 1874h | USB1RXMODE | USB1 Rx Mode Register |
| 1880h | USB1GENRNDISEP1 | USB1 Generic RNDIS Size EP1 |
| 1884h | USB1GENRNDISEP2 | USB1 Generic RNDIS Size EP2 |
| 1888h | USB1GENRNDISEP3 | USB1 Generic RNDIS Size EP3 |
| 188Ch | USB1GENRNDISEP4 | USB1 Generic RNDIS Size EP4 |
| 1890h | USB1GENRNDISEP5 | USB1 Generic RNDIS Size EP5 |
| 1894h | USB1GENRNDISEP6 | USB1 Generic RNDIS Size EP6 |
| 1898h | USB1GENRNDISEP7 | USB1 Generic RNDIS Size EP7 |
| 189Ch | USB1GENRNDISEP8 | USB1 Generic RNDIS Size EP8 |
| 18A0h | USB1GENRNDISEP9 | USB1 Generic RNDIS Size EP9 |
| 18A4h | USB1GENRNDISEP10 | USB1 Generic RNDIS Size EP10 |
| 18A8h | USB1GENRNDISEP11 | USB1 Generic RNDIS Size EP11 |
| 18ACh | USB1GENRNDISEP12 | USB1 Generic RNDIS Size EP12 |
| 18B0h | USB1GENRNDISEP13 | USB1 Generic RNDIS Size EP13 |
| 18B4h | USB1GENRNDISEP14 | USB1 Generic RNDIS Size EP14 |
| 18B8h | USB1GENRNDISEP15 | USB1 Generic RNDIS Size EP15 |
| 18D0h | USB1AUTOREQ | USB1 Auto Req Register |
| 18D4h | USB1SRPFIXTIME | USB1 SRP Fix Time |
| 18D8h | USB1TDOWN | USB1 Teardown Register |
| 18DCh | | USB1 Threshold XDMA Idle Register |
| 18E0h | USB1UTMI | USB1 PHY UTMI Register |
| 18E4h | USB1UTMILB | USB1 MGC UTMI Loopback Register |
| 18E8h | USB1MODE | USB1 Mode Register |
| 1C00h–1C68h | ... | USB1 Mentor Core Registers/FIFOs |
| 1C6Ch | USB1HWVERS | USB1 Mentor Core HWVERS Register (see Section 25.9.7.2.6) |
| 1C70h-1D9Ch | ... | USB1 Mentor Core Registers/FIFOs |

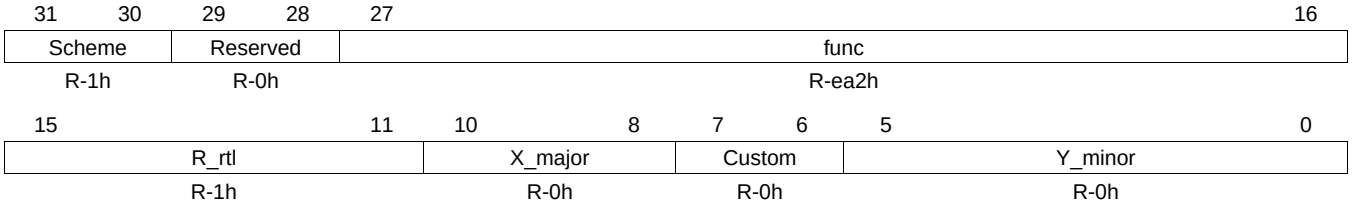

Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.1 USB1 Revision Register (USB1REV)

The USB1 revision register (USB1REV) contains the major and minor revisions for the USB1 module.

The USB1 revision register is shown in Figure 25-89 and described in Table 25-99.

**Figure 25-89. USB1 Revision Register (USB1REV)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-99. USB1 Revision Register (USB1REV) Field Descriptions**

| Bits | Field | Values | Description |
|---|---|---|---|
| 31-30 | Scheme | | Used to distinguish between old scheme and current. |
| | | 1 | Highlander 0.8 |
| 29-28 | Reserved | 0 | Reserved |
| 27-16 | func | EA2h | Function indicates a software compatible module family. |
| 15-11 | R_rtl | 1h | RTL revision. Will vary depending on release. |
| 10-8 | X_major | 0h | Major revision. |
| 7-6 | Custom | 0h | Custom revision |
| 5-0 | Y_minor | 0h | Minor revision. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.2 USB1 Control Register (USB1CTRL)

The USB1 control register (USB1CTRL) allows the CPU to control various aspects of the module. If uint is set high, then the Mentor controller generic interrupt for USB[9] will be generated (if enabled). This requires software to read the Mentor controller's registers to determine which interrupt USB[0] to USB[7] occurred. If uint is set low, then the usb20otg_f module will automatically read the Mentor controller's registers and set the appropriate interrupt USB[0] to USB[7] (if enabled). The generic interrupt for USB[9] will not be generated.

The USB1 control register is shown in Figure 25-90 and described in Table 25-100.

**Figure 25-90. USB1 Control Register (USB1CTRL)**

| 31 | 30 | 29 | 16 |
|---|---|---|---|
| dis_deb | dis_srp | Reserved | |
| R/W-0h | R/W-0h | R-0h | |

| 15 | 8 |
|---|---|
| Reserved | |
| R-0h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | soft reset isolation | rndis | uint | Reserved | clkfack | soft reset |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W1-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-100. USB1 Control Register (USB1CTRL) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31 | dis_deb | | Disable the VBUS debouncer circuit fix |
| 30 | dis_srp | | Disable the OTG Session Request Protocol (SRP) AVALID circuit fix. When enabled (=0) this allows additional time for the VBUS signal to be measured against the VBUS thresholds. The time is specified in the USB1 SRP Fix Time Register. |
| 29-06 | Reserved | 0 | Always read as 0. Writes have no effect. |
| 5 | soft reset isolation | | Soft reset isolation. When high this bit forces all USB1 signals that connect to the USBSS to zeros during a soft reset via bit 0 of this register. This bit should be set high prior to setting bit 0 and cleared after bit 0 is cleared. |
| 4 | rndis | | Global RNDIS mode enable for all endpoints. |
| 3 | uint | | USB non-highlander interrupt enable |
| | | 0 | Highlander |
| | | 1 | Non-highlander |
| 2 | Reserved | 0 | Always read as 0. Writes have no effect. |
| 1 | clkfack | | Clock stop fast ack enable. |
| 0 | soft reset | | Software reset of USB1. |
| | | Write 0 | Write 0 = no action |
| | | Write 1 | Write 1 = Initiate software reset |
| | | | Read 0 = Reset done, no action |
| | | | Read 1 = Reset ongoing |

If uint is set high, then the Mentor controller generic interrupt for USB[9] will be generated (if enabled). This requires software to read the Mentor controller's registers to determine which interrupt USB[0] to USB[7] occurred.

If uint is set low, then the usb20otg_f module will automatically read the Mentor controller's registers and set the appropriate interrupt USB[0] to USB[7] (if enabled). The generic interrupt for USB[9] will not be generated.


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.3 USB1 Status Register (USB1STAT)

The USB1 status register (USB1STAT) allows the CPU to check the voltage state level of USB1_DRVVBUS pin.

Ths USB1 status register is shown in Figure 25-91 and described in Table 25-101.

**Figure 25-91. USB1 Status Register (USB1STAT)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | drvvbus |
| R-0h | | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-101. USB1 Status Register (USB1STAT) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-1 | Reserved | Always read as 0. Writes have no effect. |
| 0 | drvvbus | Current USB1_DRVVBUS value |

#### 25.9.2.2.4 USB1 IRQ_MERGED_STATUS Register (USB1IRQMSTAT)

The USB1 IRQ_MERGED_STATUS register (USB1IRQMSTAT) contains a merged status for all the events for USB1. This register optimizes software accesses in a module where a large number of events are associated to one IRQ line. The merged status is read-only, and does not need to be actively cleared like the IRQ status. It will clear automatically.

The USB1 IRQ_MERGED_STATUS register is shown in Figure 25-92 and described in Table 25-102.

**Figure 25-92. USB1 IRQ_MERGED_STATUS Register (USB1IRQMSTAT)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | Bank1 | Bank0 |
| R-0h | | R-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-102. USB1 IRQ_MERGED_STATUS Register (USB1IRQMSTAT) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Always read as 0. Writes have no effect. |
| 1 | Bank1 | 0 | No events pending from IRQ_STATUS_1 |
| | | 1 | At least one event is pending from IRQ_STATUS_1 |
| 0 | Bank0 | 0 | No events pending from IRQ_STATUS_0 |
| | | 1 | At least one event is pending from IRQ_STATUS_0 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.5 USB1 IRQ_EOI Register (USB1IRQEOI)

The USB1 IRQ_EOI register (USB1IRQEOI) allows the CPU to acknowledge completion of an interrupt by writing to the end of interrupt (EOI). An eoi_write signal will be generated and another interrupt will be triggered if interrupt sources remain. This register will be reset one cycle after it has been written to.

The USB1 IRQ_EOI register is shown in Figure 25-93 and described in Table 25-103.

**Figure 25-93. USB1 IRQ_EOI Register (USB1IRQEOI)**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | EOI for USB1 |
| R-0h | | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-103. USB1 IRQ_EOI Register (USB1IRQEOI) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-1 | Reserved | Always read as 0. Writes have no effect. |
| 0 | EOI for USB1 | EOI for USB1 interrupt |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.6 USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW0)

The USB1 IRQ_STATUS_RAW_0 register (USB1IRQSTATRAW0) allows the USB1 interrupt sources to be manually triggered when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Set event
Read 0: No event pending
Read 1: Event pending

The USB1 IRQ_STATUS_RAW_0 register is shown in Figure 25-94 and described in Table 25-104.

**Figure 25-94. USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RX EP 15 | RX EP 14 | RX EP 13 | RX EP 12 | RX EP 11 | RX EP 10 | RX EP 9 | RX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RX EP 7 | RX EP 6 | RX EP 5 | RX EP 4 | RX EP 3 | RX EP 2 | RX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TX EP 15 | TX EP 14 | TX EP 13 | TX EP 12 | TX EP 11 | TX EP 10 | TX EP 9 | TX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX EP 7 | TX EP 6 | TX EP 5 | TX EP 4 | TX EP 3 | TX EP 2 | TX EP 1 | TX EP 0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-104. USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW0) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | RX EP 15 | Interrupt status for RX endpoint 15 |
| 30 | RX EP 14 | Interrupt status for RX endpoint 14 |
| 29 | RX EP 13 | Interrupt status for RX endpoint 13 |
| 28 | RX EP 12 | Interrupt status for RX endpoint 12 |
| 27 | RX EP 11 | Interrupt status for RX endpoint 11 |
| 26 | RX EP 10 | Interrupt status for RX endpoint 10 |
| 25 | RX EP 9 | Interrupt status for RX endpoint 9 |
| 24 | RX EP 8 | Interrupt status for RX endpoint 8 |
| 23 | RX EP 7 | Interrupt status for RX endpoint 7 |
| 22 | RX EP 6 | Interrupt status for RX endpoint 6 |
| 21 | RX EP 5 | Interrupt status for RX endpoint 5 |
| 20 | RX EP 4 | Interrupt status for RX endpoint 4 |
| 19 | RX EP 3 | Interrupt status for RX endpoint 3 |
| 18 | RX EP 2 | Interrupt status for RX endpoint 2 |
| 17 | RX EP 1 | Interrupt status for RX endpoint 1 |
| 16 | Reserved | Always read 0. Writes have no effect. |
| 15 | TX EP 15 | Interrupt status for TX endpoint 15 |
| 14 | TX EP 14 | Interrupt status for TX endpoint 14 |
| 13 | TX EP 13 | Interrupt status for TX endpoint 13 |
| 12 | TX EP 12 | Interrupt status for TX endpoint 12 |
| 11 | TX EP 11 | Interrupt status for TX endpoint 11 |
| 10 | TX EP 10 | Interrupt status for TX endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-104. USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW0) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 9 | TX EP 9 | Interrupt status for TX endpoint 9 |
| 8 | TX EP 8 | Interrupt status for TX endpoint 8 |
| 7 | TX EP 7 | Interrupt status for TX endpoint 7 |
| 6 | TX EP 6 | Interrupt status for TX endpoint 6 |
| 5 | TX EP 5 | Interrupt status for TX endpoint 5 |
| 4 | TX EP 4 | Interrupt status for TX endpoint 4 |
| 3 | TX EP 3 | Interrupt status for TX endpoint 3 |
| 2 | TX EP 2 | Interrupt status for TX endpoint 2 |
| 1 | TX EP 1 | Interrupt status for TX endpoint 1 |
| 0 | TX EP 0 | Interrupt status for TX endpoint 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.7 USB1 IRQ_STATUS_RAW_1 Register (USB1IRQSTATRAW1)

The USB1 IRQ_STATUS_RAW_1 register (USB1IRQSTATRAW1) allows the USB1 interrupt sources to be manually triggered when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Set event
Read 0: No event pending
Read 1: Event pending

The USB1 IRQ_STATUS_RAW_1 register is shown in Figure 25-95 and described in Table 25-105.

**Figure 25-95. USB1 IRQ_STATUS_RAW_1 Register (USB1IRQSTATRAW1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TX FIFO 15 | TX FIFO 14 | TX FIFO 13 | TX FIFO 12 | TX FIFO 11 | TX FIFO 10 | TX FIFO 9 | TX FIFO 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TX FIFO 7 | TX FIFO 6 | TX FIFO 5 | TX FIFO 4 | TX FIFO 3 | TX FIFO 2 | TX FIFO 1 | TX FIFO 0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | | | | | 10 | 9 | 8 |
| Reserved | | | | | | USB[9] | USB[8] |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USB[7] | USB[6] | USB[5] | USB[4] | USB[3] | USB[2] | USB[1] | USB[0] |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-105. USB1 IRQ_STATUS_RAW_1 Register (USB1IRQSTATRAW1) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | TX FIFO 15 | Interrupt status for TX FIFO endpoint 15 |
| 30 | TX FIFO 14 | Interrupt status for TX FIFO endpoint 14 |
| 29 | TX FIFO 13 | Interrupt status for TX FIFO endpoint 13 |
| 28 | TX FIFO 12 | Interrupt status for TX FIFO endpoint 12 |
| 27 | TX FIFO 11 | Interrupt status for TX FIFO endpoint 11 |
| 26 | TX FIFO 10 | Interrupt status for TX FIFO endpoint 10 |
| 25 | TX FIFO 9 | Interrupt status for TX FIFO endpoint 9 |
| 24 | TX FIFO 8 | Interrupt status for TX FIFO endpoint 8 |
| 23 | TX FIFO 7 | Interrupt status for TX FIFO endpoint 7 |
| 22 | TX FIFO 6 | Interrupt status for TX FIFO endpoint 6 |
| 21 | TX FIFO 5 | Interrupt status for TX FIFO endpoint 5 |
| 20 | TX FIFO 4 | Interrupt status for TX FIFO endpoint 4 |
| 19 | TX FIFO 3 | Interrupt status for TX FIFO endpoint 3 |
| 18 | TX FIFO 2 | Interrupt status for TX FIFO endpoint 2 |
| 17 | TX FIFO 1 | Interrupt status for TX FIFO endpoint 1 |
| 16 | TX FIFO 0 | Interrupt status for TX FIFO endpoint 0 |
| 15-10 | Reserved | Always read 0. Writes have no effect. |
| 9 | USB[9] | Interrupt status for Mentor controller USB_INT generic interrupt |
| 8 | USB[8] | Interrupt status for DRVVBUS level change |
| 7 | USB[7] | Interrupt status for VBUS < VBUS valid threshold |
| 6 | USB[6] | Interrupt status for SRP detected |
| 5 | USB[5] | Interrupt status for device disconnected (host mode) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-105. USB1 IRQ_STATUS_RAW_1 Register (USB1IRQSTATRAW1) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 4 | USB[4] | Interrupt status for device connected (host mode) |
| 3 | USB[3] | Interrupt status for SOF started |
| 2 | USB[2] | Interrupt status for Reset signaling detected (peripheral mode)<br>Babble detected (host mode) |
| 1 | USB[1] | Interrupt status for Resume signaling detected |
| 0 | USB[0] | Interrupt status for Suspend signaling detected |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.8 USB1 IRQ_STATUS_0 Register (USB1IRQSTAT0)

The USB1 IRQ_STATUS_0 register (USB1IRQSTAT0) allows the USB1 interrupt sources to be manually cleared when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Clear event
Read 0: No event pending
Read 1: Event pending

The USB1 IRQ_STATUS_0 register is shown in Figure 25-96 and described in Table 25-106.

**Figure 25-96. USB1 IRQ_STATUS_0 Register (USB1IRQSTAT0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RX EP 15 | RX EP 14 | RX EP 13 | RX EP 12 | RX EP 11 | RX EP 10 | RX EP 9 | RX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RX EP 7 | RX EP 6 | RX EP 5 | RX EP 4 | RX EP 3 | RX EP 2 | RX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TX EP 15 | TX EP 14 | TX EP 13 | TX EP 12 | TX EP 11 | TX EP 10 | TX EP 9 | TX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX EP 7 | TX EP 6 | TX EP 5 | TX EP 4 | TX EP 3 | TX EP 2 | TX EP 1 | TX EP 0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-106. USB1 IRQ_STATUS_0 Register (USBIRQSTAT0) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | RX EP 15 | Interrupt status for RX endpoint 15 |
| 30 | RX EP 14 | Interrupt status for RX endpoint 14 |
| 29 | RX EP 13 | Interrupt status for RX endpoint 13 |
| 28 | RX EP 12 | Interrupt status for RX endpoint 12 |
| 27 | RX EP 11 | Interrupt status for RX endpoint 11 |
| 26 | RX EP 10 | Interrupt status for RX endpoint 10 |
| 25 | RX EP 9 | Interrupt status for RX endpoint 9 |
| 24 | RX EP 8 | Interrupt status for RX endpoint 8 |
| 23 | RX EP 7 | Interrupt status for RX endpoint 7 |
| 22 | RX EP 6 | Interrupt status for RX endpoint 6 |
| 21 | RX EP 5 | Interrupt status for RX endpoint 5 |
| 20 | RX EP 4 | Interrupt status for RX endpoint 4 |
| 19 | RX EP 3 | Interrupt status for RX endpoint 3 |
| 18 | RX EP 2 | Interrupt status for RX endpoint 2 |
| 17 | RX EP 1 | Interrupt status for RX endpoint 1 |
| 16 | Reserved | Always read 0. Writes have no effect. |
| 15 | TX EP 15 | Interrupt status for TX endpoint 15 |
| 14 | TX EP 14 | Interrupt status for TX endpoint 14 |
| 13 | TX EP 13 | Interrupt status for TX endpoint 13 |
| 12 | TX EP 12 | Interrupt status for TX endpoint 12 |
| 11 | TX EP 11 | Interrupt status for TX endpoint 11 |
| 10 | TX EP 10 | Interrupt status for TX endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-106. USB1 IRQ_STATUS_0 Register (USBIRQSTAT0) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 9 | TX EP 9 | Interrupt status for TX endpoint 9 |
| 8 | TX EP 8 | Interrupt status for TX endpoint 8 |
| 7 | TX EP 7 | Interrupt status for TX endpoint 7 |
| 6 | TX EP 6 | Interrupt status for TX endpoint 6 |
| 5 | TX EP 5 | Interrupt status for TX endpoint 5 |
| 4 | TX EP 4 | Interrupt status for TX endpoint 4 |
| 3 | TX EP 3 | Interrupt status for TX endpoint 3 |
| 2 | TX EP 2 | Interrupt status for TX endpoint 2 |
| 1 | TX EP 1 | Interrupt status for TX endpoint 1 |
| 0 | TX EP 0 | Interrupt status for TX endpoint 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.2.2.9 USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1)

The USB1 IRQ_STATUS_1 register (USB1IRQSTAT1) allows the USB1 interrupt sources to be manually cleared when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Clear event
Read 0: No event pending
Read 1: Event pending

The USB1 IRQ_STATUS_1 register is shown in Figure 25-97 and described in Table 25-107.

**Figure 25-97. USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TX FIFO 15 | TX FIFO 14 | TX FIFO 13 | TX FIFO 12 | TX FIFO 11 | TX FIFO 10 | TX FIFO 9 | TX FIFO 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TX FIFO 7 | TX FIFO 6 | TX FIFO 5 | TX FIFO 4 | TX FIFO 3 | TX FIFO 2 | TX FIFO 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | USB[9] | USB[9] |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USB[7] | USB[6] | USB[5] | USB[4] | USB[3] | USB[2] | USB[1] | USB[0] |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-107. USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | TX FIFO 15 | Interrupt status for TX FIFO endpoint 15 |
| 30 | TX FIFO 14 | Interrupt status for TX FIFO endpoint 14 |
| 29 | TX FIFO 13 | Interrupt status for TX FIFO endpoint 13 |
| 28 | TX FIFO 12 | Interrupt status for TX FIFO endpoint 12 |
| 27 | TX FIFO 11 | Interrupt status for TX FIFO endpoint 11 |
| 26 | TX FIFO 10 | Interrupt status for TX FIFO endpoint 10 |
| 25 | TX FIFO 9 | Interrupt status for TX FIFO endpoint 9 |
| 24 | TX FIFO 8 | Interrupt status for TX FIFO endpoint 8 |
| 23 | TX FIFO 7 | Interrupt status for TX FIFO endpoint 7 |
| 22 | TX FIFO 6 | Interrupt status for TX FIFO endpoint 6 |
| 21 | TX FIFO 5 | Interrupt status for TX FIFO endpoint 5 |
| 20 | TX FIFO 4 | Interrupt status for TX FIFO endpoint 4 |
| 19 | TX FIFO 3 | Interrupt status for TX FIFO endpoint 3 |
| 18 | TX FIFO 2 | Interrupt status for TX FIFO endpoint 2 |
| 17 | TX FIFO 1 | Interrupt status for TX FIFO endpoint 1 |
| 16-10 | Reserved | Always read 0. Writes have no effect. |
| 9 | USB[9] | Interrupt status for Mentor controller USB_INT generic interrupt |
| 8 | USB[8] | Interrupt status for DRVVBUS level change |
| 7 | USB[7] | Interrupt status for VBUS < VBUS valid threshold |
| 6 | USB[6] | Interrupt status for SRP detected |
| 5 | USB[5] | Interrupt status for device disconnected (host mode) |
| 4 | USB[4] | Interrupt status for device connected (host mode) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-107. USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 3 | USB[3] | Interrupt status for SOF started |
| 2 | USB[2] | Interrupt status for Reset signaling detected (peripheral mode)<br>Babble detected (host mode) |
| 1 | USB[1] | Interrupt status for Resume signaling detected |
| 0 | USB[0] | Interrupt status for Suspend signaling detected |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.10 USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0)

The USB1 IRQ_ENABLE_SET_0 register (USB1IRQENABLESET0) allows the USB1 interrupt sources to be manually enabled when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt enabled value.

General actions per bit:
Write 0: No action
Write 1: Interrupt enabled
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB1 IRQ_ENABLE_SET_0 register is shown in Figure 25-98 and described in Table 25-108.

**Figure 25-98. USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RX EP 15 | RX EP 14 | RX EP 13 | RX EP 12 | RX EP 11 | RX EP 10 | RX EP 9 | RX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RX EP 7 | RX EP 6 | RX EP 5 | RX EP 4 | RX EP 3 | RX EP 2 | RX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TX EP 15 | TX EP 14 | TX EP 13 | TX EP 12 | TX EP 11 | TX EP 10 | TX EP 9 | TX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX EP 7 | TX EP 6 | TX EP 5 | TX EP 4 | TX EP 3 | TX EP 2 | TX EP 1 | TX EP 0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-108. USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | RX EP 15 | Interrupt enable for RX endpoint 15 |
| 30 | RX EP 14 | Interrupt enable for RX endpoint 14 |
| 29 | RX EP 13 | Interrupt enable for RX endpoint 13 |
| 28 | RX EP 12 | Interrupt enable for RX endpoint 12 |
| 27 | RX EP 11 | Interrupt enable for RX endpoint 11 |
| 26 | RX EP 10 | Interrupt enable for RX endpoint 10 |
| 25 | RX EP 9 | Interrupt enable for RX endpoint 9 |
| 24 | RX EP 8 | Interrupt enable for RX endpoint 8 |
| 23 | RX EP 7 | Interrupt enable for RX endpoint 7 |
| 22 | RX EP 6 | Interrupt enable for RX endpoint 6 |
| 21 | RX EP 5 | Interrupt enable for RX endpoint 5 |
| 20 | RX EP 4 | Interrupt enable for RX endpoint 4 |
| 19 | RX EP 3 | Interrupt enable for RX endpoint 3 |
| 18 | RX EP 2 | Interrupt enable for RX endpoint 2 |
| 17 | RX EP 1 | Interrupt enable for RX endpoint 1 |
| 16 | Reserved | Always read 0. Writes have no effect. |
| 15 | TX EP 15 | Interrupt enable for TX endpoint 15 |
| 14 | TX EP 14 | Interrupt enable for TX endpoint 14 |
| 13 | TX EP 13 | Interrupt enable for TX endpoint 13 |
| 12 | TX EP 12 | Interrupt enable for TX endpoint 12 |
| 11 | TX EP 11 | Interrupt enable for TX endpoint 11 |
| 10 | TX EP 10 | Interrupt enable for TX endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-108. USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 9 | TX EP 9 | Interrupt enable for TX endpoint 9 |
| 8 | TX EP 8 | Interrupt enable for TX endpoint 8 |
| 7 | TX EP 7 | Interrupt enable for TX endpoint 7 |
| 6 | TX EP 6 | Interrupt enable for TX endpoint 6 |
| 5 | TX EP 5 | Interrupt enable for TX endpoint 5 |
| 4 | TX EP 4 | Interrupt enable for TX endpoint 4 |
| 3 | TX EP 3 | Interrupt enable for TX endpoint 3 |
| 2 | TX EP 2 | Interrupt enable for TX endpoint 2 |
| 1 | TX EP 1 | Interrupt enable for TX endpoint 1 |
| 0 | TX EP 0 | Interrupt enable for TX endpoint 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.11 USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1)

The USB1 IRQ_ENABLE_SET_1 register (USB1IRQENABLESET1) allows the USB1 interrupt sources to be manually enabled when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt enabled value.

General actions per bit:
Write 0: No action
Write 1: Interrupt enabled
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB1 IRQ_ENABLE_SET_1 register is shown in Figure 25-99 and described in Table 25-109.

**Figure 25-99. USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TX FIFO 15 | TX FIFO 14 | TX FIFO 13 | TX FIFO 12 | TX FIFO 11 | TX FIFO 10 | TX FIFO 9 | TX FIFO 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TX FIFO 15 | TX FIFO 15 | TX FIFO 15 | TX FIFO 15 | TX FIFO 15 | TX FIFO 15 | TX FIFO 15 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | USB[9] | USB[9] |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USB[7] | USB[6] | USB[5] | USB[4] | USB[3] | USB[2] | USB[1] | USB[0] |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-109. USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | TX FIFO 15 | Interrupt enable for TX FIFO endpoint 15 |
| 30 | TX FIFO 14 | Interrupt enable for TX FIFO endpoint 14 |
| 29 | TX FIFO 13 | Interrupt enable for TX FIFO endpoint 13 |
| 28 | TX FIFO 12 | Interrupt enable for TX FIFO endpoint 12 |
| 27 | TX FIFO 11 | Interrupt enable for TX FIFO endpoint 11 |
| 26 | TX FIFO 10 | Interrupt enable for TX FIFO endpoint 10 |
| 25 | TX FIFO 9 | Interrupt enable for TX FIFO endpoint 9 |
| 24 | TX FIFO 8 | Interrupt enable for TX FIFO endpoint 8 |
| 23 | TX FIFO 7 | Interrupt enable for TX FIFO endpoint 7 |
| 22 | TX FIFO 6 | Interrupt enable for TX FIFO endpoint 6 |
| 21 | TX FIFO 5 | Interrupt enable for TX FIFO endpoint 5 |
| 20 | TX FIFO 4 | Interrupt enable for TX FIFO endpoint 4 |
| 19 | TX FIFO 3 | Interrupt enable for TX FIFO endpoint 3 |
| 18 | TX FIFO 2 | Interrupt enable for TX FIFO endpoint 2 |
| 17 | TX FIFO 1 | Interrupt enable for TX FIFO endpoint 1 |
| 16-10 | Reserved | Always read 0. Writes have no effect. |
| 9 | USB[9] | Interrupt enable for Mentor controller USB_INT generic interrupt |
| 8 | USB[8] | Interrupt enable for DRVVBUS level change |
| 7 | USB[7] | Interrupt enable for VBUS < VBUS valid threshold |
| 6 | USB[6] | Interrupt enable for SRP detected |
| 5 | USB[5] | Interrupt enable for device disconnected (host mode) |
| 4 | USB[4] | Interrupt enable for device connected (host mode) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-109. USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 3 | USB[3] | Interrupt enable for SOF started |
| 2 | USB[2] | Interrupt enable for Reset signaling detected (peripheral mode)<br>Babble detected (host mode) |
| 1 | USB[1] | Interrupt enable for Resume signaling detected |
| 0 | USB[0] | Interrupt enable for Suspend signaling detected |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.12 USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0)

The USB1 IRQ_ENABLE_CLR_0 register (USB1IRQENABLECLR0) allows the USB1 interrupt sources to be manually disabled when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt enabled value.

General actions per bit:
Write 0: No action
Write 1: Disable interrupt
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB1 IRQ_ENABLE_CLR_0 register is shown in Figure 25-100 and described in Table 25-110.

**Figure 25-100. USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RX EP 15 | RX EP 14 | RX EP 13 | RX EP 12 | RX EP 11 | RX EP 10 | RX EP 9 | RX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RX EP 7 | RX EP 6 | RX EP 5 | RX EP 4 | RX EP 3 | RX EP 2 | RX EP 1 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TX EP 15 | TX EP 14 | TX EP 13 | TX EP 12 | TX EP 11 | TX EP 10 | TX EP 9 | TX EP 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX EP 7 | TX EP 6 | TX EP 5 | TX EP 4 | TX EP 3 | TX EP 2 | TX EP 1 | TX EP 0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-110. USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | RX EP 15 | Interrupt enable for RX endpoint 15 |
| 30 | RX EP 14 | Interrupt enable for RX endpoint 14 |
| 29 | RX EP 13 | Interrupt enable for RX endpoint 13 |
| 28 | RX EP 12 | Interrupt enable for RX endpoint 12 |
| 27 | RX EP 11 | Interrupt enable for RX endpoint 11 |
| 26 | RX EP 10 | Interrupt enable for RX endpoint 10 |
| 25 | RX EP 9 | Interrupt enable for RX endpoint 9 |
| 24 | RX EP 8 | Interrupt enable for RX endpoint 8 |
| 23 | RX EP 7 | Interrupt enable for RX endpoint 7 |
| 22 | RX EP 6 | Interrupt enable for RX endpoint 6 |
| 21 | RX EP 5 | Interrupt enable for RX endpoint 5 |
| 20 | RX EP 4 | Interrupt enable for RX endpoint 4 |
| 19 | RX EP 3 | Interrupt enable for RX endpoint 3 |
| 18 | RX EP 2 | Interrupt enable for RX endpoint 2 |
| 17 | RX EP 1 | Interrupt enable for RX endpoint 1 |
| 16 | Reserved | Always read 0. Writes have no effect. |
| 15 | TX EP 15 | Interrupt enable for TX endpoint 15 |
| 14 | TX EP 14 | Interrupt enable for TX endpoint 14 |
| 13 | TX EP 13 | Interrupt enable for TX endpoint 13 |
| 12 | TX EP 12 | Interrupt enable for TX endpoint 12 |
| 11 | TX EP 11 | Interrupt enable for TX endpoint 11 |
| 10 | TX EP 10 | Interrupt enable for TX endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-110. USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 9 | TX EP 9 | Interrupt enable for TX endpoint 9 |
| 8 | TX EP 8 | Interrupt enable for TX endpoint 8 |
| 7 | TX EP 7 | Interrupt enable for TX endpoint 7 |
| 6 | TX EP 6 | Interrupt enable for TX endpoint 6 |
| 5 | TX EP 5 | Interrupt enable for TX endpoint 5 |
| 4 | TX EP 4 | Interrupt enable for TX endpoint 4 |
| 3 | TX EP 3 | Interrupt enable for TX endpoint 3 |
| 2 | TX EP 2 | Interrupt enable for TX endpoint 2 |
| 1 | TX EP 1 | Interrupt enable for TX endpoint 1 |
| 0 | TX EP 0 | Interrupt enable for TX endpoint 0 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.13 USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1)

The USB1 IRQ_ENABLE_CLR_1 register (USB1IREENABLECLR1) allows the USB1 interrupt sources to be manually disabled when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt enabled value.

General actions per bit:
Write 0: No action
Write 1: Disable interrupt
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB1 IRQ_ENABLE_CLR_1 register is shown in Figure 25-101 and described in Table 25-111.

**Figure 25-101. USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TX FIFO 15 | TX FIFO 14 | TX FIFO 13 | TX FIFO 12 | TX FIFO 11 | TX FIFO 10 | TX FIFO 9 | TX FIFO 8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TX FIFO 15 | TX FIFO 15 | TX FIFO 15 | TX FIFO 15 | TX FIFO 15 | TX FIFO 15 | TX FIFO 15 | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | USB[9] | USB[9] |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USB[7] | USB[6] | USB[5] | USB[4] | USB[3] | USB[2] | USB[1] | USB[0] |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-111. USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31 | TX FIFO 15 | Interrupt enable for TX FIFO endpoint 15 |
| 30 | TX FIFO 14 | Interrupt enable for TX FIFO endpoint 14 |
| 29 | TX FIFO 13 | Interrupt enable for TX FIFO endpoint 13 |
| 28 | TX FIFO 12 | Interrupt enable for TX FIFO endpoint 12 |
| 27 | TX FIFO 11 | Interrupt enable for TX FIFO endpoint 11 |
| 26 | TX FIFO 10 | Interrupt enable for TX FIFO endpoint 10 |
| 25 | TX FIFO 9 | Interrupt enable for TX FIFO endpoint 9 |
| 24 | TX FIFO 8 | Interrupt enable for TX FIFO endpoint 8 |
| 23 | TX FIFO 7 | Interrupt enable for TX FIFO endpoint 7 |
| 22 | TX FIFO 6 | Interrupt enable for TX FIFO endpoint 6 |
| 21 | TX FIFO 5 | Interrupt enable for TX FIFO endpoint 5 |
| 20 | TX FIFO 4 | Interrupt enable for TX FIFO endpoint 4 |
| 19 | TX FIFO 3 | Interrupt enable for TX FIFO endpoint 3 |
| 18 | TX FIFO 2 | Interrupt enable for TX FIFO endpoint 2 |
| 17 | TX FIFO 1 | Interrupt enable for TX FIFO endpoint 1 |
| 16-10 | Reserved | Always read 0. Writes have no effect. |
| 9 | USB[9] | Interrupt enable for Mentor controller USB_INT generic interrupt |
| 8 | USB[8] | Interrupt enable for DRVVBUS level change |
| 7 | USB[7] | Interrupt enable for VBUS < VBUS valid threshold |
| 6 | USB[6] | Interrupt enable for SRP detected |
| 5 | USB[5] | Interrupt enable for device disconnected (host mode) |
| 4 | USB[4] | Interrupt enable for device connected (host mode) |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-111. USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1) Field Descriptions (continued)**

| Bits | Field | Description |
|---|---|---|
| 3 | USB[3] | Interrupt enable for SOF started |
| 2 | USB[2] | Interrupt enable for Reset signaling detected (peripheral mode)<br>Babble detected (host mode) |
| 1 | USB[1] | Interrupt enable for Resume signaling detected |
| 0 | USB[0] | Interrupt enable for Suspend signaling detected |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.14 USB1 Tx Mode Register (USB1TXMODE)

The USB1 Tx mode register (USB1TXMODE) allows the CPU to individually enable RNDIS/Generic/CDC modes for each endpoint. Using the global RNDIS enable in the Control Register overrides this register and enable RNDIS mode for all endpoints.

The USB1 Tx mode register is shown in Figure 25-102 and described in Table 25-112.

**Figure 25-102. USB1 Tx Mode Register (USB1TXMODE)**

| 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 |
|---|---|---|---|---|---|---|---|
| Reserved | Tx15_mode | Tx14_mode | Tx13_mode | Tx12_mode | Tx11_mode | Tx10_mode | Tx9_mode |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| **15 14** | **13 12** | **11 10** | **9 8** | **7 6** | **5 4** | **3 2** | **1 0** |
| Tx8_mode | Tx7_mode | Tx6_mode | Tx5_mode | Tx4_mode | Tx3_mode | Tx2_mode | Tx1_mode |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-112. USB1 Tx Mode Register (USB1TXMODE) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Always read as 0. Writes have no effect. |
| 29-28 | Tx15_mode | | |
| | | 00 | Transparent Mode on TX endpoint 15 |
| | | 01 | RNDIS MODE on TX endpoint 15 |
| | | 10 | CDC Mode on TX endpoint 15 |
| | | 11 | Generic RNDIS Mode on TX endpoint 15 |
| 27-26 | Tx14_mode | | |
| | | 00 | Transparent Mode on TX endpoint 14 |
| | | 01 | RNDIS MODE on TX endpoint 14 |
| | | 10 | CDC Mode on TX endpoint 14 |
| | | 11 | Generic RNDIS Mode on TX endpoint 14 |
| 25-24 | Tx13_mode | | |
| | | 00 | Transparent Mode on TX endpoint 13 |
| | | 01 | RNDIS MODE on TX endpoint 13 |
| | | 10 | CDC Mode on TX endpoint 13 |
| | | 11 | Generic RNDIS Mode on TX endpoint 13 |
| 23-22 | Tx12_mode | | |
| | | 00 | Transparent Mode on TX endpoint 12 |
| | | 01 | RNDIS MODE on TX endpoint 12 |
| | | 10 | CDC Mode on TX endpoint 12 |
| | | 11 | Generic RNDIS Mode on TX endpoint 12 |
| 21-20 | Tx11_mode | | |
| | | 00 | Transparent Mode on TX endpoint 11 |
| | | 01 | RNDIS MODE on TX endpoint 11 |
| | | 10 | CDC Mode on TX endpoint 11 |
| | | 11 | Generic RNDIS Mode on TX endpoint 11 |
| 19-18 | Tx10_mode | | |
| | | 00 | Transparent Mode on TX endpoint 10 |
| | | 01 | RNDIS MODE on TX endpoint 10 |
| | | 10 | CDC Mode on TX endpoint 10 |
| | | 11 | Generic RNDIS Mode on TX endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-112. USB1 Tx Mode Register (USB1TXMODE) Field Descriptions (continued)**

| Bits | Field | Value | Description |
|---|---|---|---|
| 17-16 | Tx9_mode | | |
| | | 00 | Transparent Mode on TX endpoint 9 |
| | | 01 | RNDIS MODE on TX endpoint 9 |
| | | 10 | CDC Mode on TX endpoint 9 |
| | | 11 | Generic RNDIS Mode on TX endpoint 9 |
| 15-14 | Tx8_mode | | |
| | | 00 | Transparent Mode on TX endpoint 8 |
| | | 01 | RNDIS MODE on TX endpoint 8 |
| | | 10 | CDC Mode on TX endpoint 8 |
| | | 11 | Generic RNDIS Mode on TX endpoint 8 |
| 13-12 | Tx7_mode | | |
| | | 00 | Transparent Mode on TX endpoint 7 |
| | | 01 | RNDIS MODE on TX endpoint 7 |
| | | 10 | CDC Mode on TX endpoint 7 |
| | | 11 | Generic RNDIS Mode on TX endpoint 7 |
| 11-10 | Tx6_mode | | |
| | | 00 | Transparent Mode on TX endpoint 6 |
| | | 01 | RNDIS MODE on TX endpoint 6 |
| | | 10 | CDC Mode on TX endpoint 6 |
| | | 11 | Generic RNDIS Mode on TX endpoint 6 |
| 9-8 | Tx5_mode | | |
| | | 00 | Transparent Mode on TX endpoint 5 |
| | | 01 | RNDIS MODE on TX endpoint 5 |
| | | 10 | CDC Mode on TX endpoint 5 |
| | | 11 | Generic RNDIS Mode on TX endpoint 5 |
| 7-6 | Tx4_mode | | |
| | | 00 | Transparent Mode on TX endpoint 4 |
| | | 01 | RNDIS MODE on TX endpoint 4 |
| | | 10 | CDC Mode on TX endpoint 4 |
| | | 11 | Generic RNDIS Mode on TX endpoint 4 |
| 5-4 | Tx3_mode | | |
| | | 00 | Transparent Mode on TX endpoint 3 |
| | | 01 | RNDIS MODE on TX endpoint 3 |
| | | 10 | CDC Mode on TX endpoint 3 |
| | | 11 | Generic RNDIS Mode on TX endpoint 3 |
| 3-2 | Tx2_mode | | |
| | | 00 | Transparent Mode on TX endpoint 2 |
| | | 01 | RNDIS MODE on TX endpoint 2 |
| | | 10 | CDC Mode on TX endpoint 2 |
| | | 11 | Generic RNDIS Mode on TX endpoint 2 |
| 1-0 | Tx1_mode | | |
| | | 00 | Transparent Mode on TX endpoint 1 |
| | | 01 | RNDIS MODE on TX endpoint 1 |
| | | 10 | CDC Mode on TX endpoint 1 |
| | | 11 | Generic RNDIS Mode on TX endpoint 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.2.2.15 USB1 Rx Mode Register (USB1RXMODE)

The USB1 Rx mode register (USB1RXMODE) allows the CPU to individually enable RNDIS/Generic/CDC modes for each endpoint. Using the global RNDIS enable in the control register overrides this register and enable RNDIS mode for all endpoints.

The USB1 Rx mode register is shown in Figure 25-103 and described in Table 25-113.

**Figure 25-103. USB1 Rx Mode Register (USB1RXMODE)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | Rx15_mode | | Rx14_mode | | Rx13_mode | | Rx12_mode | | Rx11_mode | | Rx10_mode | | Rx9_mode | |
| R-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rx8_mode | | Rx7_mode | | Rx6_mode | | Rx5_mode | | Rx4_mode | | Rx3_mode | | Rx2_mode | | Rx1_mode | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-113. USB1 Rx Mode Register (USB1RXMODE) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Always read as 0. Writes have no effect. |
| 29-28 | Rx15_mode | | |
| | | 00 | Transparent Mode on RX endpoint 15 |
| | | 01 | RNDIS MODE on RX endpoint 15 |
| | | 10 | CDC Mode on RX endpoint 15 |
| | | 11 | Generic RNDIS Mode on RX endpoint 15 |
| 27-26 | Rx14_mode | | |
| | | 00 | Transparent Mode on RX endpoint 14 |
| | | 01 | RNDIS MODE on RX endpoint 14 |
| | | 10 | CDC Mode on RX endpoint 14 |
| | | 11 | Generic RNDIS Mode on RX endpoint 14 |
| 25-24 | Rx13_mode | | |
| | | 00 | Transparent Mode on RX endpoint 13 |
| | | 01 | RNDIS MODE on RX endpoint 13 |
| | | 10 | CDC Mode on RX endpoint 13 |
| | | 11 | Generic RNDIS Mode on RX endpoint 13 |
| 23-22 | Rx12_mode | | |
| | | 00 | Transparent Mode on RX endpoint 12 |
| | | 01 | RNDIS MODE on RX endpoint 12 |
| | | 10 | CDC Mode on RX endpoint 12 |
| | | 11 | Generic RNDIS Mode on RX endpoint 12 |
| 21-20 | Rx11_mode | | |
| | | 00 | Transparent Mode on RX endpoint 11 |
| | | 01 | RNDIS MODE on RX endpoint 11 |
| | | 10 | CDC Mode on RX endpoint 11 |
| | | 11 | Generic RNDIS Mode on RX endpoint 11 |
| 19-18 | Rx10_mode | | |
| | | 00 | Transparent Mode on RX endpoint 10 |
| | | 01 | RNDIS MODE on RX endpoint 10 |
| | | 10 | CDC Mode on RX endpoint 10 |
| | | 11 | Generic RNDIS Mode on RX endpoint 10 |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-113. USB1 Rx Mode Register (USB1RXMODE) Field Descriptions (continued)**

| Bits | Field | Value | Description |
|---|---|---|---|
| 17-16 | Rx9_mode | | |
| | | 00 | Transparent Mode on RX endpoint 9 |
| | | 01 | RNDIS MODE on RX endpoint 9 |
| | | 10 | CDC Mode on RX endpoint 9 |
| | | 11 | Generic RNDIS Mode on RX endpoint 9 |
| 15-14 | Rx8_mode | | |
| | | 00 | Transparent Mode on RX endpoint 8 |
| | | 01 | RNDIS MODE on RX endpoint 8 |
| | | 10 | CDC Mode on RX endpoint 8 |
| | | 11 | Generic RNDIS Mode on RX endpoint 8 |
| 13-12 | Rx7_mode | | |
| | | 00 | Transparent Mode on RX endpoint 7 |
| | | 01 | RNDIS MODE on RX endpoint 7 |
| | | 10 | CDC Mode on RX endpoint 7 |
| | | 11 | Generic RNDIS Mode on RX endpoint 7 |
| 11-10 | Rx6_mode | | |
| | | 00 | Transparent Mode on RX endpoint 6 |
| | | 01 | RNDIS MODE on RX endpoint 6 |
| | | 10 | CDC Mode on RX endpoint 6 |
| | | 11 | Generic RNDIS Mode on RX endpoint 6 |
| 9-8 | Rx5_mode | | |
| | | 00 | Transparent Mode on RX endpoint 5 |
| | | 01 | RNDIS MODE on RX endpoint 5 |
| | | 10 | CDC Mode on RX endpoint 5 |
| | | 11 | Generic RNDIS Mode on RX endpoint 5 |
| 7-6 | Rx4_mode | | |
| | | 00 | Transparent Mode on RX endpoint 4 |
| | | 01 | RNDIS MODE on RX endpoint 4 |
| | | 10 | CDC Mode on RX endpoint 4 |
| | | 11 | Generic RNDIS Mode on RX endpoint 4 |
| 5-4 | Rx3_mode | | |
| | | 00 | Transparent Mode on RX endpoint 3 |
| | | 01 | RNDIS MODE on RX endpoint 3 |
| | | 10 | CDC Mode on RX endpoint 3 |
| | | 11 | Generic RNDIS Mode on RX endpoint 3 |
| 3-2 | Rx2_mode | | |
| | | 00 | Transparent Mode on RX endpoint 2 |
| | | 01 | RNDIS MODE on RX endpoint 2 |
| | | 10 | CDC Mode on RX endpoint 2 |
| | | 11 | Generic RNDIS Mode on RX endpoint 2 |
| 1-0 | Rx1_mode | | |
| | | 00 | Transparent Mode on RX endpoint 1 |
| | | 01 | RNDIS MODE on RX endpoint 1 |
| | | 10 | CDC Mode on RX endpoint 1 |
| | | 11 | Generic RNDIS Mode on RX endpoint 1 |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.16 USB1 Generic RNDIS EP N Size Register (USB1GENRNDISEPn)

The USB1 generic RNDIS EP N size register (USB1GENRNDISEPn) is programmed with a RNDIS packet size in bytes. When EP(N) is in Generic RNDIS mode, the received USB packets will be collected into a single CPPI packet that is completed when the number of bytes equal to the value of this register has been received, or a “short” packet is received. Note that this register must be programmed with a value that is an integer multiple of the endpoint size. The maximum value this register can be programmed with is 10000h, or 65536. N can range from 1 to 15.

The USB1 generic RNDIS EP N size register is shown in Figure 25-104 and described in USB1 Generic RNDIS EP N Size Register (USB1GENRNDISEPn) Field Descriptions.

**Figure 25-104. USB1 Generic RNDIS EP N Size Register (USB1GENRNDISEPn)**

| 31 | 17 | 16 | 0 |
|---|---|---|---|
| Reserved | | Ep(N)_size | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**USB1 Generic RNDIS EP N Size Register (USB1GENRNDISEPn) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-17 | Reserved | Always read as 0. Writes have no effect. |
| 16-0 | Ep(N)_size | Generic RNDIS packet size. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.17 USB1 Auto Req Register (USB1AUTOREQ)

The USB1 auto req register (USB1AUTOREQ) allows the CPU to enable an automatic IN token request generation for host mode RX operation per each RX endpoint. This features has the DMA set the ReqPkt bit in the RXCSR when it clears the RxPktRdy bit after reading out a packet. The ReqPkt bit is used by the core to generate an IN token to receive data. By using this feature, the host can automatically generate an IN token after the DMA finishes receiving data and empties an endpoint buffer, thus receiving the next data packet as soon as possible from the connected device. Without this feature, the CPU will have to manually set the ReqPkt bit for every USB packet.

There are two modes that Auto Req can function: always or all except an EOP. The always mode will set the ReqPkt bit after every USB packet the DMA receives thus generating a new IN token after each USB packet. The EOP mode will set the ReqPkt bit after every USB packet that isn't an EOP (end of packet) in the CPPI descriptor. For RNDIS, CDC, and Generic RNDIS modes the auto req will stop when the EOP is received (either via a short packet for RNDIS,CDC, and Generic RNDIS or the count is reached for Generic RNDIS), making it useful for starting a large RNDIS packet and having it auto generate IN tokens until the end of the RNDIS packet. For transparent mode, every USB packet is an EOP CPPI packet, so the auto req never functions and acts like auto req is disabled.

The USB1 auto req register is shown in Figure 25-105 and described in Table 25-114.

**Figure 25-105. USB1 Auto Req Register (USB1AUTOREQ)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | Rx15_autoreq | | Rx14_autoreq | | Rx13_autoreq | | Rx12_autoreq | | Rx11_autoreq | | Rx10_autoreq | | Rx9_autoreq | |
| R-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rx8_autoreq | | Rx7_autoreq | | Rx6_autoreq | | Rx5_autoreq | | Rx4_autoreq | | Rx3_autoreq | | Rx2_autoreq | | Rx1_autoreq | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-114. USB1 Auto Req Register (USB1AUTOREQ) Field Descriptions**

| Bits | Field | Values | Description |
|---|---|---|---|
| 31-30 | Reserved | 0 | Always read as 0. Writes have no effect. |
| 29-28 | Rx15_autoreq | | RX endpoint 15 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 27-26 | Rx14_autoreq | | RX endpoint 14 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 25-24 | Rx13_autoreq | | RX endpoint 13 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 23-22 | Rx12_autoreq | | RX endpoint 12 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-114. USB1 Auto Req Register (USB1AUTOREQ) Field Descriptions (continued)**

| Bits | Field | Values | Description |
|---|---|---|---|
| 21-20 | Rx11_autoreq | | RX endpoint 11 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 19-18 | Rx10_autoreq | | RX endpoint 10 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 17-16 | Rx9_autoreq | | RX endpoint 9 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 15-14 | Rx8_autoreq | | RX endpoint 8 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 13-12 | Rx7_autoreq | | RX endpoint 7 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 11-10 | Rx6_autoreq | | RX endpoint 6 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 9-8 | Rx5_autoreq | | RX endpoint 5 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 7-6 | Rx4_autoreq | | RX endpoint 4 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 5-4 | Rx3_autoreq | | RX endpoint 3 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-114. USB1 Auto Req Register (USB1AUTOREQ) Field Descriptions (continued)**

| Bits | Field | Values | Description |
|---|---|---|---|
| 3-2 | Rx2_autoreq | | RX endpoint 2 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |
| 1-0 | Rx1_autoreq | | RX endpoint 1 Auto Req enable |
| | | 00 | No auto req |
| | | 01 | Auto req on all but EOP |
| | | 10 | Reserved |
| | | 11 | Auto req always |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.18 USB1 SRP Fix Time Register (USB1SRPFIXTIME)

The USB1 SRP fix time register (USB1SRPFIXTIME) allows the CPU to configure the maximum amount of time the SRP fix logic blocks the AVAID from the PHY to the OTG core. This time allows the VBUS signal the ability to get below the thresholds and therefore remove the chance of voltage bounces which could give false threshold measurements.

The USB1 SRP fix time register is shown in Figure 25-106 and described in Table 25-115.

**Figure 25-106. USB1 SRP Fix Time Register (USB1SRPFIXTIME)**

| 31 | 0 |
|---|---|
| srpfixtime | |
| R/W-0280de80h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-115. USB1 SRP Fix Time Register (USB1SRPFIXTIME) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-0 | srpfixtime | SRP Fix maximum time in 60 MHz cycles. Default is 700 ms. |

#### 25.9.2.2.19 USB1 Teardown Register (USB1TDOWN)

The USB1 teardown register (USB1TDOWN) controls the tearing down of rx and tx fifos in the USB controller. When a '1' is written to a valid bit in this register, the CPPI FIFO pointers for that endpoint are cleared , and the register automatically clears itself after 1 clock cycle. This register must be used in conjunction with the CPPI DMA Teardown mechanism. The Host should also write the FlushFIFO bits in the TXCSR and RXCSR Mentor USB Controller registers to ensure a complete teardown of the endpoint. See the Mentor specification for details.

The USB1 teardown register is shown in Figure 25-107 and described in Table 25-116.

**Figure 25-107. USB1 Teardown Register (USB1TDOWN)**

| 31 | 17 | 16 | 15 | 0 |
|---|---|---|---|---|
| tx_tdown | | Rsvd | rx_tdown | Rsvd |
| R/W-0h | | R-0h | R/W-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-116. USB1 Teardown Register (USB1TDOWN) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-17 | tx_tdown | Tx Endpoint Teardown. Write '1' to corresponding bit to set. Read as '0'.<br>Bit 31 = Endpoint 15<br>...<br>Bit 17 = Endpoint 1 |
| 16 | Reserved | Always read as 0. Writes have no effect. |
| 15-1 | rx_tdown | RX Endpoint Teardown. Write '1' to corresponding bit to set. Read as '0'.<br>Bit 15 = Endpoint 15<br>...<br>Bit 1 = Endpoint 1 |
| 0 | Reserved | Always read as 0. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.20 USB1 Threshold XDMA Idle Register

The USB1 Threshold XDMA Idle register provides a cycle threshold number that is programmable. This cycle threshold and cycle count is used by the XDMA state machine when the EP_STATE machine is in the IDLE mode. When the threshold is reached, the XDMA leaves the G_IDLE state and reads the TXCSR registers within the Mentor controller and possibly generates an interrupt for TX_FIFO[15:0].

See the XDMA section in the Integration Specification for more details about the various state machines.

The USB1 threshold XDMA idle register is shown in Figure 25-108 and described in Table 25-117.

**Figure 25-108. USB1 Threshold XDMA Idle Register**

| 31 | 8 | 7 | 0 |
|---|---|---|---|
| Reserved | | thres_xdma_idle | |
| R-0h | | R/W-18h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-117. USB1 Threshold XDMA Idle Register Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-8 | Reserved | Always read as 0. Writes have no effect. |
| 7-0 | thres_xdma_idle | Threshold XDMA Idle cycle parameter. Valid values are 0 to 255. Default is 24 cycles. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.21 USB1 PHY UTMI Register (USB1UTMI)

The USB1 PHY UTMI register (USB1UTMI) controls various PHY UTMI signals. The UTMI interface is located between the PHY module and the Mentor Graphics Controller. These signals are inputs to the PHY but are not outputs from the Mentor Graphics Controller.

The USB1 PHY UTMI register is shown in Figure 25-109 and described in Table 25-118.

**Figure 25-109. USB1 PHY UTMI Register (USB1UTMI)**

| 31 | | | | | | | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| txbitstuffen | Reserved | Reserved | Reserved | Reserved | txenablen | Reserved | Reserved |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | | | | | | | 8 |
| Reserved | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | | | | 3 | 2 | 1 | 0 |
| Reserved | | | | | Reserved | Reserved | Reserved |
| R-0h | | | | | R/W-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-118. USB1 PHY UTMI Register (USB1UTMI) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-24 | Reserved | Always read as 0. Writes have no effect. |
| 23 | txbitstuffen | PHY UTMI input for signal txbitstuffen |
| 22 | Reserved | Reserved, this bit has no function. |
| 21 | Reserved | Reserved, this bit has no function. |
| 20 | Reserved | Reserved, this bit has no function. |
| 19 | Reserved | Reserved, this bit has no function. |
| 18 | txenablen | PHY UTMI input for signal txenablen |
| 17 | Reserved | Reserved, this bit has no function. |
| 16 | Reserved | Reserved, this bit has no function. |
| 15-8 | Reserved | Reserved, this bit has no function. |
| 7-3 | Reserved | Reserved, this bit has no function. |
| 2 | Reserved | Reserved, this bit has no function. |
| 1 | Reserved | Reserved, this bit has no function. |
| 0 | Reserved | Reserved, this bit has no function. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.2.2.22 USB1 MGC UTMI Loopback Register (USB1UTMILB)

The USB1 MGC UTMI loopback register (USB1UTMILB) contains most of the input and output UTMI signals for the Mentor Graphics Controller. The UTMI interface is located between the PHY module and the Mentor Graphics Controller.

In the loopback mode test register bits 11 to 0 control various UTMI signals that are input to the Mentor Graphics Controller.

In the loopback mode test register bits 28 to 16 observe various UTMI signals that are output from the Mentor Graphics Controller.

The USB1 MGC UTMI loopback register is shown in Figure 25-110 and described in Table 25-119.

**Figure 25-110. USB1 MGC UTMI Loopback Register (USB1UTMILB)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | suspendm | opmode | | txvalid | xcvrsel |
| R-0h | | | R-0h | R-0h | | R-0h | R-0h |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| xcvrsel | termsel | drvvbus | chrgvbus | dischrgvbus | dppulldown | dmpulldown | idpullup |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| Reserved | | | | iddig | hostdiscon | sessend | avalid |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| vbusvalid | rxerror | Reserved | | linestate | | Reserved | |
| R/W-0h | R/W-0h | R-0h | | R/W-0h | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-119. USB1 MGC UTMI Loopback Register (USB1UTMILB) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-29 | Reserved | Always read as 0. Writes have no effect. |
| 28 | suspendm | Loopback test observed value for suspendm |
| 27-26 | opmode | Loopback test observed value for opmode |
| 25 | txvalid | Loopback test observed value for txvalid |
| 24-23 | xcvrsel | Loopback test observed value for xcvrsel |
| 22 | termsel | Loopback test observed value for termsel |
| 21 | drvvbus | Loopback test observed value for drvvbus |
| 20 | chrgvbus | Loopback test observed value for chrgvbus |
| 19 | dischrgvbus | Loopback test observed value for dischrgvbus |
| 18 | dppulldown | Loopback test observed value for dppulldown |
| 17 | dmpulldown | Loopback test observed value for dmpulldown |
| 16 | idpullup | Loopback test observed value for idpullup |
| 15-12 | Reserved | Always read as 0. Writes have no effect. |
| 11 | iddig | Loopback test value for iddig |
| 10 | hostdiscon | Loopback test value for hostdiscon |
| 9 | sessend | Loopback test value for sessend |
| 8 | avalid | Loopback test value for avalid |
| 7 | vbusvalid | Loopback test value for vbusvalid |
| 6 | rxerror | Loopback test value for rxerror |
| 5-4 | Reserved | Always read as 0. Writes have no effect. |
| 3-2 | linestate | Loopback test value for linestate |
| 1-0 | Reserved | Always read as 0. Writes have no effect. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.2.2.23 USB1 Mode Register (USB1MODE)

The USB1 mode register (USB1MODE) allows the user a flexibility of controlling the role that the controller assumes. Via the setting of the USB1MODE, the user has the capability of allowing the control to be coming from a register field (USB1MODE[Bit8=iddig]) or from the USB_ID pin which is indirectly controlled by the cable end. if USB1MODE=0, the state of the USB_ID pin controls the role the controller assumes.

The loopback bit enables the loopback test. This test allows MGC1 to be connected to MGC0. It is important to set both loopback bits in both USB0/1 Mode registers. The USB0 MGC UTMI Loopback Register contains the various UTMI signals to be controlled and observed during loopback test.

The phy_test bit enables the phy_test mode. This test mode is intended to allow additional control of the UTMI inputs to the PHY. Currently, these inputs are drvvbus, dppulldown, dmpulldown, and idpullup. When phy_test is high, then the pin inputs for these signals control the inputs to the PHY instead of the Mentor controller outputs. The phy_test mode is not active with loopback mode is active.

When phy_test is active then the PHY inputs datainh is equal to datain and txvalidh is equal to txvalid.

The USB1 mode register is shown in Figure 25-111 and described in Table 25-120.

**Figure 25-111. USB1 Mode Register (USB1MODE)**

| 31 | 16 |
|---|---|
| Reserved | |
| R-0h | |

| 15 | 9 | 8 | 7 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Reserved | | iddig | Reserved | | phy_test | loopback |
| R-0h | | R/W-1h | R-0h | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-120. USB1 Mode Register (USB1MODE) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-9 | Reserved | | Always read as 0. Writes have no effect. |
| 8 | iddig | | MGC input value for iddig |
| | | 0 | A type |
| | | 1 | B type |
| 7-2 | Reserved | | Always read as 0. Writes have no effect. |
| 1 | phy_test | | PHY test |
| | | 0 | Normal mode |
| | | 1 | PHY test mode |
| 0 | loopback | | Loopback test mode |
| | | 0 | Normal mode |
| | | 1 | Loopback test mode |

#### 25.9.2.2.24 USB1 Mentor Core Registers

A description of the mentor core registers is available in Section 25.9.6. Two instantiations of the USB core exists, USB0 and USB1. These registers reside back to back within USB0 space. The core registers that are used by USB0 subsystem are defined within offsets 1400h-159Ch, while the core registers that are used by USB1 subsystem are defined within offsets 1C00h–1D9Ch.


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.3 CPPI DMA Controller Registers

Table 25-121 lists the registers for the CPPI DMA Controller submodule.

**Table 25-121. CPPI DMA Controller Registers**

| Address Offset | Acronym | CPPI DMA Controller Register |
|---|---|---|
| 2000h | DMAREVID | Revision Register |
| 2004h | TDFDQ | Teardown Free Descriptor Queue Control Register |
| 2008h | DMAEMU | Emulation Control Register |
| 2800h | TXGCR0 | Tx Channel 0 Global Configuration Register |
| 2808h | RXGCR0 | Rx Channel 0 Global Configuration Register |
| 280Ch | RXHPCRA0 | Rx Channel 0 Host Packet Configuration Register A |
| 2810h | RXHPCRB0 | Rx Channel 0 Host Packet Configuration Register B |
| 2820h | TXGCR1 | Tx Channel 1 Global Configuration Register |
| 2828h | RXGCR1 | Rx Channel 1 Global Configuration Register |
| 282Ch | RXHPCRA1 | Rx Channel 1 Host Packet Configuration Register A |
| 2830h | RXHPCRB1 | Rx Channel 1 Host Packet Configuration Register B |
| 2840h | TXGCR2 | Tx Channel 2 Global Configuration Register |
| 2848h | RXGCR2 | Rx Channel 2 Global Configuration Register |
| 284Ch | RXHPCRA2 | Rx Channel 2 Host Packet Configuration Register A |
| 2850h | RXHPCRB2 | Rx Channel 2 Host Packet Configuration Register B |
| 2860h | TXGCR3 | Tx Channel 3 Global Configuration Register |
| 2868h | RXGCR3 | Rx Channel 3 Global Configuration Register |
| 286Ch | RXHPCRA3 | Rx Channel 3 Host Packet Configuration Register A |
| 2870h | RXHPCRB3 | Rx Channel 3 Host Packet Configuration Register B |
| 2880h-2B9Fh | ... | ... |
| 2BA0h | TXGCR29 | Tx Channel 29 Global Configuration Register |
| 2BA8h | RXGCR29 | Rx Channel 29 Global Configuration Register |
| 2BACh | RXHPCRA29 | Rx Channel 29 Host Packet Configuration Register A |
| 2BB0h | RXHPCRB29 | Rx Channel 29 Host Packet Configuration Register B |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.3.1 CPPI DMA Revision Register (DMAREVID)

The CPPI DMA revision register (DMAREVID) contains the major and minor revisions for the module. The CPPI DMA revision register is shown in Figure 25-112 and described in Table 25-122.

**Figure 25-112. CPPI DMA Revision Register (DMAREVID)**

| 31 30 | 29 16 | 15 11 | 10 8 | 7 0 |
|---|---|---|---|---|
| Rsvd | modID | revrtl | revmaj | revmin |
| R-0h | R-53h | R-1h | R-1h | R-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-122. CPPI DMA Revision Registe (DMAREVID) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-30 | Reserved | Always read as 0. Writes have no effect. |
| 29-16 | modID | Module ID field |
| 15-11 | revrtl | RTL revision. Will vary depending on release. |
| 10-8 | revmaj | Major revision. |
| 7-0 | revmin | Minor revision. |

### 25.9.3.2 CPPI DMA Teardown Free Descriptor Queue Control Register (TDFDQ)

The CPPI DMA teardown free descriptor queue control register (TDFDQ) is used to inform the DMA of the location in memory or descriptor array which is to be used for signaling of a teardown complete for each Tx and Rx channel. The CPPI DMA teardown free descriptor queue control register is shown in Figure 25-113 and described in Table 25-123.

**Figure 25-113. CPPI DMA Teardown Free Descriptor Queue Control Register (TDFDQ)**

| 31 14 | 13 12 | 11 0 |
|---|---|---|
| Reserved | td_desc_qmgr | td_desc_qnum |
| R-0h | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-123. CPPI DMA Teardown Free Descriptor Queue Control Register (TDFDQ) Field Descriptions**

| Bits | Field | Description |
|---|---|---|
| 31-14 | Reserved | Reserved |
| 13-12 | td_desc_qmgr | This field controls which of the four Queue Managers the DMA will access in order to allocate a channel teardown descriptor from the teardown descriptor queue. |
| 11-0 | td_desc_qnum | This field controls which of the 2K queues in the indicated queue manager should be read in order to allocate channel teardown descriptors. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.3.3 CPPI DMA Emulation Control Register (DMAEMU)

The CPPI DMA emulation control register (DMAEMU) is used to control the behavior of the DMA when the emususp input is asserted. The CPPI DMA emulation control register is shown in Figure 25-114 and described in Table 25-124.

**Figure 25-114. CPPI DMA Emulation Control Register (DMAEMU)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | soft | free |
| R-0h | | R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-124. CPPI DMA Emulation Control Register (DMAEMU) Field Descriptions**

| Bits | Field | Value | Description |
|---|---|---|---|
| 31-2 | Reserved | 0 | Always read as 0. Writes have no effect. |
| 1 | soft | | Control for emulation pause request |
| | | 0 | Forces emu_pause_req low |
| | | 1 | Does not force emu_pause_req low |
| 0 | free | | Enable for emulation suspend |
| | | 0 | Emulation halt disabled |
| | | 1 | Emulation halt enabled |

### 25.9.3.4 Tx Channel N Global Configuration Register (TXGCRn)

The Tx channel N global configuration register (TXGCRn) are used to initialize the behavior of each of the Tx CPPI DMA channels. N,n ranges from 0 to 14. The Tx channel N global configuration register is shown in Figure 25-115 and described in Table 25-125.

**Figure 25-115. Tx Channel N Global Configuration Register (TXGCRn)**

| 31 | 30 | 29 | 28 | 16 |
|---|---|---|---|---|
| tx_enable | tx_teardown | tx_pause | Reserved | |
| R/W-0h | R/W-0h | R/W-0h | R-0h | |

| 15 | 14 | 13 | 12 | 11 | 0 |
|---|---|---|---|---|---|
| Reserved | | tx_default_qmgr | | tx_default_qnum | |
| R-0h | | W-0h | | W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

**Table 25-125. Tx Channel N Global Configuration Register (TXGCRn) Field Descriptions**

| Bits | Field Name | | Description |
|---|---|---|---|
| 31 | tx_enable | | This field enables or disables the channel |
| | | 0 | Channel is disabled |
| | | 1 | Channel is enabled |
| | | | This field will be cleared after a channel teardown is complete. |
| 30 | tx_teardown | 0 - 1 | Setting this bit will request the channel to be torn down. This field will remain set after a channel teardown is complete. |
| 29 | tx_pause | 0 - 1 | Setting this bit causes the CPPI DMA to be suspended for tx channels. If a pause is being requested and the channel is not in a packet then drop the credit. |
| 28-14 | Reserved | 0 | Reserved |
| 13-12 | tx_default_qmgr | 0 - 3 | This field controls the default queue manager number that will be used to queue teardown descriptors back to the host. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-125. Tx Channel N Global Configuration Register (TXGCRn) Field Descriptions (continued)**

| Bits | Field Name | | Description |
|---|---|---|---|
| 11-0 | tx_default_qnum | 0 - FFFh | This field controls the default queue number within the selected queue manager onto which teardown descriptors will be queued back to the host. |

### 25.9.3.5 Rx Channel N Global Configuration Register (RXGCRn)

The Rx channel global configuration registers are used to initialize the global (non descriptor type specific) behavior of each of the Rx CPPI DMA channels. If the enable bit is being set, the Rx Channel Global Configuration Register should only be written after all of the other Rx Configuration Registers have been initialized. N,n ranges from 0 to 14.

The Rx channel N global configuration register is shown in Figure 25-116 and described in Table 25-126

**Figure 25-116. Rx Channel N Global Configuration Register (RXGCRn)**

| 31 | 30 | 29 | 28 26 | 25 | 24 |
|---|---|---|---|---|---|
| rx_enable | rx_teardown | rx_pause | rx_max_buffer_cnt | Reserved | rx_error_handling |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | W-0h |

| 23 16 |
|---|
| rx_sop_offset |
| W-0h |

| 15 14 | 13 12 | 11 8 |
|---|---|---|
| rx_default_desc_type | rx_default_rq_qmgr | rx_default_rq_qnum |
| W-0h | W-0h | W-0h |

| 7 0 |
|---|
| rx_default_rq_qnum |
| W-0h |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

**Table 25-126. Rx Channel N Global Configuration Register (RXGCRn) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31 | rx_enable | | This field enables or disables the channel |
| | | 0 | Channel is disabled |
| | | 1 | Channel is enabled |
| | | | This field will be cleared after a channel teardown is complete. |
| 30 | rx_teardown | 0 - 1 | This field indicates whether or not an Rx teardown operation is complete. This field should be cleared when a channel is initialized. This field will be set after a channel teardown is complete. |
| 29 | rx_pause | 0 - 1 | Setting this bit causes the CPPI DMA to be suspended for rx channels. If a pause is being requested and the channel is not in a packet then drop the credit. |
| 28-26 | rx_max_buffer_cnt | | This bit is used to control an optional Rx mode of operation where packets are delimited based on a an integer number of buffers in addition to the EOP bit in the CPPI FIFO information word 0. |
| | | 0 | Mode is disabled – the DMA will delimit Rx packets only using the SOP and EOP fields in the info word 0. |
| | | 1h | The DMA ignores the SOP bit and closes up a packet after a single buffer has been filled OR if the EOP field is set in the info word 0. |
| | | 2h | The DMA ignores the SOP bit and closes up a packet after two buffers have been filled OR if the EOP field is set in the info word 0. |
| | | 3h | The DMA ignores the SOP bit and closes up a packet after three buffers have been filled OR if the EOP field is set in the info word 0. |
| | | 4h | The DMA ignores the SOP bit and closes up a packet after four buffers have been filled OR if the EOP field is set in the info word 0. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-126. Rx Channel N Global Configuration Register (RXGCRn) Field Descriptions (continued)**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| | | 5h-7h | Reserved |
| 25 | Reserved | 0 | Reserved |
| 24 | rx_error_handling | | This bit controls the error handling mode for the channel and is only used when channel errors (i.e. descriptor or buffer starvation occurs): |
| | | 0 | Starvation errors result in dropping packet and reclaiming any used descriptor or buffer resources back to the original queues/pools they were allocated to |
| | | 1 | Starvation errors result in subsequent re-try of the descriptor allocation operation. In this mode, the DMA will return to the IDLE state without saving it's internal operational state back to the internal state RAM and without issuing an advance operation on the FIFO interface. This results in the DMA re-initiating the FIFO block transfer at a later time with the intention that additional free buffers and/or descriptors will have been added. |
| | | | Regardless of the value of this bit, the DMA will assert the cdma_rx_sof_overrun (for SOP) or cdma_rx_mof_overrun (for non-SOP) when |
| 23-16 | rx_sop_offset | 0 - FFh | This field specifies the number of bytes that are to be skipped in the SOP buffer before beginning to write the payload. This value must be less than the minimum size of a buffer in the system. Valid values are 0 – 255 bytes. |
| 15-14 | rx_default_desc_type | | This field indicates the default descriptor type to use: |
| | | 0 | Reserved |
| | | 1h | Host |
| | | 2h-3h | Reserved |
| | | | The actual descriptor type that will be used for reception can be overridden by information provided in the CPPI FIFO data block. |
| 13-12 | rx_default_rq_qmgr | 0 - 3h | This field indicates the default receive queue manager that this channel should use. The actual receive queue manager index can be overridden by information provided in the CPPI FIFO data block. |
| 11-0 | rx_default_rq_qnum | 0 - FFFh | This field indicates the default receive queue that this channel should use. The actual receive queue that will be used for reception can be overridden by information provided in the CPPI FIFO data block. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.3.6 Rx Channel N Host Packet Configuration Register A (RXHPCRAn)

The Rx channel host packet configuration A registers are used to initialize the behavior of each of the Rx CPPI DMA channels for reception of host type packets. N,n ranges from 0 to 14.

The Rx channel host packet configuration A registers are shown in Figure 25-117 and described in Table 25-127

**Figure 25-117. Rx Channel N Host Packet Configuration Register A (RXHPCRAn)**

| 31 30 | 29 28 | 27 16 |
|---|---|---|
| Reserved | rx_host_fdq1_qmgr | rx_host_fdq1_qnum |
| R-0h | W-0h | W-0h |

| 15 14 | 13 12 | 11 0 |
|---|---|---|
| Reserved | rx_host_fdq0_qmgr | rx_host_fdq0_qnum |
| R-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

**Table 25-127. Rx Channel N Host Packet Configuration Register A (RXHPCRAn) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-30 | Reserved | Reserved |
| 29-28 | rx_host_fdq1_qmgr | This field specifies which Buffer Manager should be used for the second Rx buffer in a host type packet. |
| 27-16 | rx_host_fdq1_qnum | This field specifies which Free Descriptor / Buffer Pool should be used for the second Rx buffer in a host type packet. |
| 15-14 | Reserved | Reserved |
| 13-12 | rx_host_fdq0_qmgr | This field specifies which Buffer Manager should be used for the first Rx buffer in a host type packet. |
| 11-0 | rx_host_fdq0_qnum | This field specifies which Free Descriptor / Buffer Pool should be used for the frst Rx buffer in a host type packet. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.3.7 Rx Channel N Host Packet Configuration Register B (RXHPCRBn)

The Rx channel host packet configuration B registers are used to initialize the behavior of each of the Rx CPPI DMA channels for reception of host type packets. N,n ranges from 0 to 14.

The Rx channel host packet configuration B registers are shown in Figure 25-118 and described in Table 25-128

**Figure 25-118. Rx Channel N Host Packet Configuration Register B (RXHPCRBn)**

| 31 30 | 29 28 | 27 16 |
|---|---|---|
| Reserved | rx_host_fdq3_qmgr | rx_host_fdq3_qnum |
| R-0h | W-0h | W-0h |

| 15 14 | 13 12 | 11 0 |
|---|---|---|
| Reserved | rx_host_fdq2_qmgr | rx_host_fdq2_qnum |
| R-0h | W-0h | W-0h |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

**Table 25-128. Rx Channel N Host Packet Configuration Register B (RXHPCRBn) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-30 | Reserved | Reserved |
| 29-28 | rx_host_fdq3_qmgr | This field specifies which Manager should be used for the fourth or later Rx buffers in a host type packet. |
| 27-16 | rx_host_fdq3_qnum | This field specifies which Free Descriptor Queue should be used for the fourth or later Rx buffers in a host type packet. |
| 15-14 | Reserved | Reserved |
| 13-12 | rx_host_fdq2_qmgr | This field specifies which Buffer Manager should be used for the third Rx buffer in a host type packet. |
| 11-0 | rx_host_fdq2_qnum | This field specifies which Free Descriptor / Buffer Pool should be used for the third Rx buffer in a host type packet. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.4 CPPI DMA Scheduler Registers

Table 25-129 lists the registers for the CPPI DMA Scheduler submodule.

**Table 25-129. CPPI DMA Scheduler Registers**

| Address Offset | Acronym | CPPI DMA Scheduler Register |
|---|---|---|
| 3000h | DMA_SCHED_CTRL | CPPI DMA Scheduler Control Register |
| 3800h-38FCh | WORD0-WORD63 | CPPI DMA Scheduler Table Word 0-Word 63 |

#### 25.9.4.1 CPPI DMA Scheduler Control Register (DMA_SCHED_CTRL)

The CPPI DMA scheduler control revision register contains the major and minor revisions for the module. The revision register is shown in Figure 25-119 and described in Table 25-130.

**Figure 25-119. CPPI DMA Scheduler Control Register (DMA_SCHED_CTRL)**

| 31 | 30 8 | 7 0 |
|---|---|---|
| enable | Reserved | last_entry |
| R/W-0h | R-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-130. CPPI DMA Scheduler Control Register (DMA_SCHED_CTRL) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31 | enable | | This is the enable bit for the scheduler and is encoded as follows: |
| | | 0 | Scheduler is disabled and will no longer fetch entries from the scheduler table or pass credits to the DMA controller. |
| | | 1 | Scheduler is enabled. |
| | | | This bit should only be set after the table has been initialized. |
| 30-8 | Reserved | 0 | Reserved |
| 7-0 | last_entry | | This field indicates the last valid entry in the scheduler table. There are 64 words in the table and there are 4 entries in each word. The table can be programmed with any integer number of entries from 1 to 256. The corresponding encoding for this field is as follows: |
| | | 0 | 1 entry |
| | | 1 | 2 entries |
| | | ... | … |
| | | 254 | 255 entries |
| | | 255 | 256 entries |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.4.2 CPPI DMA Scheduler Table Word N (WORDn)

The CPPI DMA scheduler table word N (WORDn) registers are used to initialize the behavior of each of the Tx DMA channels. N ranges from 0 to 63.

The CPPI DMA scheduler table word N (WORDn) registers are shown in Figure 25-120 and described in Table 25-131.

**Figure 25-120. CPPI DMA Scheduler Table Word N (WORDn)**

| 31 | 30 | 29 | 28 | 24 |
|---|---|---|---|---|
| entry3_rxtx | Reserved | | entry3_channel | |
| W-0h | R-0h | | W-0h | |

| 23 | 22 | 21 | 20 | 16 |
|---|---|---|---|---|
| entry2_rxtx | Reserved | | entry2_channel | |
| W-0h | R-0h | | W-0h | |

| 15 | 14 | 13 | 12 | 8 |
|---|---|---|---|---|
| entry1_rxtx | Reserved | | entry1_channel | |
| W-0h | R-0h | | W-0h | |

| 7 | 6 | 5 | 4 | 0 |
|---|---|---|---|---|
| entry0_rxtx | Reserved | | entry0_channel | |
| W-0h | R-0h | | W-0h | |

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

**Table 25-131. CPPI DMA Scheduler Table Word N (WORDn) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 31 | entry3_rxtx | | This bit indicates if this entry is for a Tx or an Rx channel and is encoded as follows: |
| | | 0 | Tx Channel |
| | | 1 | Rx Channel |
| 30-29 | Reserved | 0 | Reserved |
| 28-24 | entry3_channel | 0 - 1Fh | This field indicates the channel number that is to be given an opportunity to transfer data. If this is a Tx entry, the DMA will be presented with a scheduling 'credit' for that exact Tx channel. If this is an Rx entry, the DMA will be presented with a scheduling credit for the Rx FIFO that is associated with this channel. For Rx FIFOs which carry traffic for more than 1 Rx DMA channel, the exact channel number that is given in the Rx credit will actually be the channel number which is currently on the head element of that Rx FIFO, which is not necessarily the channel number given in the scheduler table entry. |
| 23 | entry2_rxtx | | This bit indicates if this entry is for a Tx or an Rx channel and is encoded as follows: |
| | | 0 | Tx Channel |
| | | 1 | Rx Channel |
| 22-21 | Reserved | 0 | Reserved |
| 20-16 | entry2_channel | 0 - 1Fh | This field indicates the channel number that is to be given an opportunity to transfer data. If this is a Tx entry, the DMA will be presented with a scheduling 'credit' for that exact Tx channel. If this is an Rx entry, the DMA will be presented with a scheduling credit for the Rx FIFO that is associated with this channel. For Rx FIFOs which carry traffic for more than 1 Rx DMA channel, the exact channel number that is given in the Rx credit will actually be the channel number which is currently on the head element of that Rx FIFO, which is not necessarily the channel number given in the scheduler table entry. |
| 15 | entry1_rxtx | | This bit indicates if this entry is for a Tx or an Rx channel and is encoded as follows: |
| | | 0 | Tx Channel |
| | | 1 | Rx Channel |
| 14-13 | Reserved | 0 | |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-131. CPPI DMA Scheduler Table Word N (WORDn) Field Descriptions (continued)**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 12-8 | entry1_channel | 0 - 1Fh | This field indicates the channel number that is to be given an opportunity to transfer data. If this is a Tx entry, the DMA will be presented with a scheduling 'credit' for that exact Tx channel. If this is an Rx entry, the DMA will be presented with a scheduling credit for the Rx FIFO that is associated with this channel. For Rx FIFOs which carry traffic for more than 1 Rx DMA channel, the exact channel number that is given in the Rx credit will actually be the channel number which is currently on the head element of that Rx FIFO, which is not necessarily the channel number given in the scheduler table entry. |
| 7 | entry0_rxtx | | This bit indicates if this entry is for a Tx or an Rx channel and is encoded as follows: |
| | | 0 | Tx Channel |
| | | 1 | Rx Channel |
| 6-5 | Reserved | 0 | Reserved |
| 4-0 | entry0_channel | 0 - 1Fh | This field indicates the channel number that is to be given an opportunity to transfer data. If this is a Tx entry, the DMA will be presented with a scheduling 'credit' for that exact Tx channel. If this is an Rx entry, the DMA will be presented with a scheduling credit for the Rx FIFO that is associated with this channel. For Rx FIFOs which carry traffic for more than 1 Rx DMA channel, the exact channel number that is given in the Rx credit will actually be the channel number which is currently on the head element of that Rx FIFO, which is not necessarily the channel number given in the scheduler table entry. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5 CPPI DMA Queue Manager Registers

Table 25-132 lists the registers for the CPPI DMA Queue Manager submodule.

**Table 25-132. CPPI DMA Queue Manager Registers**

| Address Offset | Queue Manager Register |
|---|---|
| 4000h | Qmgr Revision Register |
| 4008h | Qmgr Queue Diversion Register |
| 4020h | Qmgr Free Descriptor/Buffer Starvation Count Qmgr Register 0 |
| 4024h | Qmgr Free Descriptor/Buffer Starvation Count Register 1 |
| 4028h | Qmgr Free Descriptor/Buffer Starvation Count Register 2 |
| 402Ch | Qmgr Free Descriptor/Buffer Starvation Count Register 3 |
| 4030h | Qmgr Free Descriptor/Buffer Starvation Count Qmgr Register 4 |
| 4034h | Qmgr Free Descriptor/Buffer Starvation Count Register 5 |
| 4038h | Qmgr Free Descriptor/Buffer Starvation Count Register 6 |
| 403Ch | Qmgr Free Descriptor/Buffer Starvation Count Register 7 |
| 4080h | Qmgr Linking RAM Region 0 Base Address Register |
| 4084h | Qmgr Linking RAM Region 0 Size Register |
| 4088h | Qmgr Linking RAM Region 1 Base Address Register |
| 4090h | Qmgr Queue Pending Register 0 |
| 4094h | Qmgr Queue Pending Register 1 |
| 4098h | Qmgr Queue Pending Register 2 |
| 409Ch | Qmgr Queue Pending Register 3 |
| 40A0h | Qmgr Queue Pending Register 4 |
| 5000h + 16 x R | Memory Region R Base Address Register |
| 5000h + 16 x R + 4 | Memory Region R Control Register |
| 6000h + 16 x N | Queue Manager Queue N Register A |
| 6004h + 16 x N | Queue Manager Queue N Register B |
| 6008h + 16 x N | Queue Manager Queue N Register C |
| 600Ch + 16 x N | Queue N Register D |
| 7000h + 16 x N | Queue N Status Register A |
| 7004h + 16 x N | Queue N Status Register B |
| 7008h + 16 x N | Queue N Status Register C |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.1 Queue Manager Revision Register (QMGRREVID)

The queue manager revision register contains the major and minor revisions for the module. It does not support byte accesses.

The revision register is shown in Figure 25-121 and described in Table 25-133.

**Figure 25-121. Queue Manager Revision Register (QMGRREVID)**

| 31 | 30 | 29 | 28 | 27 | 16 |
|---|---|---|---|---|---|
| Scheme | | Reserved | | function | |
| R-1h | | R-0h | | R-e53h | |

| 15 | 11 | 10 | 8 | 7 | 6 | 5 | 0 |
|---|---|---|---|---|---|---|---|
| revrtl | | revmaj | | revcustom | | revmin | |
| R-1h | | R-0h | | R-0h | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-133. Queue Manager Revision Register (QMGRREVID) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-30 | scheme | Scheme that this register is compliant with |
| 29-28 | Reserved | Reserved |
| 27-16 | function | Function |
| 15-11 | revrtl | RTL revision |
| 10-8 | revmaj | Major revision |
| 7-6 | revcustom | Custom revision |
| 5-0 | revmin | Minor revision |

### 25.9.5.2 Queue Manager Queue Diversion Register (DIVERSION)

The queue manager queue diversion register is used to transfer the contents of one queue onto another queue. It does not support byte accesses.

The queue manager queue diversion register is shown in Figure 25-122 and described in Table 25-134.

**Figure 25-122. Queue Manager Queue Diversion Register (DIVERSION)**

| 31 | 30 | 29 | 16 |
|---|---|---|---|
| head_tail | Rsvd | dest_qnum | |
| W-0h | R-0h | W-0h | |

| 15 | 14 | 13 | 0 |
|---|---|---|---|
| Reserved | | source_qnum | |
| R-0h | | W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-134. Queue Manager Queue Diversion Register (DIVERSION) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31 | head_tail | Indicates whether queue contents should be merged on to head or tail of destination queue. Clear this field for head and set for tail. |
| 30 | Reserved | Reserved |
| 29-16 | dest_qnum | Destination Queue Number |
| 15-14 | Reserved | Reserved |
| 13-0 | source_qnum | Source Queue Number |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.3 Queue Manager Free Descriptor/Buffer Starvation Count Register 0 (FDBSC0)

The free descriptor/buffer queue starvation count register 0 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 0-3. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 0 is shown in Figure 25-123 and described in Table 25-135.

**Figure 25-123. Queue Manager Free Descriptor/Buffer Starvation Count Register 0 (FDBSC0)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| fdbq3_starve_cnt | fdbq2_starve_cnt | fdbq1_starve_cnt | fdbq0_starve_cnt |
| RC-0h | RC-0h | RC-0h | RC-0h |

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

**Table 25-135. Queue Manager Free Descriptor/Buffer Starvation Count Register 0 (FDBSC0) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-24 | fdbq3_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 3 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 23-16 | fdbq2_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 2 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 15-8 | fdbq1_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 1 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 7-0 | fdbq0_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 0 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |

### 25.9.5.4 Queue Manager Free Descriptor/Buffer Starvation Count Register 1 (FDBSC1)

The free descriptor/buffer queue starvation count register 1 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 4-7. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 1 is shown in Figure 25-124 and described in Table 25-136.

**Figure 25-124. Queue Manager Free Descriptor/Buffer Starvation Count Register 1 (FDBSC1)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| fdbq7_starve_cnt | fdbq6_starve_cnt | fdbq5_starve_cnt | fdbq4_starve_cnt |
| RC-0h | RC-0h | RC-0h | RC-0h |

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

**Table 25-136. Queue Manager Free Descriptor/Buffer Starvation Count Register 1 (FDBSC1) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-24 | fdbq7_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 7 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 23-16 | fdbq6_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 6 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 15-8 | fdbq5_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 5 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 7-0 | fdbq4_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 4 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.5 Queue Manager Free Descriptor/Buffer Starvation Count Register 2 (FDBSC2)

The free descriptor/buffer queue starvation count register 2provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 8-11. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 2 is shown in Figure 25-125 and described in Table 25-137.

**Figure 25-125. Queue Manager Free Descriptor/Buffer Starvation Count Register 2 (FDBSC2)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| fdbq11_starve_cnt | fdbq10_starve_cnt | fdbq9_starve_cnt | fdbq8_starve_cnt |
| RC-0h | RC-0h | RC-0h | RC-0h |

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

**Table 25-137. Queue Manager Free Descriptor/Buffer Starvation Count Register 2 (FDBSC2) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-24 | fdbq11_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 11 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 23-16 | fdbq10_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 10 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 15-8 | fdbq9_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 9 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 7-0 | fdbq8_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 8 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |

### 25.9.5.6 Queue Manager Free Descriptor/Buffer Starvation Count Register 3 (FDBSC3)

The free descriptor/buffer queue starvation count register 3 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 12-15. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 3 is shown in Figure 25-126 and described in Table 25-138.

**Figure 25-126. Queue Manager Free Descriptor/Buffer Starvation Count Register 3 (FDBSC3)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| fdbq15_starve_cnt | fdbq14_starve_cnt | fdbq13_starve_cnt | fdbq12_starve_cnt |
| RC-0h | RC-0h | RC-0h | RC-0h |

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

**Table 25-138. Queue Manager Free Descriptor/Buffer Starvation Count Register 3 (FDBSC3) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-24 | fdbq15_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 15 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 23-16 | fdbq14_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 14 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 15-8 | fdbq13_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 13 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 7-0 | fdbq12_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 12 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.7 Queue Manager Free Descriptor/Buffer Starvation Count Register 4 (FDBSC4)

The free descriptor/buffer queue starvation count register 4 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 16-19. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 4 is shown in Figure 25-127 and described in Table 25-139.

**Figure 25-127. Queue Manager Free Descriptor/Buffer Starvation Count Register 4 (FDBSC4)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| fdbq19_starve_cnt | fdbq18_starve_cnt | fdbq17_starve_cnt | fdbq16_starve_cnt |
| RC-0h | RC-0h | RC-0h | RC-0h |

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

**Table 25-139. Queue Manager Free Descriptor/Buffer Starvation Count Register 4 (FDBSC4) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-24 | fdbq19_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 19 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 23-16 | fdbq18_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 18 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 15-8 | fdbq17_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 17 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 7-0 | fdbq16_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 16 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |

### 25.9.5.8 Queue Manager Free Descriptor/Buffer Starvation Count Register 5 (FDBSC5)

The free descriptor/buffer queue starvation count register 5 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 20-23. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 5 is shown in Figure 25-128 and described in Table 25-140.

**Figure 25-128. Queue Manager Free Descriptor/Buffer Starvation Count Register 5 (FDBSC5)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| fdbq23_starve_cnt | fdbq22_starve_cnt | fdbq21_starve_cnt | fdbq20_starve_cnt |
| RC-0h | RC-0h | RC-0h | RC-0h |

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

**Table 25-140. Queue Manager Free Descriptor/Buffer Starvation Count Register 5 (FDBSC5) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-24 | fdbq23_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 23 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 23-16 | fdbq22_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 22 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 15-8 | fdbq21_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 21 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 7-0 | fdbq20_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 20 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.9 Queue Manager Free Descriptor/Buffer Starvation Count Register 6 (FDBSC6)

The free descriptor/buffer queue starvation count register 6 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 24-27. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 6 is shown in Figure 25-129 and described in Table 25-141.

**Figure 25-129. Queue Manager Free Descriptor/Buffer Starvation Count Register 6 (FDBSC6)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| fdbq27_starve_cnt | fdbq26_starve_cnt | fdbq25_starve_cnt | fdbq24_starve_cnt |
| RC-0h | RC-0h | RC-0h | RC-0h |

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

**Table 25-141. Queue Manager Free Descriptor/Buffer Starvation Count Register 6 (FDBSC6) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-24 | fdbq27_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 27 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 23-16 | fdbq26_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 26 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 15-8 | fdbq25_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 25 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 7-0 | fdbq24_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 24 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |

### 25.9.5.10 Queue Manager Free Descriptor/Buffer Starvation Count Register 7 (FDBSC7)

The free descriptor/buffer queue starvation count register 7 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 28-31. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 7 is shown in Figure 25-130 and described in Table 25-142.

**Figure 25-130. Queue Manager Free Descriptor/Buffer Starvation Count Register 7 (FDBSC7)**

| 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|
| fdbq31_starve_cnt | fdbq30_starve_cnt | fdbq29_starve_cnt | fdbq28_starve_cnt |
| RC-0h | RC-0h | RC-0h | RC-0h |

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

**Table 25-142. Queue Manager Free Descriptor/Buffer Starvation Count Register 7 (FDBSC7) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-24 | fdbq31_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 31 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 23-16 | fdbq30_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 30 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 15-8 | fdbq29_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 29 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |
| 7:0 | fdbq28_starve_cnt | This field increments each time the Free Descriptor/Buffer Queue 28 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.11 Queue Manager Linking RAM Region 0 Base Address Register (LRAM0BASE)

The Linking RAM region 0 base address register is used to set the base address for the first portion of the Linking RAM. This address must be 32-bit aligned. It is used by the Queue Manager to calculate the 32-bit linking address for a given descriptor index. It does not support byte accesses.

The Linking RAM region 0 base address register is shown in Figure 25-131 and described in Table 25-143.

**Figure 25-131. Queue Manager Linking RAM Region 0 Base Address Register (LRAM0BASE)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| region0_base | | Rsvd | |
| R/W-0h | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

**Table 25-143. Queue Manager Linking RAM Region 0 Base Address Register (LRAM0BASE) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-2 | region0_base | This field stores the base address for the first region of the linking RAM. This may be anywhere in 32-bit address space but would be typically located in on-chip memory. |
| 1-0 | Reserved | Reserved. All data accesses must be words. |

### 25.9.5.12 Queue Manager Linking RAM Region 0 Size Register (LRAM0SIZE)

The Linking RAM region 0 size register is used to set the size of the array of linking pointers that are located in Region 0 of Linking RAM. The size specified the number of descriptors for which linking information is stored in this region. It does not support byte accesses.

The Linking RAM region 0 size register is shown in Figure 25-132 and described in Table 25-144.

**Figure 25-132. Queue Manager Linking RAM Region 0 Size Register (LRAM0SIZE)**

| 31 | 14 | 13 | 0 |
|---|---|---|---|
| Reserved | | region0_size | |
| R-0h | | R/W-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-144. Queue Manager Linking RAM Region 0 Size Register (LRAM0SIZE) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-14 | Reserved | Reserved |
| 13-0 | region0_size | This field indicates the number of entries that are contained in the linking RAM region 0. A descriptor with index less than region0_size value has its linking location in region 0. A descriptor with index greater than region0_size has its linking location in region 1. The queue manager will add the index (left shifted by 2 bits) to the appropriate regionX_base_addr to get the absolute 32-bit address to the linking location for a descriptor. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.13 Queue Manager Linking RAM Region 1 Base Address Register (LRAM1BASE)

The Linking RAM region 1 base address register is used to set the base address for the second portion of the Linking RAM. This base address is used by the Queue Manager to calculate the 32-bit linking address from the descriptor index. All descriptors with index higher than that given in Linking RAM 0 Size register have linking information stored in linking RAM region 1. It does not support byte accesses.

The Linking RAM region 1 base address register is shown in Figure 25-133 and described in Table 25-145.

**Figure 25-133. Queue Manager Linking RAM Region 1 Base Address Register (LRAM1BASE)**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| region1_base | | Rsvd | |
| R/W-0h | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-145. Queue Manager Linking RAM Region 1 Base Address Register (LRAM1BASE) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-2 | region1_base | This field stores the base address for the second region of the linking RAM. This may be anywhere in 32-bit address space but would be typically located in off-chip memory. |
| 1-0 | Reserved | Reserved |

### 25.9.5.14 Queue Manager Queue Pending Register 0 (PEND0)

The queue pending register 0 can be read to find the pending status for queues 31 to 0. It does not support byte accesses.

The queue pending register 0 is shown in Figure 25-134 and described in Table 25-146.

**Figure 25-134. Queue Manager Queue Pending Register 0 (PEND0)**

| 31 | 0 |
|---|---|
| qpend0 | |
| R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-146. Queue Manager Queue Pending Register 0 (PEND0) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | qpend0 | This field indicates the queue pending status for queues[31:0]. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.15 Queue Manager Queue Pending Register 1 (PEND1)

The queue pending register 1 can be read to find the pending status for queues 63 to 32. It does not support byte accesses.

The queue pending register 1 is shown in Figure 25-135 and described in Table 25-147.

**Figure 25-135. Queue Manager Queue Pending Register 1 (PEND1)**

| 31 | 0 |
|---|---|
| qpend1 | |
| R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-147. Queue Manager Queue Pending Register 1 (PEND1) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | qpend1 | This field indicates the queue pending status for queues[63:32] |

### 25.9.5.16 Queue Manager Queue Pending Register 2 (PEND2)

The queue pending register 2 can be read to find the pending status for queues 95 to 64. It does not support byte accesses.

The queue pending register 2 is shown in Figure 25-136 and described in Table 25-148.

**Figure 25-136. Queue Manager Queue Pending Register 2 (PEND2)**

| 31 | 0 |
|---|---|
| qpend2 | |
| R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-148. Queue Manager Queue Pending Register 2 (PEND1) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | qpend2 | This field indicates the queue pending status for queues[95:64] |

### 25.9.5.17 Queue Manager Queue Pending Register 3 (PEND3)

The queue pending register 3 can be read to find the pending status for queues 127 to 96. It does not support byte accesses.

The queue pending register 3 is shown in Figure 25-137 and described in Table 25-149.

**Figure 25-137. Queue Manager Queue Pending Register 3 (PEND3)**

| 31 | 0 |
|---|---|
| qpend3 | |
| R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-149. Queue Manager Queue Pending Register 3 (PEND3) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | qpend3 | This field indicates the queue pending status for queues[127:96] |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.18 Queue Manager Queue Pending Register 4 (PEND4)

The queue pending register 4 can be read to find the pending status for queues 159 to 128. It does not support byte accesses.

The queue pending register 4 is shown in Figure 25-138 and described in Table 25-150.

**Figure 25-138. Queue Manager Queue Pending Register 4 (PEND4)**

| 31 | 0 |
|---|---|
| qpend4 | |
| R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-150. Queue Manager Queue Pending Register 4 (PEND4) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-0 | qpend4 | This field indicates the queue pending status for queues[159:128] |

### 25.9.5.19 Queue Manager Memory Region R Base Address Register (QMEMRBASEr)

The memory region R base address register is written by the host to set the base address of memory region R. This memory region will store a number of descriptors of a particular size as determined by the Memory Region R Control Register. It does not support byte accesses. R ranges from 0 to 15.

The memory region R base address register is shown in Figure 25-139 and described in Table 25-151.

**Figure 25-139. Queue Manager Memory Region R Base Address Register (QMEMRBASEr)**

| 31 | 5 | 4 | 0 |
|---|---|---|---|
| reg | | Reserved | |
| R/W-0h | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-151. Queue Manager Memory Region R Base Address Register (QMEMRBASEr) Fi`eld Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-5 | reg | This field contains the base address of the memory region R. |
| 4-0 | Reserved | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.20 Queue Manager Memory Region R Control Register (QMEMRCTRLr)

The memory region R control register is written by the host to configure various parameters of this memory region. It does not support byte accesses. R ranges from 0 to 15.

The memory region R control register is shown in Figure 25-140 and described in Table 25-152.

**Figure 25-140. Queue Manager Memory Region R Control Register (QMEMRCTRLr)**

| 31 30 | 29 16 | 15 12 | 11 8 | 7 3 | 2 0 |
|---|---|---|---|---|---|
| Rsvd | start_index | Reserved | desc_size | Reserved | reg_size |
| R-0h | R/W-0h | R-0h | R/W-0h | R-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-152. Queue Manager Memory Region R Control Register (QMEMRCTRLr) Field Descriptions**

| Bits | Field Name | Description | Type | Reset Value |
|---|---|---|---|---|
| 31-30 | Reserved | Reserved | R | 0h |
| 29-16 | start_index | This field indicates where in linking RAM does the descriptor linking information corresponding to memory region R starts. | R/W | 0h |
| 15-12 | Reserved | Reserved | R | 0h |
| 11:08 | desc_size | This field indicates the size of each descriptor in this memory region. It is an encoded value that specifies descriptor size as 2^(5+desc_size) number of bytes. The settings of desc_size from 9-15 are reserved. | R/W | 0h |
| 7-3 | Reserved | Reserved | R | 0h |
| 2-0 | reg_size | This field indicates the size of the memory region (in terms of number of descriptors). It is an encoded value that specifies region size as 2^(5+reg_size) number of descriptors. | R/W | 0h |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.21 Queue Manager Queue N Register A (CTRLAn)

The queue N register A is an optional register that is only implemented for a queue if the queue supports entry/byte count feature. The entry count feature provides a count of the number of entries that are currently valid in the queue. It does not support byte accesses. N ranges from 0 to 155.

The queue N register A is shown in Figure 25-141 and described in Table 25-153.

**Figure 25-141. Queue Manager Queue N Register A (CTRLAn)**

| 31 | 14 | 13 | 0 |
|---|---|---|---|
| Reserved | | queue_entry_count | |
| R-0h | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-153. Queue Manager Queue N Register A (CTRLAn) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-14 | Reserved | Reserved |
| 13-0 | queue_entry_count | This field indicates how many packets are currently queued on the queue. This count is incremented by 1 whenever a packet is added to the queue. This count is decremented by 1 whenever a packet is popped from the queue. |

### 25.9.5.22 Queue Manager Queue N Register B (CTRLBn)

The Queue N Register B is an optional register that is only implemented for a queue if the queue supports a total byte count feature. The total byte count feature provides a count of the total number of bytes in all of the packets that are currently valid in the queue. This register must be read prior to reading Queue N register D during packet pop operation if the total size information is desired. It does not support byte accesses. N ranges from 0 to 155.

The queue N register B is shown in Figure 25-142 and described in Table 25-154.

**Figure 25-142. Queue Manager Queue N Register B (CTRLBn)**

| 31 | 28 | 27 | 0 |
|---|---|---|---|
| Reserved | | queue_byte_count | |
| R-0h | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-154. Queue Manager Queue N Register B (CTRLBn) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-28 | Reserved | Reserved |
| 27-0 | queue_byte_count | This field indicates how many bytes total are contained in all of the packets which are currently queued on this queue. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.23 Queue Manager Queue N Register C (CTRLCn)

The Queue N Register C is used to provide additional information about the packet that is being pushed or popped from the queue. This register provides an option for the packet to be pushed onto either the tail of the queue (default) or the head of the queue (override). This register must be written prior to writing the Queue N register D during packet write operations. This register must be read prior to reading Queue N register D during pop operations if the packet size information is desired. It does not support byte accesses. N ranges from 0 to 155.

The queue N register C is shown in Figure 25-143 and described in Table 25-155.

**Figure 25-143. Queue Manager Queue N Register C (CTRLCn)**

| 31 | 30 14 | 13 0 |
|---|---|---|
| head_tail | Reserved | packet_size |
| W-0h | R-0h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-155. Queue Manager Queue N Register C (CTRLCn) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31 | head_tail | Head/Tail Push Control. Set to 0 in order to push packet onto tail of queue and set to 1 in order to push packet onto head of queue. |
| 30-14 | Reserved | Reserved |
| 13-0 | packet_size | This field indicates packet size of the head element of a queue. This field indicates packet size for packet pop operation. |

### 25.9.5.24 Queue Manager Queue N Register D (CTRLDn)

The queue N register D is written to add a packet to the queue and read to pop a packets off a queue. The packet is only pushed or popped to/from the queue when the queue register D is written. It does not support byte accesses. N ranges from 0 to 155.

The queue N register D is shown in Figure 25-144 and described in Table 25-156.

**Figure 25-144. Queue Manager Queue N Register D (CTRLDn)**

| 31 5 | 4 0 |
|---|---|
| desc_ptr | desc_size |
| R/W-0h | R/W-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-156. Queue Manager Queue N Register D (CTRLDn) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-5 | desc_ptr | Descriptor pointer. It will be read as zero if the queue is empty. It will indicate a 32-bit aligned address that points to a descriptor when the queue is not empty. |
| 4-0 | desc_size | Descriptor Size. It is encoded in 4-byte increments with values 0 to 31 representing 24 and so on to 148 bytes. This field will return a 0x0 when an empty queue is read. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.25 Queue Manager Queue N Status Register A (QSTATAn)

The Queue N Status Register A is an optional register that is only implemented for a queue if the queue supports entry/byte count feature. The entry count feature provides a count of the number of entries that are currently valid in the queue. It does not support byte accesses. N ranges from 0 to 155.

The queue N status register A is shown in Figure 25-145 and described in Table 25-157.

**Figure 25-145. Queue Manager Queue N Register A (QSTATAn)**

| 31 | 14 | 13 | 0 |
|---|---|---|---|
| Reserved | | queue_entry_count | |
| R-0h | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-157. Queue Manager Queue N Status Register A (QSTATAn) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-14 | Reserved | Reserved |
| 13-0 | queue_entry_count | This field indicates how many packets are currently queued on the queue. |

### 25.9.5.26 Queue Manager Queue N Status Register B (QSTATBn)

The Queue N Status Register B is an optional register that is only implemented for a queue if the queue supports a total byte count feature. The total byte count feature provides a count of the total number of bytes in all of the packets that are currently valid in the queue. It does not support byte accesses. N ranges from 0 to 155.

The queue N status register B is shown in Figure 25-146 and described in Table 25-158.

**Figure 25-146. Queue Manager Queue N Status Register B (QSTATBn)**

| 31 | 28 | 27 | 0 |
|---|---|---|---|
| Reserved | | queue_byte_count | |
| R-0h | | R-0h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-158. Queue Manager Queue N Status Register B (QSTATBn) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-28 | Reserved | Reserved |
| 27-0 | queue_byte_count | This field indicates how many bytes total are contained in all of the packets which are currently queued on this queue. |

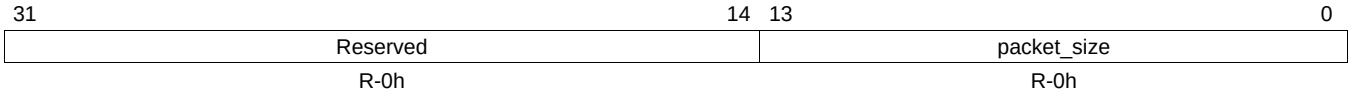

Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.5.27 Queue Manager Queue N Status Register C (QSTATCn)

The Queue N Status Register C specifies the packet size for the head element of a queue. It does not support byte accesses. N ranges from 0 to 155.

The queue N status register C is shown in Figure 25-147 and described in Table 25-159.

**Figure 25-147. Queue Manager Queue N Status Register C (QSTATCn)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-159. Queue Manager Queue N Status Register C (QSTATCn) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 31-14 | Reserved | Reserved |
| 13-0 | packet_size | This field indicates packet size of the head element of a queue. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.6 USB Mentor Core Registers

The Mentor Control register map is divided into the following sections:

- **Common USB registers (0h–Fh)** – These registers provide control and status for the complete core.
- **Indexed Endpoint Control/Status registers (10h–1Fh)** – These registers provide control and status for the currently selected endpoint. The registers mapped into this section depend on whether the core is in Peripheral mode or in Host mode and on the value of the Index register.
- **FIFOs (20h–5Fh)** – This address range provides access to the endpoint FIFOs.
- Additional Control and Configuration registers (60h–7Fh) – These registers provide additional device status and control.
- **Target Endpoint Control Registers (80h–FFh)** – These registers provide target function and hub address details for each of the endpoints.
- **Non-Indexed Endpoint Control/Status registers (100h and above)** – The registers available at 10h–1Fh, accessible independently of the setting of the Index register. 100h–10Fh, EP0 registers; 110h–11Fh, EP1 registers; 120h–12Fh, EP2; and so on.

Two instances of the Mentor Core registers exist within USB0 space. Since both USB modules operate independent of each other, each USB core has its own set of registers. USB0 Mentor Base Address is 0x4740_1400 while USB1 Mentor Core Registers Base address is 0x4740_1C00. Below offset of the core registers The absolute address is computed by summing the CORE_OFFSET with USBn_OFFSET.

#### 25.9.6.1 Common USB Registers

These registers provide control and status for the complete core. Table 25-160 lists the registers.

**Table 25-160. Common USB Registers**

| Core Address Offset | Common USB Registers |
|---|---|
| 0h | Function Address Register |
| 1h | Power Management Register |
| 2h | Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 |
| 4h | Interrupt Register for Receive Endpoints 1 to 15 |
| 6h | Interrupt Enable Register for INTRTX |
| 8h | Interrupt Enable Register for INTRRX |
| Ah | Interrupt Register for Common USB Interrupts |
| Bh | Interrupt Enable Register for INTRUSB |
| Ch | Frame Number Register |
| Eh | Index Register for Selecting the Endpoint Status and Control Registers |
| Fh | Register to Enable the USB 2.0 Test Modes |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.1.1 Function Address Register (USBn_FADDR)

The function address register (USBn_FADDR) is an 8-bit register that should be written with the 7-bit address of the peripheral part of the transaction.

Since USB cores are configured with multipoint support, this register only applies to operations carried out when the controller is in peripheral mode. In Host mode, this register is ignored.

The function address register is shown in Figure 25-148 and described in Table 25-161.

**Figure 25-148. Function Address Register (USBn_FADDR)**

| 7 | 6 | 0 |
|---|---|---|
| Reserved | FUNCADDR | |
| R-0h | W-0-7Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-161. Function Address Register (USBn_FADDR) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 7 | Reserved | Reserved |
| 6-0 | FUNCADDR | 7_bit address of the peripheral part of the transaction. When used in peripheral mode, this register should be written with the address received through a SET_ADDRESS command, which is then used for decoding the function address in subsequent token packets. |

#### 25.9.6.1.2 Power Management Register (USBn_POWER)

The power management register (USBn_POWER) is an 8-bit register that is used for controlling suspend and resume signaling, and some basic operational aspects of the USB core. This register is shown in Figure 25-149 and described in Table 25-162.

**Figure 25-149. Power Management Register (USBn_POWER)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ISOUPDATE | SOFTCONN | HSEN | HSMODE | RESET | RESUME | SUSPENDM | ENSUSPM |
| R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-162. Power Management Register (USBn_POWER) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 7 | ISOUPDATE | When set, the USB controller will wait for an SOF token from the time TxPktRdy is set before sending the packet. If an IN token is received before an SOF token, then a zero length data packet will be sent. Note: this is only valid in Peripheral Mode. This bit only affects endpoints performing Isochronous transfers. |
| 6 | SOFTCONN | If Soft Connect/Disconnect feature is enabled, then the USB D+/D-lines are enabled when this bit is set and tri-stated when this bit is cleared. Note: This is only valid in Peripheral Mode. |
| 5 | HSEN | When set, the USB controller negotiates for high-speed mode when the device is reset by the hub. If not set, the device will only operate in full-speed mode. |
| 4 | HSMODE | This bit is set when the USB controller has successfully negotiated for high-speed mode. |
| 3 | RESET | This bit is set when reset signaling is present on the bus. Note: this bit is Read/Write in Host Mode, but read-only in peripheral mode. |
| 2 | RESUME | Set to generate resume signaling when the controller is in suspend mode. The bit should be cleared after 10 ms (a maximum of 15 ms) to end resume signaling. In Host mode, this bit is also automatically set when resume signaling from the target is detected while the USB controller is suspended. |
| 1 | SUSPENDM | In Host mode, this bit should be set to enter suspend mode. In peripheral mode, this bit is set on entry into suspend mode. It is cleared when the interrupt register is read, or the RESUME bit is set. |
| 0 | ENSUSPM | Set to enable the SUSPENDM output. |



Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.1.3 Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 (USBn_INTRTX)

The interrupt register for endpoint 0 plus transmit endpoints 1 to 15 (USBn_INTRTX) is a 16-bit read-only register that indicates which interrupts are currently active for endpoint 0 and the TX endpoints 1–15. Note also that all active interrupts are cleared when this register is read.

The interrupt register for endpoint 0 plus transmit endpoints 1 to 15 is shown in Figure 25-150 and described in Table 25-163.

**Figure 25-150. Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 (USBn_INTRTX)**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EP15TX | EP14TX | EP13TX | EP12TX | EP11TX | EP10TX | EP9TX | EP8TX |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EP7TX | EP6TX | EP5TX | EP4TX | EP3TX | EP2TX | EP1TX | EP0 |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-163. Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 (USBn_INTRTX) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 15 | EP15TX | Transmit endpoint 15 interrupt active |
| 14 | EP14TX | Transmit endpoint 14 interrupt active |
| : | : | : |
| 4 | EP4TX | Transmit endpoint 4 interrupt active |
| 3 | EP3TX | Transmit endpoint 3 interrupt active |
| 2 | EP2TX | Transmit endpoint 2 interrupt active |
| 1 | EP1TX | Transmit endpoint 1 interrupt active |
| 0 | EP0 | Endpoint 0 (transmit or receive) interrupt active |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.1.4 Interrupt Register for Receive Endpoints 1 to 15 (INTRRX)

The interrupt register for receive endpoints 1 to 15 (INTRRX) is a 16-bit read-only register that indicates which of the interrupts for Rx Endpoints 1 – 15 are currently active. Note also that all active interrupts are cleared when this register is read.

The interrupt register for receive endpoints 1 to 15 is shown in Figure 25-151 and described in Table 25-164.

**Figure 25-151. Interrupt Register for Receive Endpoints 1 to 15 (INTRRX)**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EP15RX | EP14RX | EP13RX | EP12RX | EP11RX | EP10RX | EP9RX | EP8RX |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EP7RX | EP6RX | EP5RX | EP4RX | EP3RX | EP2RX | EP1RX | Reserved |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-164. Interrupt Register for Receive Endpoints 1 to 15 (INTRRX) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 15 | EP15RX | Receive endpoint 15 interrupt active |
| 14 | EP14RX | Receive endpoint 14 interrupt active |
| : | : | : |
| 4 | EP4RX | Receive endpoint 4 interrupt active |
| 3 | EP3RX | Receive endpoint 3 interrupt active |
| 2 | EP2RX | Receive endpoint 2 interrupt active |
| 1 | EP1RX | Receive endpoint 1 interrupt active |
| 0 | Reserved | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.1.5 Interrupt Enable Register for INTRTX (INTRTXE)

The interrupt enable register for INTRTX (INTRTXE) is a 16-bit register that provides interrupt enable bits for the interrupts in IntrTx. On reset, the bits corresponding to endpoint 0 and the TX endpoints are set to 1. This register is shown in Figure 25-152 and described in Table 25-165.

**Figure 25-152. Interrupt Enable Register for INTRTX (INTRTXE)**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EP15TX | EP14TX | EP13TX | EP12TX | EP11TX | EP10TX | EP9TX | EP8TX |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EP7TX | EP6TX | EP5TX | EP4TX | EP3TX | EP2TX | EP1TX | EP0 |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-165. Interrupt Enable Register for INTRTX (INTRTXE) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 15 | EP15TX | Transmit endpoint 15 interrupt enable |
| 14 | EP14TX | Transmit endpoint 14 interrupt enable |
| : | : | : |
| 4 | EP4TX | Transmit endpoint 4 interrupt enable |
| 3 | EP3TX | Transmit endpoint 3 interrupt enable |
| 2 | EP2TX | Transmit endpoint 2 interrupt enable |
| 1 | EP1TX | Transmit endpoint 1 interrupt enable |
| 0 | EP0 | Endpoint 0 interrupt active |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.1.6 Interrupt Enable Register for INTRRX (INTRRXE)

The interrupt enable register for INTRRX (INTRRXE) is a 16-bit register that provides interrupt enable bits for the interrupts in USBn_INTRRX. On reset, the bits corresponding to the Rx endpoints are set to 1. This register is shown in Figure 25-153 and described in Table 25-166.

**Figure 25-153. Interrupt Enable Register for INTRRX (INTRRXE)**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EP15RX | EP14RX | EP13RX | EP12RX | EP11RX | EP10RX | EP9RX | EP8RX |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EP7RX | EP6RX | EP5RX | EP4RX | EP3RX | EP2RX | EP1RX | Reserved |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-166. Interrupt Enable Register for INTRRX (INTRRXE) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 15 | EP15RX | Receive endpoint 15 interrupt enable |
| 14 | EP14RX | Receive endpoint 14 interrupt enable |
| : | : | : |
| 4 | EP4RX | Receive endpoint 4 interrupt enable |
| 3 | EP3RX | Receive endpoint 3 interrupt enable |
| 2 | EP2RX | Receive endpoint 2 interrupt enable |
| 1 | EP1RX | Receive endpoint 1 interrupt enable |
| 0 | Reserved | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.1.7 Interrupt Register for Common USB Interrupts (USBn_INTRUSB)

The interrupt register for common USB interrupts (USBn_INTRUSB) is an 8-bit read-only register that indicates which USB interrupts are currently active. All active interrupts will be cleared when this register is read. This register is shown in Figure 25-154 and described in Table 25-167.

**Figure 25-154. Interrupt Register for Common USB Interrupts (USBn_INTRUSB)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VBUSERR | SESSREQ | DISCON | CONN | SOF | RESET_BABBLE | RESUME | SUSPEND |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-167. Interrupt Register for Common USB Interrupts (USBn_INTRUSB) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 7 | VBUSERR | Set when VBus drops below the VBus valid threshold during a session. Only valid when the USB controller is 'A' device. All active interrupts will be cleared when this register is read. |
| 6 | SESSREQ | Set when session request signaling has been detected. Only valid when USB controller is 'A' device. |
| 5 | DISCON | Set in host mode when a device disconnect is detected. Set in peripheral mode when a session ends. |
| 4 | CONN | Set when a device connection is detected. Only valid in host mode. |
| 3 | SOF | Set when a new frame starts. |
| 2 | RESET_BABBLE | Set in peripheral mode when reset signaling is detected on the bus set in host mode when babble is detected. |
| 1 | RESUME | Set when resume signaling is detected on the bus while the USB controller is in suspend mode. |
| 0 | SUSPEND | Set when suspend signaling is detected on the bus only valid in peripheral mode. |

#### 25.9.6.1.8 Interrupt Enable Register for INTRUSB (USBn_INTRUSBE)

The interrupt enable register for INTRUSB (USBn_INTRUSBE) is an 8-bit register that provides interrupt enable bits for each of the interrupts in USBn_INTRUSB. This register is shown in Figure 25-155 and described in Table 25-168.

**Figure 25-155. Interrupt Enable Register for INTRUSB (USBn_INTRUSBE)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VBUSERR | SESSREQ | DISCON | CONN | SOF | RESET_BABBLE | RESUME | SUSPEND |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-168. Interrupt Enable Register for INTRUSB (USBn_INTRUSBE) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 7 | VBUSERR | Vbus error interrupt enable |
| 6 | SESSREQ | Session request interrupt enable |
| 5 | DISCON | Disconnect interrupt enable |
| 4 | CONN | Connect interrupt enable |
| 3 | SOF | Start of frame interrupt enable |
| 2 | RESET_BABBLE | Reset interrupt enable |
| 1 | RESUME | Resume interrupt enable |
| 0 | SUSPEND | Suspend interrupt enable |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.1.9 Frame Number Register (USBn_FRAME)

The frame number register (USBn_FRAME) is a 16-bit read-only register that holds the last received frame number. This register is shown in Figure 25-156 and described in Table 25-169.

**Figure 25-156. Frame Number Register (USBn_FRAME)**

| 15 | 11 | 10 | 0 |
|---|---|---|---|
| Reserved | | FRAMENUMBER | |
| R-0h | | R-0-7FFh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-169. Frame Number Register (USBn_FRAME) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 15-11 | Reserved | Reserved |
| 10-0 | FRAMENUMBER | Last received frame number |

#### 25.9.6.1.10 Index Register for Selecting the Endpoint Status and Control Registers (USBn_INDEX)

Each TX endpoint and each Rx endpoint have their own set of control/status registers located between Core_Offset 0x0100 – 0x01FF. In addition one set of TX control/status and one set of Rx control/status registers appear at Core_Offset 0x0010 – 0x0019. This block of memory can be used as a proxy to access endpoint configuration registers. Index is a 4-bit register that determines which endpoint control/status registers are accessed.

Before accessing an endpoint's control/status registers at Core_Offset 0x0010 – 0x0019, the endpoint number should be written to the Index register to ensure that the correct control/status registers appear in the memory map.

The index register for selecting the endpoint status and control registers is shown in Figure 25-157 and described in Table 25-170.

**Figure 25-157. Index Register for Selecting the Endpoint Status and Control Registers (USBn_INDEX)**

| 7 | 4 | 3 | 0 |
|---|---|---|---|
| Reserved | | EPSEL | |
| R-0h | | R/W-0-Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-170. Index Register for Selecting the Endpoint Status and Control Registers (USBn_INDEX) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 7-4 | Reserved | Reserved |
| 3-0 | EPSEL | Each transmit endpoint and each receive endpoint have their own set of control/status registers. EPSEL determines which endpoint control/status registers are accessed. Before accessing an endpoint's control/status registers, the endpoint number should be written to the Index register to ensure that the correct control/status registers appear in the memory-map. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.1.11 Register to Enable the USB 2.0 Test Modes (USBn_TESTMODE)

The register to enable the USB 2.0 test modes (USBn_TESTMODE) is an 8-bit register that is primarily used to put the USB controller into one of the four test modes for high-speed operation described in the USB 2.0 specification – in response to a SET FEATURE: TESTMODE command. It is not used in normal operation.

The register to enable the USB 2.0 test modes is shown in Figure 25-158 and described in Table 25-171.

**Figure 25-158. Register to Enable the USB 2.0 Test Modes (USBn_TESTMODE)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FORCE_HOST | FIFO_ACCESS | FORCE_FS | FORCE_HS | TEST_PACKET | TEST_K | TEST_J | TEST_SE0_NAK |
| R/W-0-1h | W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-171. Register to Enable the USB 2.0 Test Modes (USBn_TESTMODE) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 7 | FORCE_HOST | Set this bit to forcibly put the USB controller into Host mode when SESSION bit is set, regardless of whether it is connected to any peripheral. The controller remains in Host mode until the session bit is cleared, even if a device is disconnected. And if the FORCE_HOST but remains set, it re-enters Host mode next time the SESSION bit is set. The operating speed is determined using the FORCE_HS and FORCE_FS bits. |
| 6 | FIFO_ACCESS | Set this bit to transfer the packet in EP0 Tx FIFO to EP0 Receive FIFO. It is cleared automatically. |
| 5 | FORCE_FS | Set this bit to force the USB controller into full-speed mode when it receives a USB reset. |
| 4 | FORCE_HS | Set this bit to force the USB controller into high-speed mode when it receives a USB reset. |
| 3 | TEST_PACKET | Set this bit to enter the Test_Packet test mode. In this mode, the USB controller repetitively transmits a 53-byte test packet on the bus, the form of which is defined in the Universal Serial Bus Specification Revision 2.0. Note: The test packet has a fixed format and must be loaded into the endpoint 0 FIFO before the test mode is entered. |
| 2 | TEST_K | Set this bit to enter the Test_K test mode. In this mode, the USB controller transmits a continuous K on the bus. |
| 1 | TEST_J | Set this bit to enter the Test_J test mode. In this mode, the USB controller transmits a continuous J on the bus. |
| 0 | TEST_SE0_NAK | Set this bit to enter the Test_SE0_NAK test mode. In this mode, the USB controller remains in high-speed mode, but responds to any valid IN token with a NAK. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.6.2 Indexed Register Space

This is a block of the core register space that is used as a proxy to access the configuration and status registers of an endpoint. Mapping is done by programming the INDEX register with the endpoint number. Table 25-172 lists the registers.

**Table 25-172. Indexed Region Registers**

| Core Address Offset | Indexed Region Registers |
|---|---|
| 10h | Maximum Packet Size for Peripheral/Host Transmit Endpoint |
| 12h | Control Status Register for Endpoint 0 in Peripheral Mode |
| 12h | Control Status Register for Endpoint 0 in Host Mode |
| 12h | Control Status Register for Peripheral Transmit Endpoint |
| 12h | Control Status Register for Host Transmit Endpoint |
| 14h | Maximum Packet Size for Peripheral/Host Receive Endpoint |
| 16h | Control Status Register for Peripheral Receive Endpoint |
| 16h | Control Status Register for Host Receive Endpoint |
| 18h | Count 0 Register |
| 18h | Receive Count Register |
| 1Ah | Type Register (Host mode only) |
| 1Ah | Transmit Type Register (Host mode only) |
| 1Bh | NAKLimit0 Register (Host mode only) |
| 1Bh | Transmit Interval Register (Host mode only) |
| 1Ch | Receive Type Register (Host mode only) |
| 1Dh | Receive Interval Register (Host mode only) |
| 1Fh | Configuration Data Register |

#### *25.9.6.2.1 Maximum Packet Size for Peripheral/Host Transmit Endpoint Register (USBn_TXMAXP)*

The maximum packet size for peripheral/host transmit endpoint register (USBn_TXMAXP) defines the maximum amount of data that can be transferred through the selected Tx endpoint in a single operation. There is a TxMaxP register for each Tx endpoint (except Endpoint 0). Endpoint 0 resource is hardened in design and max packet size is set to 64 bytes and no user configuration is required or does not exist.

The maximum packet size for peripheral/host transmit endpoint register is shown in Figure 25-159 and described in Table 25-173.

**Figure 25-159. Maximum Packet Size for Peripheral/Host Transmit Endpoint Register (USBn_TXMAXP)**

| 15 | 11 | 10 | 0 |
|---|---|---|---|
| Reserved | | MAXPAYLOAD | |
| R-0h | | R/W-0-400h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-173. Maximum Packet Size for Peripheral/Host Transmit Endpoint (USBn_TXMAXP) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 15-11 | Reserved | Reserved |
| 10-0 | MAXPAYLOAD | The maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes, but is subject to the constraints placed by the USB specification on packet sizes for bulk, interrupt, and isochronous transfers in full-speed and high-speed operations. The value written to this register should match the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch could cause unexpected results. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.2 Control Status Register for Endpoint 0 in Peripheral Mode (USBn_PERI_CSR0)

The control status register for endpoint 0 in peripheral mode (USBn_PERI_CSR0) is a 16-bit register that provides control and status bits for endpoint 0 when USB controller assumes the role of a peripheral. This register is shown in Figure 25-160 and described in Table 25-174.

**Figure 25-160. Control Status Register for Endpoint 0 in Peripheral Mode (USBn_PERI_CSR0)**

| 15 | | | | | | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | FLUSHFIFO |
| R-0h | | | | | | | W-0-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SERV_ SETUPEND | SERV_ RXPKTRDY | SENDSTALL | SETUPEND | DATAEND | SENTSTALL | TXPKTRDY | RXPKTRDY |
| W-0-1h | W-0-1h | W-0-1h | R-0-1h | W-0-1h | R/W-0-1h | R/W-0-1h | R-0h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-174. Control Status Register for Endpoint 0 in Peripheral Mode (USBn_PERI_CSR0) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 15-9 | Reserved | Reserved |
| 8 | FLUSHFIFO | Set this bit to flush the next packet to be transmitted/read from the endpoint 0 FIFO. The FIFO pointer is reset and the TXPKTRDY/RXPKTRDY bit is cleared. Note: FLUSHFIFO has no effect unless TXPKTRDY/RXPKTRDY is set. |
| 7 | SERV_SETUPEND | Set this bit to clear the SETUPEND bit. It is cleared automatically. |
| 6 | SERV_RXPKTRDY | Set this bit to clear the RXPKTRDY bit. It is cleared automatically. |
| 5 | SENDSTALL | Set this bit to terminate the current transaction. The STALL handshake will be transmitted and then this bit will be cleared automatically. |
| 4 | SETUPEND | This bit will be set when a control transaction ends before the DATAEND bit has been set. An interrupt is generated, and the FIFO will be flushed at this time. The bit is cleared by the writing a 1 to the SERV_SETUPEND bit. |
| 3 | DATAEND | Set this bit to 1: a. When setting TXPKTRDY for the last data packet. b. When clearing RXPKTRDY after unloading the last data packet. c. When setting TXPKTRDY for a zero length data packet. It is cleared automatically. |
| 2 | SENTSTALL | This bit is set when a STALL handshake is transmitted. This bit should be cleared. |
| 1 | TXPKTRDY | Set this bit after loading a data packet into the FIFO. It is cleared automatically when the data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared. |
| 0 | RXPKTRDY | This bit is set when a data packet has been received. An interrupt is generated when this bit is set. This bit is cleared by setting the SERV_RXPKTRDY bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.6.2.3 Control Status Register for Endpoint 0 in Host Mode (USBn_HOST_CSR0)

The control status register for endpoint 0 in host mode (USBn_HOST_CSR0) is a 16-bit register that provides control and status bits for endpoint 0 when USB controller assumes the role of a host. This register is shown in Figure 25-161 and described in Table 25-175.

**Figure 25-161. Control Status Register for Endpoint 0 in Host Mode (USBn_HOST_CSR0)**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | DSPING | DATATOGWREN | DATATOG | FLUSHFIFO |
| R-0h | | | | R/W-0-1h | W-0-1h | R/W-0-1h | W-0-1h |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| NAK_TIMEOUT | STATUSPKT | REQPKT | ERROR | SETUPPKT | RXSTALL | TXPKTRDY | RXPKTRDY |
| W-0-1h | R/W-0-1h | R/W-0-1h | W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 25-175. Control Status Register for Endpoint 0 in Host Mode (USBn_HOST_CSR0) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 15-12 | Reserved | Reserved |
| 11 | DSPING | The CPU writes a 1 to the DSPING bit to instruct the core not to issue PING tokens in the data and status phases of a high-speed control transfer (for use with devices that do not respond to PING). |
| 10 | DATATOGWREN | Write 1 to this bit to enable the DATATOG bit to be written. This bit is automatically cleared once the new value is written to DATATOG. |
| 9 | DATATOG | When read, this bit indicates the current state of the EP0 data toggle. If DATATOGWREN is high, this bit can be written with the required setting of the data toggle. If DATATOGWREN is low, any value written to this bit is ignored. |
| 8 | FLUSHFIFO | Write 1 to this bit to flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The FIFO pointer is reset and the TXPKTRDY/RXPKTRDY bit is cleared. Note: FLUSHFIFO has no effect unless TXPKTRDY/RXPKTRDY is set. |
| 7 | NAK_TIMEOUT | This bit will be set when endpoint 0 is halted following the receipt of NAK responses for longer than the time set by the NAKLIMIT0 register. This bit should be cleared to allow the endpoint to continue. |
| 6 | STATUSPKT | Set this bit at the same time as the TXPKTRDY or REQPKT bit is set, to perform a status stage transaction. Setting this bit ensures that the data toggle is set so that a DATA1 packet is used for the Status Stage transaction. |
| 5 | REQPKT | Set this bit to request an IN transaction. It is cleared when RXPKTRDY is set. |
| 4 | ERROR | This bit is set when three attempts have been made to perform a transaction with no response from the peripheral. You should clear this bit. An interrupt is generated when this bit is set. |
| 3 | SETUPPKT | Set this bit, at the same time as the TXPKTRDY bit is set, to send a SETUP token instead of an OUT token for the transaction. |
| 2 | RXSTALL | This bit is set when a STALL handshake is received. You should clear this bit. |
| 1 | TXPKTRDY | Set this bit after loading a data packet into the FIFO. It is cleared automatically when the data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared. |
| 0 | RXPKTRDY | This bit is set when a data packet has been received. An interrupt is generated when this bit is set. Clear this bit by setting the SERV_RXPKTRDY bit. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.4 Control Status Register for Peripheral Transmit Endpoint (USBn_PERI_TXCSR)

The control status register for peripheral transmit endpoint (USBn_PERI_TXCSR) is a 16-bit register that provides control and status bits for transfers through the currently-selected Tx endpoint when controller assumes the role of a peripheral. There is a TXCSR register for each configured Tx endpoint (not including Endpoint 0).

The control status register for peripheral transmit endpoint is shown in Figure 25-162 and described in Table 25-176.

**Figure 25-162. Control Status Register for Peripheral Transmit Endpoint (USBn_PERI_TXCSR)**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| AUTOSET | ISO | MODE | DMAEN | FRCDATATOG | DMAMODE | Reserved | |
| R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R-0h | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | CLRDATATOG | SENTSTALL | SENDSTALL | FLUSHFIFO | UNDERRUN | FIFONOTEMPTY | TXPKTRDY |
| R-0h | W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 25-176. Control Status Register for Peripheral Transmit Endpoint (USBn_PERI_TXCSR) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 15 | AUTOSET | DMA Mode: The CPU needs to set the AUTOSET bit prior to enabling the Tx DMA. CPU Mode: If the CPU sets the AUTOSET bit, the TXPKTRDY bit will be automatically set when data of the maximum packet size (value in TXMAXP) is loaded into the Tx FIFO. If a packet of less than the maximum packet size is loaded, then the TXPKTRDY bit will have to be set manually. |
| 14 | ISO | Set this bit to enable the Tx endpoint for Isochronous transfers, and clear it to enable the Tx endpoint for Bulk or Interrupt transfers. |
| 13 | MODE | Set this bit to enable the endpoint direction as Tx, and clear the bit to enable it as Rx. Note: This bit has any effect only where the same endpoint FIFO is used for both Transmit and Receive transactions. |
| 12 | DMAEN | Set this bit to enable the DMA request for the Tx endpoint. |
| 11 | FRCDATATOG | Set this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by Interrupt Tx endpoints that are used to communicate rate feedback for Isochronous endpoints. |
| 10 | DMAMODE | This bit should always be set to 1 when the DMA is enabled. |
| 9-7 | Reserved | Reserved |
| 6 | CLRDATATOG | Write a 1 to this bit to reset the endpoint data toggle to 0. |
| 5 | SENTSTALL | This bit is set automatically when a STALL handshake is transmitted. The FIFO is flushed and the TXPKTRDY bit is cleared. You should clear this bit. |
| 4 | SENDSTALL | Write a 1 to this bit to issue a STALL handshake to an IN token. Clear this bit to terminate the stall condition. Note: This bit has no effect where the endpoint is being used for Isochronous transfers. |
| 3 | FLUSHFIFO | Write a 1 to this bit to flush the next packet to be transmitted from the endpoint Tx FIFO. The FIFO pointer is reset and the TXPKTRDY bit is cleared. Note: FlushFIFO has no effect unless the TXPKTRDY bit is set. Also note that, if the FIFO is double-buffered, FlushFIFO may need to be set twice to completely clear the FIFO. |
| 2 | UNDERRUN | This bit is set automatically if an IN token is received when TXPKTRDY is not set. You should clear this bit. |
| 1 | FIFONOTEMPTY | This bit is set when there is at least 1 packet in the Tx FIFO. You should clear this bit. |
| 0 | TXPKTRDY | Set this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.6.2.5 Control Status Register for Host Transmit Endpoint (USBn_HOST_TXCSR)

The control status register for host transmit endpoint (USBn_HOST_TXCSR) is a 16-bit register that provides control and status bits for transfers through the currently-selected Tx endpoint when controller assumes the role of a host. There is a TXCSR register for each configured Tx endpoint (not including Endpoint 0).

The control status register for host transmit endpoint is shown in Figure 25-163 and described in Figure 25-163.

**Figure 25-163. Control Status Register for Host Transmit Endpoint (USBn_HOST_TXCSR)**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| AUTOSET | Reserved | MODE | DMAEN | FRCDATATOG | DMAMODE | DATATOGWREN | DATATOG |
| R/W-0-1h | R-0h | | R/W-0-1h | | R/W-0-1h | W-0-1h | R/W-0-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NAK_TIMEOUT | CLRDATATOG | RXSTALL | SETUPPKT | FLUSHFIFO | ERROR | FIFONOTEMPTY | TXPKTRDY |
| R/W-0-1h | W-0-1h | R/W-0-1h | R/W-0-1h | W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 25-177. Control Status Register for Host Transmit Endpoint (USBn_HOST_TXCSR) Field Descriptions**

| Bits | Field Name | Description |
|---|---|---|
| 15 | AUTOSET | DMA Mode: The CPU needs to set the AUTOSET bit prior to enabling the Tx DMA. CPU Mode: If the CPU sets the AUTOSET bit, the TXPKTRDY bit will be automatically set when data of the maximum packet size (value in TXMAXP) is loaded into the Tx FIFO. If a packet of less than the maximum packet size is loaded, then the TXPKTRDY bit will have to be set manually. |
| 14 | Reserved | Reserved |
| 13 | MODE | Set this bit to enable the endpoint direction as Tx, and clear the bit to enable it as Rx. Note: This bit has any effect only where the same endpoint FIFO is used for both Transmit and Receive transactions. |
| 12 | DMAEN | Set this bit to enable the DMA request for the Tx endpoint. |
| 11 | FRCDATATOG | Set this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by Interrupt Tx endpoints that are used to communicate rate feedback for Isochronous endpoints. |
| 10 | DMAMODE | This bit should always be set to 1 when the DMA is enabled. |
| 9 | DATATOGWREN | Write 1 to this bit to enable the DATATOG bit to be written. This bit is automatically cleared once the new value is written to DATATOG. |
| 8 | DATATOG | When read, this bit indicates the current state of the Tx EP data toggle. If DATATOGWREN is high, this bit can be written with the required setting of the data toggle. If DATATOGWREN is low, any value written to this bit is ignored. |
| 7 | NAK_TIMEOUT | This bit will be set when the Tx endpoint is halted following the receipt of NAK responses for longer than the time set as the NAKLIMIT by the TXINTERVAL register. It should be cleared to allow the endpoint to continue. Note: This is valid only for Bulk endpoints. |
| 6 | CLRDATATOG | Write a 1 to this bit to reset the endpoint data toggle to 0. |
| 5 | RXSTALL | This bit is set when a STALL handshake is received. The FIFO is flushed and the TXPKTRDY bit is cleared (see below). You should clear this bit. |
| 4 | SETUPPKT | Set this bit at the same time as TXPKTRDY is set, to send a SETUP token instead of an OUT token for the transaction. Note: Setting this bit also clears the DATATOG bit. |
| 3 | FLUSHFIFO | Write a 1 to this bit to flush the next packet to be transmitted from the endpoint Tx FIFO. The FIFO pointer is reset and the TXPKTRDY bit (below) is cleared. Note: FlushFIFO has no effect unless the TXPKTRDY bit is set. Also note that, if the FIFO is double-buffered, FLUSHFIFO may need to be set twice to completely clear the FIFO. |
| 2 | ERROR | The USB controller sets this bit when 3 attempts have been made to send a packet and no handshake packet has been received. You should clear this bit. An interrupt is generated when the bit is set. This is valid only when the endpoint is operating in Bulk or Interrupt mode. |
| 1 | FIFONOTEMPTY | The USB controller sets this bit when there is at least 1 packet in the Tx FIFO. |
| 0 | TXPKTRDY | Set this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.6 Maximum Packet Size for Peripheral/Host Receive Endpoint Register (USBn_RXMAXP)

The maximum packet size for peripheral/host receive endpoint register (USBn_RXMAXP) defines the maximum amount of data that can be transferred through the selected Rx endpoint in a single operation. There is a RxMaxP register for each Rx endpoint (except Endpoint 0). Endpoint 0 resource is hardened in design and max packet size is set to 64 bytes and no user configuration is required or does not exist.

The maximum packet size for peripheral/host receive endpoint register is shown in Figure 25-164 and described in Table 25-178.

**Figure 25-164. Maximum Packet Size for Peripheral/Host Receive Endpoint Register (USBn_RXMAXP)**

| 15 | 11 | 10 | 0 |
|---|---|---|---|
| Reserved | | MAXPAYLOAD | |
| R-0h | | RW-0-400h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-178. Maximum Packet Size for Peripheral Host Receive Endpoint (USBn_RXMAXP) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 15-11 | Reserved | Reserved |
| 10-0 | MAXPAYLOAD | Defines the maximum amount of data that can be transferred through the selected Receive endpoint in a single frame/microframe (high-speed transfers). The value set can be up to 1024 bytes, but is subject to the constraints placed by the USB specification on packet sizes for bulk, interrupt, and isochronous transfers in full-speed and high-speed operations. The value written to this register should match the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch could cause unexpected results. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.7 Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR)

The control status register for peripheral receive endpoint (USBn_PERI_RXCSR) is a 16-bit register that provides control and status bits for transfers through the currently-selected Tx endpoint when controller assumes the role of a peripheral. There is a RXCSR register for each configured Rx endpoint (not including endpoint 0).

The control status register for peripheral receive endpoint is shown in Figure 25-165 and described in Table 25-179.

**Figure 25-165. Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR)**

| 15 | 14 | 13 | 12 |
|---|---|---|---|
| AUTOCLEAR | ISO | DMAEN | DISNYET |
| R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h |
| 11 | 10 | 9 | 8 |
| DMAMODE | DATATOGWREN | DATATOG | Reserved |
| R/W-0-1h | W-0-1h | R/W-0-1h | R-0h |
| 7 | 6 | 5 | 4 |
| CLRDATATOG | SENTSTALL | SENDSTALL | FLUSHFIFO |
| W-0-1h | R/W-0-1h | R/W-0-1h | W-0-1h |
| 3 | 2 | 1 | 0 |
| DATAERROR | OVERRUN | FIFOFULL | RXPKTRDY |
| R-0-1h | R/W-0-1h | R-0-1h | R/W-0-1h |

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

**Table 25-179. Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 15 | AUTOCLEAR | DMA Mode: The CPU sets the AUTOCLEAR bit prior to enabling the Rx DMA. CPU Mode: If the CPU sets the AUTOCLEAR bit, then the RXPKTRDY bit will be automatically cleared when a packet of RXMAXP bytes has been unloaded from the Receive FIFO. When packets of less than the maximum packet size are unloaded, RXPKTRDY will have to be cleared manually. |
| 14 | ISO | Set this bit to enable the Receive endpoint for Isochronous transfers, and clear it to enable the Receive endpoint for bulk/interrupt transfers. |
| 13 | DMAEN | Set this bit to enable the DMA request for the Receive endpoints. |
| 12 | DISNYET | DISNYET: Applies only for bulk/interrupt transactions: The CPU sets this bit to disable the sending of NYET handshakes. When set, all successfully received Rx packets are ACK'd including at the point at which the FIFO becomes full. Note: This bit only has any effect in high-speed mode, in which mode it should be set for all Interrupt endpoints. PID_ERROR: Applies only for ISO Transactions: The core sets this bit to indicate a PID error in the received packet. |
| 11 | DMAMODE | Always clear this bit to 0. |
| 10 | DATATOGWREN | Write 1 to this bit to enable the DATATOG bit to be written. This bit is automatically cleared once the new value is written to DATATOG. |
| 9 | DATATOG | When read, this bit indicates the current state of the Receive EP data toggle. If DATATOGWREN is high, this bit can be written with the required setting of the data toggle. If DATATOGWREN is low, any value written to this bit is ignored. |
| 8 | Reserved | Reserved |
| 7 | CLRDATATOG | Write a 1 to this bit to reset the endpoint data toggle to 0. |
| 6 | SENTSTALL | This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the TXPKTRDY bit is cleared. You should clear this bit. |
| 5 | SENDSTALL | Write a 1 to this bit to issue a STALL handshake. Clear this bit to terminate the stall condition. Note: This bit has no effect where the endpoint is being used for Isochronous transfers. |
| 4 | FLUSHFIFO | Write a 1 to this bit to flush the next packet to be read from the endpoint Receive FIFO. The FIFO pointer is reset and the RXPKTRDY bit is cleared. Note: FLUSHFIFO has no effect unless RXPKTRDY is set. Also note that, if the FIFO is double-buffered, FLUSHFIFO may need to be set twice to completely clear the FIFO. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-179. Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR) Field Descriptions (continued)**

| Bit | Field | Description |
|---|---|---|
| 3 | DATAERROR | This bit is set when RXPKTRDY is set if the data packet has a CRC or bit-stuff error. It is cleared when RXPKTRDY is cleared. Note: This bit is only valid when the endpoint is operating in ISO mode. In Bulk mode, it always returns zero. |
| 2 | OVERRUN | This bit is set if an OUT packet cannot be loaded into the Receive FIFO. You should clear this bit. Note: This bit is only valid when the endpoint is operating in ISO mode. In bulk mode, it always returns zero. |
| 1 | FIFOFULL | This bit is set when no more packets can be loaded into the Receive FIFO. |
| 0 | RXPKTRDY | This bit is set when a data packet has been received. You should clear this bit when the packet has been unloaded from the Receive FIFO. An interrupt is generated when the bit is set. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.8 Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR)

The control status register for host receive endpoint (USBn_HOST_RXCSR) is a 16-bit register that provides control and status bits for transfers through the currently-selected Rx endpoint when controller assumes the role of a host. There is a TXCSR register for each configured Rx endpoint (not including endpoint 0).

The control status register for host receive endpoint is shown in Figure 25-166 and described in Table 25-180.

**Figure 25-166. Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR)**

| 15 | 14 | 13 | 12 |
|---|---|---|---|
| AUTOCLEAR | AUTOREQ | DMAEN | DISNYET |
| R/W-0-1h | R/W-0-1h | R/W-0-1h | R/W-0-1h |
| **11** | **10** | **9** | **8** |
| DMAMODE | DATATOGWREN | DATATOG | Reserved |
| R/W-0-1h | W-0-1h | R/W-0-1h | R-0h |
| **7** | **6** | **5** | **4** |
| CLRDATATOG | RXSTALL | REQPKT | FLUSHFIFO |
| W-0-1h | R/W-0-1h | R/W-0-1h | W-0-1h |
| **3** | **2** | **1** | **0** |
| DATAERR_NAKTIMEOUT | ERROR | FIFOFULL | RXPKTRDY |
| R-0-1h | R/W-0-1h | R-0-1h | R/W-0-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-180. Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 15 | AUTOCLEAR | DMA Mode: The CPU sets the AUTOCLEAR bit prior to enabling the Rx DMA. CPU Mode: If the CPU sets the AUTOCLEAR bit, then the RXPKTRDY bit will be automatically cleared when a packet of RXMAXP bytes has been unloaded from the Receive FIFO. When packets of less than the maximum packet size are unloaded, RXPKTRDY will have to be cleared manually. |
| 14 | AUTOREQ | If the CPU sets the AUTOREQ bit, then the REQPKT bit will be automatically set when the RXPKTRDY bit is cleared. Note: This bit is automatically cleared when a short packet is received. |
| 13 | DMAEN | Set this bit to enable the DMA request for the Receive endpoints. |
| 12 | DISNYET | Set this bit to disable the sending of NYET handshakes. When set, all successfully received Receive packets are ACKED including at the point at which the FIFO becomes full. Note: This bit only has any effect in high-speed mode, in which mode it should be set for all Interrupt endpoints. |
| 11 | DMAMODE | Always clear this bit to 0. |
| 10 | DATATOGWREN | Write 1 to this bit to enable the DATATOG bit to be written. This bit is automatically cleared once the new value is written to DATATOG. |
| 9 | DATATOG | When read, this bit indicates the current state of the Receive EP data toggle. If DATATOGWREN is high, this bit can be written with the required setting of the data toggle. If DATATOGWREN is low, any value written to this bit is ignored. |
| 8 | Reserved | Reserved |
| 7 | CLRDATATOG | Write a 1 to this bit to reset the endpoint data toggle to 0. |
| 6 | RXSTALL | When a STALL handshake is received, this bit is set and an interrupt is generated. You should clear this bit. |
| 5 | REQPKT | Write a 1 to this bit to request an IN transaction. It is cleared when RXPKTRDY is set. |
| 4 | FLUSHFIFO | Write a 1 to this bit to flush the next packet to be read from the endpoint Receive FIFO. The FIFO pointer is reset and the RXPKTRDY bit is cleared. Note: FLUSHFIFO has no effect unless RXPKTRDY is set. Also note that, if the FIFO is double-buffered, FLUSHFIFO may need to be set twice to completely clear the FIFO. |



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 25-180. Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR) Field Descriptions (continued)**

| Bit | Field | Description |
|---|---|---|
| 3 | DATAERR_NAKTIMEOUT | When operating in ISO mode, this bit is set when RXPKTRDY is set if the data packet has a CRC or bit-stuff error and cleared when RXPKTRDY is cleared. In Bulk mode, this bit will be set when the Receive endpoint is halted following the receipt of NAK responses for longer than the time set as the NAK Limit by the RXINTERVAL register. You should clear this bit to allow the endpoint to continue. |
| 2 | ERROR | The USB controller sets this bit when three attempts have been made to receive a packet and no data packet has been received. You should clear this bit. An interrupt is generated when the bit is set. Note: This bit is only valid when the transmit endpoint is operating in Bulk or Interrupt mode. In ISO mode, it always returns zero. |
| 1 | FIFOFULL | This bit is set when no more packets can be loaded into the Receive FIFO. |
| 0 | RXPKTRDY | This bit is set when a data packet has been received. You should clear this bit when the packet has been unloaded from the Receive FIFO. An interrupt is generated when the bit is set. |

### *25.9.6.2.9 Count 0 Register (USBn_COUNT0)*

The count 0 register (USBn_COUNT0) is a 7-bit read-only register that indicates the number of received data bytes in the Endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while RxPktRdy (bit 0) is set. This register is shown in Figure 25-167 and described in Table 25-181.

**Figure 25-167. Count 0 Register (USBn_COUNT0)**

| 15 | 7 | 6 | 0 |
|---|---|---|---|
| Reserved | | EP0RXCOUNT | |
| R-0h | | R-0-7h | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-181. Count 0 Register (USBn_COUNT0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 15-7 | Reserved | Reserved |
| 6-0 | EP0RXCOUNT | Indicates the number of received data bytes in the endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while RXPKTRDY of USBn_PERI_CSR0 or USBn_HOST_CSR0 is set. |


Copyright © 2011–2016, Texas Instruments Incorporated

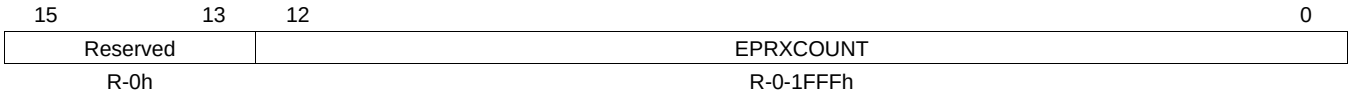
#### 25.9.6.2.10 Receive Count Register (USBn_RXCOUNT)

The receive count register (USBn_RXCOUNT) is a 13-bit read-only register that holds the number of received data bytes in the packet currently in line to be read from the Rx FIFO for endpoints other than endpoint 0.

Note: The value returned changes as the FIFO is unloaded and is only valid while RxPktRdy (bit 0) is set.

The receive count register is shown in Figure 25-168 and described in Table 25-182.

**Figure 25-168. Receive Count Register (USBn_RXCOUNT)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

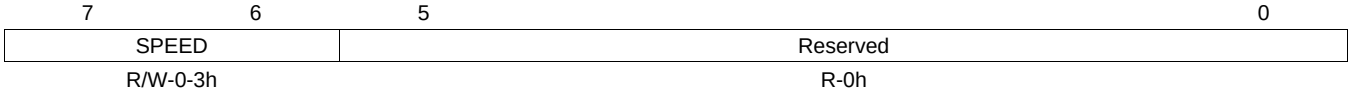
**Table 25-182. Receive Count Register (USBn_RXCOUNT) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 15-13 | Reserved | Reserved |
| 12-0 | EPRXCOUNT | Holds the number of received data bytes in the packet in the Receive FIFO. The value returned changes as the contents of the FIFO change and is only valid while RXPKTRDY of PERI_RXCSR or HOST_RXCSR is set. |

#### 25.9.6.2.11 Type Register (Host Mode Only) (USBn_HOST_TYPE0)

The type register (Host mode only) (USBn_HOST_TYPE0) is an 8-bit register, meaningful only when controller assumes the role of a host, of which only bits 6 and 7 are implemented. These bits should be written with the operating speed of the targeted device. This register is shown in Figure 25-169 and described in Table 25-183.

**Figure 25-169. Type Register (Host Mode Only) (USBn_HOST_TYPE0)**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-183. Type Register (Host Mode Only) (USBn_HOST_TYPE0) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7-6 | SPEED | | Operating Speed of Target Device |
| | | 0h | Illegal |
| | | 1h | High |
| | | 2h | Full |
| | | 3h | Low |
| 5-0 | Reserved | 0 | Reserved |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.12 Transmit Type Register (Host Mode Only) (USBn_HOST_TXTYPE)

The transmit type register (Host mode only) (USBn_HOST_TXTYPE) is an 8-bit register that should be written with the endpoint number to be targeted by the endpoint, the transaction protocol to use for the currently-selected Tx endpoint, and its operating speed. There is a TxType register for each configured Tx endpoint (except Endpoint 0).

The transmit type register is shown in Figure 25-170 and described in Table 25-184.

**Figure 25-170. Transmit Type Register (Host Mode Only) (USBn_HOST_TXTYPE)**

| 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| SPEED | | PROT | | TENDPN | |
| R/W-0-3h | | R/W-0-3h | | R/W-0-Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-184. Transmit Type Register (Host Mode Only) (USBn_HOST_TXTYPE) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7-6 | SPEED | | Operating Speed of Target Device |
| | | 0h | Illegal |
| | | 1h | High |
| | | 2h | Full |
| | | 3h | Low |
| 5-4 | PROT | | Set this to select the required protocol for the transmit endpoint |
| | | 0h | Control |
| | | 1h | Isochronous |
| | | 2h | Bulk |
| | | 3h | Interrupt |
| 3-0 | TENDPN | | Set this value to the endpoint number contained in the transmit endpoint descriptor returned to the USB controller during device enumeration. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.13 NAKLimit0 Register (Host Mode Only) (USBn_HOST_NAKLIMIT0)

The NAKLimit0 register (Host mode only) (USBn_HOST_NAKLIMIT0) is a 5-bit register that sets the number of frames/microframes (high-speed transfers) after which endpoint 0 should timeout on receiving a stream of NAK responses. (Equivalent settings for other endpoints can be made through their TxInterval and RxInterval registers.) The number of frames/microframes selected is 2(m-1) (where m is the value set in the register, valid values 2 – 16). If the host receives NAK responses from the target for more frames than the number represented by the Limit set in this register, the endpoint will be halted. Note: A value of 0 or 1 disables the NAK timeout function.

The NAKLimit0 register is shown in Figure 25-171 and described in Table 25-185.

**Figure 25-171. NAKLimit0 Register (Host Mode Only) (USBn_HOST_NAKLIMIT0)**

| 7 | 5 | 4 | 0 |
|---|---|---|---|
| Reserved | | EP0NAKLIMIT | |
| R-0h | | R/W-0-1Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-185. NAKLimit0 Register (Host Mode Only) (USBn_HOST_NAKLIMIT0) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 7-5 | Reserved | Reserved |
| 4-0 | EP0NAKLIMIT | Sets the number of frames/microframes (high-speed transfers) after which endpoint 0 should time out on receiving a stream of NAK responses. The number of frames/microframes selected is 2(-1) (where m is the value set in the register, valid values 2-16). If the host receives NAK responses from the target for more frames than the number represented by the Limit set in this register, the endpoint will be halted. Note: A value of 0 or 1 disables the NAK timeout function. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.14 Transmit Interval Register (Host Mode Only) (USBn_HOST_TXINTERVAL)

The transmit interval register (Host mode only) (USBn_HOST_TXINTERVAL) is an 8-bit register (meaningful only when controller assumes the role of a host) that, for interrupt and isochronous transfers, defines the polling interval for the currently-selected Tx endpoint. For bulk endpoints, this register sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a TxInterval register for each configured Tx endpoint (except endpoint 0).

The transmit interval register is shown in Figure 25-172 and described in Table 25-186.

**Figure 25-172. Transmit Interval Register (Host Mode Only) (USBn_HOST_TXINTERVAL)**

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| POLINTVL_NAKLIMIT | | | | | | | |
| R/W-0-FFh | | | | | | | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-186. Transmit Interval Register (Host Mode Only) (USBn_HOST_TXINTERVAL) Field Descriptions**

| Bit | Field | Description | | | |
|---|---|---|---|---|---|
| 7-0 | POLINTVL_ NAKLIMIT | For interrupt and isochronous transfers, defines the polling interval for the currently-selected transmit endpoint. For bulk endpoints, sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a transmit interval register for each configured transmit endpoint (except endpoint 0). In each case, the value that is set defines a number of frames/microframes (high-speed transfers), as follows: | | | |
| | | Transfer Type | Speed | Valid values (m) | Interpretation |
| | | Interrupt | Low Speed or Full Speed | 1-255 | Polling interval is m frames |
| | | | | 1-16 | Polling interval is 2(-1) microframes |
| | | Isochronous | Full Speed or High Speed | 1-16 | Polling interval is 2(-1) frames/microframes |
| | | Bulk | Full Speed or High Speed | 2-16 | NAK Limit is 2(-1)frames/microframes |
| | | Note: A value of 0 or 1 disables the NAK timeout function. | | | |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.15 Receive Type Register (Host Mode Only) (USBn_HOST_RXTYPE)

The receive type register (Host mode only) (USBn_HOST_RXTYPE) is an 8-bit register that should be written with the endpoint number to be targeted by the endpoint, the transaction protocol to use for the currently-selected Rx endpoint, and its operating speed. There is a RxType register for each configured Rx endpoint (except endpoint 0).

The receive type register is shown in Figure 25-173 and described in Table 25-187.

**Figure 25-173. Receive Type Register (Host Mode Only) (USBn_HOST_RXTYPE)**

| 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| SPEED | | PROT | | RENDPN | |
| R/W-0-3h | | R/W-0-3h | | R/W-0-Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-187. Receive Type Register (Host Mode Only) (USBn_HOST_RXTYPE) Field Descriptions**

| Bits | Field Name | Value | Description |
|---|---|---|---|
| 7-6 | SPEED | | Operating Speed of Target Device |
| | | 0h | Illegal |
| | | 1h | High |
| | | 2h | Full |
| | | 3h | Low |
| 5-4 | PROT | | Set this to select the required protocol for the transmit endpoint |
| | | 0h | Control |
| | | 1h | Isochronous |
| | | 2h | Bulk |
| | | 3h | Interrupt |
| 3-0 | RENDPN | | Set this value to the endpoint number contained in the Receive endpoint descriptor returned to the USB controller during device enumeration |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.16 Receive Interval Register (Host Mode Only) (USBn_HOST_RXINTERVAL)

The receive interval register (Host mode only) (USBn_HOST_RXINTERVAL) is an 8-bit register (only meaningful in host mode only) that, for Interrupt and Isochronous transfers, defines the polling interval for the currently-selected Rx endpoint. For bulk endpoints, this register sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a RxInterval register for each configured Rx endpoint (except endpoint 0).

The receive interval register is shown in Figure 25-174 and described in Table 25-188.

**Figure 25-174. Receive Interval Register (Host Mode Only) (USBn_HOST_RXINTERVAL)**

| 7 | 0 |
|---|---|
| POLINTVL_NAKLIMIT | |
| R/W-0-FFh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-188. Transmit Interval Register (Host Mode Only) (USBn_HOST_RXINTERVAL) Field Descriptions**

| Bit | Field | Description | | | |
|---|---|---|---|---|---|
| 7-0 | POLINTVL_ NAKLIMIT | For interrupt and isochronous transfers, defines the polling interval for the currently-selected transmit endpoint. For bulk endpoints, sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a transmit interval register for each configured transmit endpoint (except endpoint 0). In each case, the value that is set defines a number of frames/microframes (high-speed transfers), as follows: | | | |
| | | Transfer Type | Speed | Valid values (m) | Interpretation |
| | | Interrupt | Low Speed or Full Speed | 1-255 | Polling interval is m frames |
| | | | High Speed | 1-16 | Polling interval is 2(-1) microframes |
| | | Isochronous | Full Speed or High Speed | 1-16 | Polling interval is 2(-1) frames/microframes |
| | | Bulk | Full Speed or High Speed | 2-16 | NAK Limit is 2(-1)frames/microframes |
| | | Note: A value of 0 or 1 disables the NAK timeout function. | | | |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.6.2.17 Configuration Data Register (USBn_CONFIGDATA)

The configuration data register (USBn_CONFIGDATA) is an 8-bit read-only register that returns information about the selected core configuration. This register information can only be accessed from Indexed Region and INDEX has to be programmed, with zero, selecting endpoint 0. This register is shown in Figure 25-175 and described in Table 25-189.

**Figure 25-175. Configuration Data Register (USBn_CONFIGDATA)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MPRXE | MPTXE | BIGENDIAN | HBRXE | HBTXE | DYNFIFO | SOFTCONE | UTMIDATAWIDTH |
| R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h | R-0-1h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-189. Configuration Data Register (USBn_CONFIGDATA) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7 | MPRXE | | Indicates automatic amalgamation of bulk packets. (INDEX register needs to be set to zero.) |
| | | 0 | Automatic amalgamation of bulk packets is not selected. |
| | | 1 | Automatic amalgamation of bulk packets is selected. |
| 6 | MPTXE | | Indicates automatic splitting of bulk packets. (INDEX register needs to be set to zero.) |
| | | 0 | Automatic splitting of bulk packets is not selected. |
| | | 1 | Automatic splitting of bulk packets is selected. |
| 5 | BIGENDIAN | | Indicates endian ordering. (INDEX register needs to be set to zero.) |
| | | 0 | Little-endian ordering is selected. |
| | | 1 | Big-endian ordering is selected. |
| 4 | HBRXE | | Indicates high-bandwidth Rx ISO endpoint support. (INDEX register needs to be set to zero.) |
| | | 0 | High-bandwidth Rx ISO endpoint support is not selected. |
| | | 1 | High-bandwidth Rx ISO endpoint support is selected. |
| 3 | HBTXE | | Indicates high-bandwidth Tx ISO endpoint support. (INDEX register needs to be set to zero.) |
| | | 0 | High-bandwidth Tx ISO endpoint support is not selected. |
| | | 1 | High-bandwidth Tx ISO endpoint support is selected. |
| 2 | DYNFIFO | | Indicates dynamic FIFO sizing. (INDEX register needs to be set to zero.) |
| | | 0 | Dynamic FIFO sizing option is not selected. |
| | | 1 | Dynamic FIFO sizing option is selected. |
| 1 | SOFTCONE | | Indicates soft connect/disconnect. (INDEX register needs to be set to zero.) |
| | | 0 | Soft connect/disconnect option is not selected. |
| | | 1 | Soft connect/disconnect option is selected. |
| 0 | UTMIDATAWIDTH | | Indicates selected UTMI data width. (INDEX register needs to be set to zero.) |
| | | 0 | 8 bits |
| | | 1 | 16 bits |


Copyright © 2011–2016, Texas Instruments Incorporated

## 25.9.7 FIFOs

This address range provides access to the endpoint FIFOs. Register offset addresses for this block is 20h to 5Fh.

### 25.9.7.1 Transmit and Receive FIFO Register for Endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15)

This address range provides 16 addresses for CPU access to the FIFOs for each endpoint. Writing to these addresses loads data into the TxFIFO for the corresponding endpoint. Reading from these addresses unloads data from the RxFIFO for the corresponding endpoint. The address range is 20h – 5Fh and the FIFOs are located on 32-bit double-word boundaries (endpoint 0 at 20h, endpoint 1 at 24h ... Endpoint 15 at 5Ch).

Note 1: Transfers to and from FIFOs may be 8-bit, 16-bit or 32-bit as required, and any combination of access is allowed provided the data accessed is contiguous. However, all the transfers associated with one packet must be of the same width so that the data is consistently byte-, word- or double-word-aligned. The last transfer may however contain fewer bytes than the previous transfers in order to complete an odd-byte or odd-word transfer.

Note 2: Depending on the size of the FIFO and the expected maximum packet size, the FIFOs support either single-packet or doublepacket buffering. However, burst writing of multiple packets is not supported as flags need to be set after each packet is written.

Note 3: Following a STALL response or a Tx strike out error on dndpoint 1 – 15, the associated FIFO is completely flushed.

The transmit and receive FIFO register for endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15) is shown in Figure 25-176 and described in Table 25-190.

**Figure 25-176. Transmit and Receive FIFO Register for Endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15)**

| 31 | 0 |
|---|---|
| DATA | |
| R/W-0-FFFF FFFFh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-190. FIFOs: Transmit and Receive FIFO Register for Endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 31-0 | DATA | Writing to these addresses loads data into the Transmit FIFO for the corresponding endpoint. Reading from these addresses unloads data from the Receive FIFO for the corresponding endpoint. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.7.2 Additional Control and Configuration Registers

Table 25-191 lists the additional Control and Configuration registers.

**Table 25-191. Additional Control and Configuration Registers**

| Core Address Offset | Control and Configuration Registers |
|---|---|
| 60h | Device Control Register |
| 62h | Transmit Endpoint FIFO Size Register |
| 63h | Receive Endpoint FIFO Size Register |
| 64h | Transmit Endpoint FIFO Address Register |
| 66h | Receive Endpoint FIFO Address Register |
| 6Ch | Hardware Version Register |

#### *25.9.7.2.1 Device Control Register (USBn_DEVCTL)*

The device control register (USBn_DEVCTL) is an 8-bit register that is used to select whether the USB controller is operating in peripheral mode or in host mode, and for controlling and monitoring the USB VBus line. If the PHY is suspended no PHY clock is received and the VBus is not sampled. This register is shown in Figure 25-177 and described in Table 25-192.

**Figure 25-177. Device Control Register (USBn_DEVCTL)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BDEVICE | FSDEV | LSDEV | VBUS | | HOSTMODE | HOSTREQ | SESSION |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-192. Device Control Register (USBn_DEVCTL) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7 | BDEVICE | | This read-only bit indicates whether the USB controller is operating as the 'A' device or the 'B' device. |
| | | 0 | A device |
| | | 1 | B device |
| 6 | FSDEV | | This read-only bit is set when a full-speed or high-speed device has been detected being connected to the port (high-speed devices are distinguished from full-speed by checking for high-speed chirps when the device is reset). Only valid in Host mode. |
| 5 | LSDEV | | This read-only bit is set when a low-speed device has been detected being connected to the port. Only valid in Host mode. |
| 4-3 | VBUS | | These read-only bits encode the current VBus level as follows: |
| | | 0h | Below session end |
| | | 1h | Above session dnd, below AValid |
| | | 2h | Above AValid, below VBusValid |
| | | 3h | Above VBusValid |
| 2 | HOSTMODE | | This read-only bit is set when the USB controller is acting as a Host. |
| 1 | HOSTREQ | | When set, the USB controller initiates the Host negotiation when suspend mode is entered. It is cleared when Host negotiation is completed. ('B' device only) |
| 0 | SESSION | | When operating as an 'A' device, you must set or clear this bit start or end a session. When operating as a 'B' device, this bit is set/cleared by the USB controller when a session starts/ends. You must also set this bit to initiate the session request protocol. When the USB controller is in suspend mode, you may clear the bit to perform a software disconnect. A special software routine is required to perform SRP. Details will be made available in a later document version. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.7.2.2 Transmit Endpoint FIFO Size Register (USBn_TXFIFOSZ)

The allocation of FIFO space to the different endpoints requires the specification for each Tx endpoint:

- The start address of the FIFO within the RAM block
- The maximum size of packet to be supported
- Whether double-buffering is required

The transmit endpoint FIFO size register (USBn_TXFIFOSZ) defines the amount of space that needs to be allocated to the FIFO in either single or double buffering context.

For more information, see the *Dynamic FIFO Allocation* section.

The transmit endpoint FIFO size register is shown in Figure 25-178 and described in Table 25-193.

**Figure 25-178. Transmit Endpoint FIFO Size Register (USBn_TXFIFOSZ)**

| 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | | DPB | SZ | |
| R-0h | | | R-0-1h | R-0-Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-193. Transmit Endpoint FIFO Size Register (USBn_TXFIFOSZ) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7-5 | Reserved | | Reserved |
| 4 | DPB | | Double packet buffering enable. |
| | | 0 | Single packet buffering is supported. |
| | | 1 | Double packet buffering is enabled |
| 3-0 | SZ | | Maximum packet size to be allowed (before any splitting within the FIFO of Bulk packets prior to transmission). If m = SZ, the FIFO size is calculated as 2(m+3) for single packet buffering and 2(m+4) for dual packet buffering. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.7.2.3 Receive Endpoint FIFO Size Register (USBn_RXFIFOSZ)

The allocation of FIFO space to the different endpoints requires the specification for each Rx endpoint

- The start address of the FIFO within the RAM block
- The maximum size of packet to be supported
- Whether double-buffering is required

The receive endpoint FIFO size register (USBn_RXFIFOSZ) defines the amount of space that needs to be allocated to the FIFO in either single or double buffering context.

For more information, see the *Dynamic FIFO Allocation* section.

The receive endpoint FIFO size register is shown in Figure 25-179 and described in Table 25-194.

**Figure 25-179. Receive Endpoint FIFO Size Register (USBn_RXFIFOSZ)**

| 7 | 6 | 5 | 4 | 3 | 0 |
|---|---|---|---|---|---|
| Reserved | | | DPB | SZ | |
| 0h | | | 0-1h | 0-Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-194. Receive Endpoint FIFO Size Register (USBn_RXFIFOSZ) Field Descriptions**

| Bit | Field | Value | Description |
|---|---|---|---|
| 7-5 | Reserved | | Reserved |
| 4 | DPB | | Double packet buffering enable. |
| | | 0 | Single packet buffering is supported. |
| | | 1 | Double packet buffering is enabled |
| 3-0 | SZ | | Maximum packet size to be allowed (before any splitting within the FIFO of Bulk packets prior to transmission). If m = SZ, the FIFO size is calculated as 2(m+3) for single packet buffering and 2(m+4) for dual packet buffering. |

#### 25.9.7.2.4 Transmit Endpoint FIFO Address Register (USBn_TXFIFOADDR)

The allocation of FIFO space to the different endpoints requires the specification for each Tx endpoint:

- The start address of the FIFO within the RAM block
- The maximum size of packet to be supported
- Whether double-buffering is required

The transmit endpoint FIFO address register (USBn_TXFIFOADDR) the coding for the start address of the FIFO. Note that the 1st 64 bytes of the FIFO RAM is reserved for the use of endpoint 0. Allocation of FIFO should exclude this reserved location.

The transmit endpoint FIFO address register is shown in Figure 25-180 and described in Table 25-195.

**Figure 25-180. Transmit Endpoint FIFO Address Register (USBn_TXFIFOADDR)**

| 15 | 13 | 12 | 0 |
|---|---|---|---|
| Reserved | | ADDR | |
| 0h | | 0-1FFFh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-195. Transmit Endpoint FIFO Address Register (USBn_TXFIFOADDR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 15-13 | Reserved | Reserved |
| 12-0 | ADDR | Start address of endpoint FIFO in units of 8 bytes If m = ADDR, then the start address is 8 × m |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.7.2.5 Receive Endpoint FIFO Address Register (USBn_RXFIFOADDR)

The allocation of FIFO space to the different endpoints requires the specification for each Rx endpoint:

- The start address of the FIFO within the RAM block
- The maximum size of packet to be supported
- Whether double-buffering is required

The receive endpoint FIFO address register (USBn_RXFIFOADDR) the coding for the start address of the FIFO. Note that the 1st 64 bytes of the FIFO RAM is reserved for the use of endpoint 0. Allocation of FIFO should exclude this reserved location.

The receive endpoint FIFO address register is shown in Figure 25-181 and described in Table 25-196.

**Figure 25-181. Receive Endpoint FIFO Address Register (USBn_RXFIFOADDR)**

| 15 13 | 12 0 |
|---|---|
| Reserved | ADDR |
| 0h | 0-1FFFh |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-196. Receive Endpoint FIFO Address Register (USBn_RXFIFOADDR) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 15-13 | Reserved | Reserved |
| 12-0 | ADDR | Start address of endpoint FIFO in units of 8 bytes If m = ADDR, then the start address is 8 × m |

#### 25.9.7.2.6 Hardware Version Register (USBn_HWVERS)

The hardware version register (USBn_HWVERS) is a 16-bit read-only register that returns information about the version of the RTL from which the core hardware was generated, in particular the RTL version number (REVMAJ.REVMIN). This register is shown in Figure 25-182 and described in Table 25-197.

**Figure 25-182. Hardware Version Register (USBn_HWVERS)**

| 15 | 14 10 | 9 0 |
|---|---|---|
| RC | REVMAJ | REVMIN |
| 0-1h | 0-1Fh | 0-3E7h |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-197. Hardware Version Register (USBn_HWVERS) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 15 | RC | Set to 1 if RTL is used from a Release Candidate, rather than from a full release of the core. |
| 14-10 | REVMAJ | Major version of RTL. Range is 0-31. |
| 9-0 | REVMIN | Minor version of RTL. Range is 0-999. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.7.3 Target Endpoint Control Registers

Table 25-198 lists the target endpoint control registers. These registers provide target function and hub address details for each of the endpoints. Register addresses consumed by this block is 0x0080 to 0x00FF.

**Table 25-198. Target Endpoint Control Registers**

| Core Address Offset | Target Endpoint Control Registers |
|---|---|
| 80h | Transmit Function Address Register |
| 82h | Transmit Hub Address Register |
| 83h | Transmit Hub Port Register |
| 84h | Receive Function Address Register |
| 86h | Receive Hub Address Register |
| 87h | Receive Hub Port Register |

#### *25.9.7.3.1 Transmit Function Address Register (USBn_TXFUNCADDRm)*

The transmit function address register (USBn_TXFUNCADDRm) is a 7-bit read/write register (meaningful in host mode operation only) that records the address of the target function that is to be accessed through the associated endpoint (EPn). USBn_ TXFUNCADDR needs to be defined for each Tx endpoint that is used.

Note: USBn_TXFUNCADDRm needs to be defined for endpoint 0. This register, together with the USBn_TXHUBADDRm and USBn_TXHUBPORTm registers described below, allow the allocation of multiple devices to the USB controller endpoints.

The transmit function address register is shown in Figure 25-183 and described in Table 25-199.

**Figure 25-183. Transmit Function Address Register (USBn_TXFUNCADDRm)**

| 7 | 6 | 0 |
|---|---|---|
| Reserved | FUNCADDR | |
| 0h | 0-7Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-199. Transmit Function Address (USBn_TXFUNCADDRm) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 7 | Reserved | Reserved |
| 6-0 | FUNCADDR | Address of target function |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.7.3.2 Transmit Hub Address Register (USBn_TXHUBADDRm)

The transmit hub address register (USBn_TXHUBADDRm) is an 8-bit read/write register which, like USBn_TXHUBPORTm, only need to be written where a full- or low-speed device is connected to Tx endpoint EPn via a high-speed USB 2.0 hub that carries out the necessary transaction translation to convert between high-speed transmission and full-/low-speed transmission. In such circumstances:

- The lower 7 bits should record the address of this USB 2.0 hub
- The top bit should record whether the hub has multiple transaction translators (set to '0' if single transaction translator; set to '1' if multiple transaction translators).

Note: If endpoint 0 is connected to a hub, then USBn_TXHUBADDRm need to be defined for this endpoint.

The transmit hub address register is shown in Figure 25-184 and described in Table 25-200.

**Figure 25-184. Transmit Hub Address Register (USBn_TXHUBADDRm)**

| 7 | 6 | 0 |
|---|---|---|
| Reserved | FUNCADDR | |
| 0h | 0-7Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-200. Transmit Hub Address Register (USBn_TXHUBADDRm) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 7 | MULT_TRANS | Set to 1 if hub has multiple transaction translators. Cleared to 0 if only single transaction translator is available. |
| 6-0 | HUBADDR | Address of hub |

#### 25.9.7.3.3 Transmit Hub Port Register (USBn_TXHUBPORTm)

The transmit hub port register (USBn_TXHUBPORTm) (meaningful in host mode operation only) only needs to be written where a full- or low-speed device is connected to Tx Endpoint EPn via a high-speed USB 2.0 hub which carries out the necessary transaction translation. In such circumstances, these 7-bit read/write registers need to be used to record the port of that USB 2.0 hub through which the target associated with the endpoint is accessed.

Note: If Endpoint 0 is connected to a hub, then USBn_TXHUBPORT needs to be defined for this endpoint.

The transmit hub port register is shown in Figure 25-185 and described in Table 25-201.

**Figure 25-185. Transmit Hub Port Register (USBn_TXHUBPORTm)**

| 7 | 6 | 0 |
|---|---|---|
| Reserved | FUNCADDR | |
| 0h | 0-7Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-201. Transmit Hub Port Register (USBn_TXHUBPORTm) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 7 | Reserved | Reserved |
| 6-0 | HUBPORT | Port number of the hub |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.7.3.4 Receive Function Address Register (USBn_RXFUNCADDRm)

The receive function address register (USBn_RXFUNCADDRm) is a 7-bit read/write register (meaningful in host mode operation only) that record the address of the target function that is to be accessed through the associated endpoint (EPn). USBn_ RXFUNCADDR needs to be defined for each Rx endpoint that is used.

Note: USBn_RXFUNCADDRm need to be defined for Endpoint 0. This register (together with the USBn_RXHUBADDRm and USBn_RXHUBPORTm registers described below) allow the allocation of multiple devices to the USB controller endpoints.

The receive function address register is shown in Figure 25-186 and described in Table 25-202.

**Figure 25-186. Receive Function Address Register (USBn_RXFUNCADDRm)**

| 7 | 6 | 0 |
|---|---|---|
| Reserved | FUNCADDR | |
| 0h | 0-7Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-202. Receive Function Address Register (USBn_RXFUNCADDRm) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 7 | Reserved | Reserved |
| 6-0 | FUNCADDR | Address of target function. |

#### 25.9.7.3.5 Receive Hub Address Register (USBn_RXHUBADDRm)

The receive hub address register (USBn_RXHUBADDRm) is an 8-bit read/write register (meaningful in host mode operation only) which, like USBn_RXHUBPORTm, only needs to be written where a full- or low-speed device is connected to Rx Endpoint EPn via a high-speed USB 2.0 hub which carries out the necessary transaction translation to convert between high-speed transmission and full-/low-speed transmission. In such circumstances:

- The lower 7 bits should record the address of this USB 2.0 hub
- The top bit should record whether the hub has multiple transaction translators (set to ‘0’ if single transaction translator; set to ‘1’ if multiple transaction translators).

Note: If Endpoint 0 is connected to a hub, then USBn_RXHUBADDRm need to be defined for this endpoint.

The receive hub address register is shown in Figure 25-187 and described in Table 25-203.

**Figure 25-187. Receive Hub Address Register (USBn_RXHUBADDRm)**

| 7 | 6 | 0 |
|---|---|---|
| Reserved | FUNCADDR | |
| 0h | 0-7Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-203. Receive Hub Address Register (USBn_RXHUBADDRm) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 7 | MULT_TRANS | Set to 1 if hub has multiple transaction translators. Cleared to 0 if only single transaction translator is available. |
| 6-0 | HUBADDR | Address of HUB. |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 25.9.7.3.6 Receive Hub Port Register (USBn_RXHUBPORTm)

The receive hub port register (USBn_RXHUBPORTm) (meaningful in host mode operation only) only needs to be written where a full- or low-speed device is connected to Rx Endpoint EPn via a high-speed USB 2.0 hub which carries out the necessary transaction translation. In such circumstances, these 7-bit read/write registers need to be used to record the port of that USB 2.0 hub through which the target associated with the endpoint is accessed.

Note: If Endpoint 0 is connected to a hub, then USBn_RXHUBPORT needs to be defined for this endpoint.

The receive hub port register is shown in Figure 25-188 and described in Table 25-204.

**Figure 25-188. Receive Hub Port Register (USBn_RXHUBPORTm)**

| 7 | 6 | 0 |
|---|---|---|
| Reserved | FUNCADDR | |
| 0h | 0-7Fh | |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 25-204. Receive Hub Port Register (USBn_RXHUBPORTm) Field Descriptions**

| Bit | Field | Description |
|---|---|---|
| 7 | Reserved | Reserved |
| 6-0 | HUBPORT | Port number of HUB. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 25.9.7.4 Non-Indexed Endpoint Control/Status Registers

The registers available at indexed region, offset 10h–1Fh, are accessible independently of the setting of the Index register. The Non-Indexed region has dedicated block of register space for each endpoint with memory block between 100h–10Fh are reserved for endpoint 0 registers; memory block between 110h–11Fh are reserved for endpoint 1 registers; memory block between 120h–12Fh are reserved for endpoint 2 registers; and so on.

Table 25-205 and Table 25-206 display the set of endpoint registers that are accessible from their dedicated block of memory in both peripheral and host mode, respectively. Note that the definitions of these register is discussed in detail in Section 25.9.6.2.

**Table 25-205. Peripheral Mode Configuration and Status Register Mapping where n=0,1 (USB0 or USB1) and m=1-15 (endpoint number)**

| Offset | Name | Description (within Indexed Region) |
|---|---|---|
| 100h + 10h * m | USBn_TXMAXPm | See USBn_TXMAXP Reg Def |
| 102h | USBn_ PERI_CSR0 | See USBn_PERI_CSR0 Reg Def |
| 102h + 10h * m | USBn_PERI_TXCSRm | See USBn_PERI_TXCSR Reg Def |
| 104h + 10h * m | USBn_RXMAXPm | See USBn_RXMAXP Reg Def |
| 106h + 10h * m | USBn_ PERI_RXCSRm | See USBn_PERI_RXCSR Reg Def |
| 108h | USBn_COUNT0 | See USBn_COUNT0 Reg Def |
| 108h + 10h * m | USBn_ RXCOUNTm | See USBn_RXCOUNT Reg Def |
| 10Fh | USBn_CONFIGDATA | See USBn_CONFIGDATA Reg Def |

**Table 25-206. Host Mode Configuration and Status Register Mapping where n=0,1 (USB0 or USB1) and m=1-15 (endpoint number)**

| Offset | Name | Description |
|---|---|---|
| 100h +10h * m | USBn_TXMAXPm | See USBn_TXMAXP Reg Def |
| 102h | USBn_HOST_CSR0 | See USBn_HOST_CSR0 Reg Def |
| 102h + 10h * m | USBn_HOST_TXCSRm | See USBn_HOST_TXCSR Reg Def |
| 104h + 10h * m | USBn_RXMAXPm | See USBn_RXMAXP Reg Def |
| 106h + 10h * m | USBn_HOST _RXCSRm | See USBn_HOST_RXCSR Reg Def |
| 108h | USBn_COUNT0 | See USBn_COUNT0 Reg Def |
| 108h + 10h * m | USBn_RXCOUNTm | See USBn_RXCOUNT Reg Def |
| 10Ah | USBn_HOST_TYPE0 | See USBn_HOST_TYPE0 Reg Def |
| 10Ah + 10h * m | USBn_HOST_TXTYPEm | See USBn_HOST_TXTYPE Reg Def |
| 10Bh | USBn_HOST_NAKLIMIT0 | See USBn_HOST_NAKLIMIT0 Reg Def |
| 10Bh + 10h * m | USBn_HOST_TXINTERVALm | See USBn_HOST_TXINTERVAL Def |
| 10Ch | USBn_HOST_TYPE0 | See USBn_HOST_TYPE0 Reg Def |
| 10Ch + 10h * m | USBn_HOST_RXTYPEm | See USBn_HOST_RXTYPE Reg Def |
| 10Dh + 10h * m | USBn_HOST_RXINTERVALm | See USBn_HOST_RXINTERVAL Def |
| 10Fh | USBn_CONFIGDATA | See USBn_CONFIGDATA Reg Def |


Copyright © 2011–2016, Texas Instruments Incorporated

# Watchdog Timer

This chapter describes the watchdog timer.

**Topic** **Page**

**26.1 Introduction** ..... **3889**
**26.2 Architecture** ..... **3890**
**26.3 Low-Level Programming Model** ..... **3896**
**26.4 Watchdog Timer Registers** ..... **3898**


Copyright © 2011–2016, Texas Instruments Incorporated

## 26.1 Introduction

### 26.1.1 Overview

The watchdog timer is an upward counter capable of generating a pulse on the reset pin and an interrupt to the device system modules following an overflow condition. The watchdog timer serves resets to the PRCM module and serves watchdog interrupts to the host subsystem. The reset of the PRCM module causes warm reset of the device.

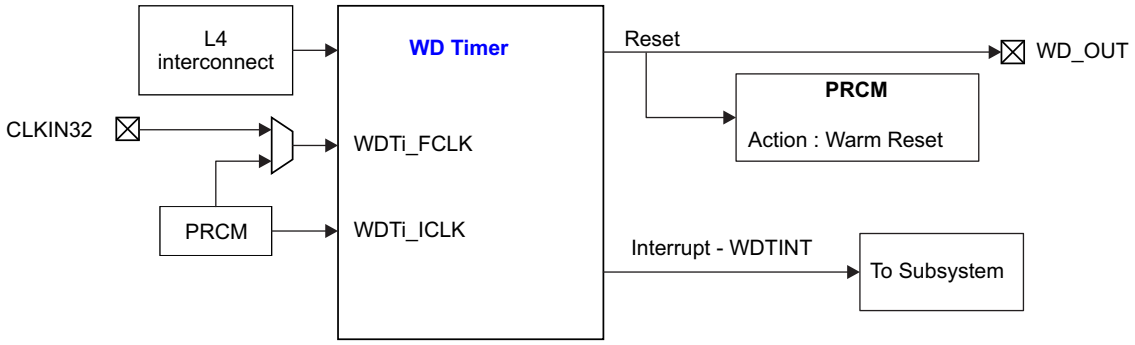
The watchdog timer can be accessed, loaded, and cleared by registers through the L4 interface. The watchdog timer has the 32-kHz clock for its timer clock input.

The watchdog timer connects to a single target agent port on the L4 interconnect. The default state of the watchdog timer is enabled and not running.

### 26.1.2 Functional Block Diagram

Figure 26-1 shows a block diagram of the watchdog timer.

**Figure 26-1. Watchdog Timer Block Diagram**




### 26.1.3 Features

The main features of the watchdog timer controllers are:

- L4 slave interface support:
  - 32-bit data bus width
  - 32-/16-bit access supported
  - 8-bit access not supported
  - 11-bit address bus width
  - Burst mode not supported
  - Write nonposted transaction mode only
- Free-running 32-bit upward counter
- Programmable divider clock source ($2^n$ where n = 0-7)
- On-the-fly read/write register (while counting)
- Subset programming model of the timer
- The watchdog timers are reset either on power-on or after a warm reset before they start counting.
- Reset or interrupt actions when a timer overflow condition occurs
- The watchdog timer generates a reset or an interrupt in its hardware integration.

### 26.1.4 Watchdog Timer Environment

The watchdog timers are accessible through the L4 interface.


Copyright © 2011–2016, Texas Instruments Incorporated

## 26.2 Architecture

### 26.2.1 Power Management

There are two clock domains in the watchdog timers:

- Functional clock domain: WDTi_FCLK is a 32 kHz watchdog timer functional clock. It is used to clock the watchdog timer internal logic.
- Interface clock domain: WDTi_ICLK is a 125 MHz watchdog timer interface clock. It is used to synchronize the watchdog timer L4 port to the L4 interconnect. All accesses from the interconnect are synchronous to WDTi_ICLK.

In this device, the watchdog timer clocks are always On. The clocks cannot be turned off, if the watchdog timer is not being used.

### 26.2.2 Interrupts

Table 26-1 list the event flags, and their masks, that cause module interrupts.

**Table 26-1. Watchdog Timer Events**

| Event Flag | Event Mask | Mapping | Comments |
|---|---|---|---|
| WDT_WIRQSTAT[0] EVENT_OVF | WDT_WIRQENSET/WDT_WIRQENCLR[0] OVF_IT_ENA | WDTINT | Watchdog timer overflow |
| WDT_WIRQSTAT[1] EVENT_DLY | WDT_WIRQENSET/WDT_WIRQENCLR[1] DLY_IT_ENA | WDTINT | Watchdog delay value reached |

### 26.2.3 General Watchdog Timer Operation

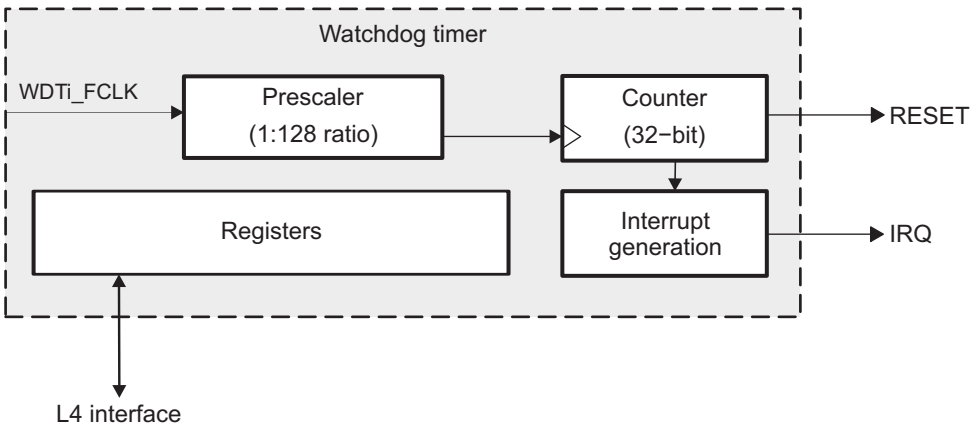
The watchdog timers are based on an upward 32-bit counter coupled with a prescaler. The counter overflow is signaled through two independent signals: a simple reset signal and an interrupt signal, both active low. Figure 26-2 is a functional block diagram of the watchdog timer.

The interrupt generation mechanism is controlled through the WDT_WIRQENSET/WDT_WIRQENCLR and WDT_WIRQSTAT registers.

The prescaler ratio can be set from 1 to 128 by accessing the WDT_WCLR[4:2] PTV bit field and the WDT_WCLR[5] PRE bit of the watchdog control register (WDT_WCLR).

The current timer value can be accessed on-the-fly by reading the watchdog timer counter register (WDT_WCRR), can be modified by accessing the watchdog timer load register (WDT_WLDR) (no on-the-fly update), or can be reloaded by following a specific reload sequence on the watchdog timer trigger register (WDT_WTGR). A start/stop sequence applied to the watchdog timer start/stop register (WDT_WSPR) can start and stop the watchdog timers.

**Figure 26-2. 32-Bit Watchdog Timer Functional Block Diagram**





Copyright © 2011–2016, Texas Instruments Incorporated

### 26.2.4 Reset Context

The watchdog timers are enabled after reset. Table 26-2 lists the default reset values of the two watchdog timer load registers (the WDT_WLDR) and prescaler ratios (the WDT_WCLR[4:2] PTV bit field). To get these values, software must read the corresponding WDT_WCLR[4:2] PTV bit field and the 32-bit register to retrieve the static configuration of the module.

**Table 26-2. Count and Prescaler Default Reset Values**

| Timer | WDT_WLDR Reset Value | PTV Reset Value |
|---|---|---|
| WDT | FFFF FFBEh | 0 |

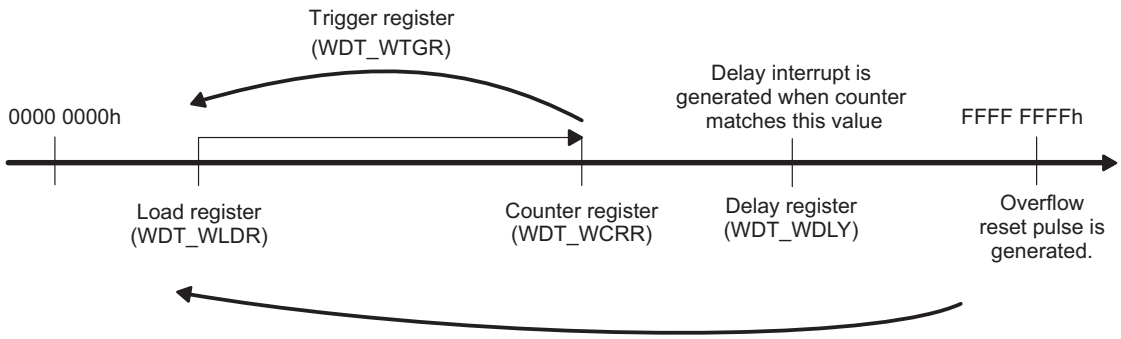
### 26.2.5 Overflow/Reset Generation

When the watchdog timer counter register (WDT_WCRR) overflows, an active-low reset pulse is generated to the PRCM module. This RESET pulse causes the PRCM module to generate global WARM reset of the device. It is also driven out of the device through the WD_OUT pin. This pulse is one prescaled timer clock cycle wide and occurs at the same time as the timer counter overflow.

After reset generation, the counter is automatically reloaded with the value stored in the watchdog load register (WDT_WLDR) and the prescaler is reset (the prescaler ratio remains unchanged). When the reset pulse output is generated, the timer counter begins incrementing again.

Figure 26-3 shows a general functional view of the watchdog timers.

**Figure 26-3. Watchdog Timers General Functional View**




### 26.2.6 Prescaler Value/Timer Reset Frequency

Each watchdog timer is composed of a prescaler stage and a timer counter.

The timer rate is defined by the following values:

- Value of the prescaler fields (the WDT_WCLR[5] PRE bit and the WDT_WCLR[4:2] PTV bit field)
- Value loaded into the timer load register (WDT_WLDR)

The prescaler stage is clocked with the timer clock and acts as a clock divider for the timer counter stage. The ratio is managed by accessing the ratio definition field (the WDT_WCLR[4:2] PTV bit field) and is enabled with the WDT_WCLR[5] PRE bit.

Table 26-3 lists the prescaler clock ratio values.



Copyright © 2011–2016, Texas Instruments Incorporated

**Table 26-3. Prescaler Clock Ratio Values**

| WDT_WCLR[5] PRE | WDT_WCLR[4:2] PTV | Clock Divider (PS) |
|---|---|---|
| 0 | X | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 2 |
| 1 | 2 | 4 |
| 1 | 3 | 8 |
| 1 | 4 | 16 |
| 1 | 5 | 32 |
| 1 | 6 | 64 |
| 1 | 7 | 128 |

Thus the watchdog timer overflow rate is expressed as:

*OVF_Rate = (FFFF FFFFh – WDT_WLDR + 1) × (wd-functional clock period) × PS*

where wd-functional clock period = 1/(wd-functional clock frequency) and PS = $2^{(PTV)}$

**CAUTION**

Internal resynchronization causes some latency in any software write to WDT_WSPR before WDT_WSPR is updated with the programmed value:

1.5 × functional clock cycles ≤ write_WDT_WSPR_latency ≤ 2.5 × functional clock cycles

Remember to consider this latency whenever the watchdog timer must be started or stopped.

For example, for a timer clock input of 32 kHz with a prescaler ratio value of 1 (clock divided by 2) and WDT_WCLR[5] PRE = 1 (clock divider enabled), the reset period is as listed in Table 26-4.

**Table 26-4. Reset Period Examples**

| WDT_WLDR Value | Reset Period |
|---|---|
| 0000 0000h | 74 h 56 min |
| FFFF 0000h | 4 s |
| FFFF FFF0h | 1 ms |
| FFFF FFFFh | 62.5 us |

**CAUTION**

- Ensure that the reloaded value allows the correct operation of the application. When a watchdog timer is enabled, software must periodically trigger a reload before the counter overflows. Hence, the value of the WDT_WLDR[31:0] bit field must be chosen according to the ongoing activity preceding the watchdog reload.
- Due to design reasons, WDT_WLDR[31:0] = FFFF FFFFh is a special case, although such a value of WDT_WLDR is meaningless. When WDT_WLDR is programmed with the overflow value, a triggering event generates a reset/interrupt one functional clock cycle later, even if the watchdog timer is stopped.


Copyright © 2011–2016, Texas Instruments Incorporated

Table 26-5 lists the default reset periods for the watchdog timers.

**Table 26-5. Default Watchdog Timer Reset Periods**

| Watchdog Timers | Clock Source | Default Reset Period |
|---|---|---|
| WDT | 32 kHz | 2 s |

### 26.2.7 Triggering a Timer Reload

To reload the timer counter and reset the prescaler before reaching overflow, a reload command is executed by accessing the watchdog timer trigger register (WDT_WTGR) using a specific reload sequence.

The specific reload sequence is performed whenever the written value on the WDT_WTGR register differs from its previous value. In this case, reload is executed in the same way as an overflow autoreload, but without the generation of a reset pulse.

The timer counter is loaded with the value of the watchdog timer load register (the WDT_WLDR[31:0] TIMER_LOAD bit field), and the prescaler is reset.

### 26.2.8 Start/Stop Sequence for Watchdog Timers (Using the WDT_WSPR Register)

To start and stop a watchdog timer, access must be made through the start/stop register (WDT_WSPR) using a specific sequence.

To disable the timer, follow this sequence:

1. Write XXXX AAAAh in WDT_WSPR.
2. Write XXXX 5555h in WDT_WSPR.

To enable the timer, follow this sequence:

1. Write XXXX BBBBh in WDT_WSPR.
2. Write XXXX 4444h in WDT_WSPR.

All other write sequences on the WDT_WSPR register have no effect on the start/stop feature of the module.

### 26.2.9 Modifying Timer Count/Load Values and Prescaler Setting

To modify the timer counter value (the WDT_WCRR register), prescaler ratio (the WDT_WCLR[4:2] PTV bit field), delay configuration value (the WDT_WDLY[31:0] DLY_VALUE bit field), or the load value (the WDT_WLDR[31:0] TIMER_LOAD bit field), the watchdog timer must be disabled by using the start/stop sequence (the WDT_WSPR register).

After a write access, the load register value and prescaler ratio registers are updated immediately, but new values are considered only after the next consecutive counter overflow or after a new trigger command (the WDT_WTGR register).

### 26.2.10 Watchdog Counter Register Access Restriction (WDT_WCRR Register)

A 32-bit shadow register is implemented to read a coherent value of the WDT_WCRR register because the WDT_WCRR register is directly related to the timer counter value and is updated on the timer clock (WDT_FCLK). The shadow register is updated by a 16-bit LSB read command.

**NOTE:** Although the L4 clock (WDT_ICLK) is completely asynchronous with the timer clock (WDT_FCLK), some synchronization is performed to ensure that the value of the WDT_WCRR register is not read while it is being incremented.

When 32-bit read access is performed, the shadow register is not updated. Read access is performed directly from the accessed register.


Copyright © 2011–2016, Texas Instruments Incorporated

To ensure that a coherent value is read inside WDT_WCRR, the first read access is to the lower 16 bits (offset = 8h), followed by read access to the upper 16 bits (offset = Ah).

### 26.2.11 Watchdog Timer Interrupt Generation

When an interrupt source occurs, the interrupt status bit (the WDT_WIRQSTAT[0] EVENT_OVF or WDT_WIRQSTAT[1] EVENT_DLY bit) is set to 1. The output interrupt line (WDTi_IRQ) is asserted (active low) when status (the EVENT_xxx bit) and enable (the xxx_IT_ENA bit) flags are set to 1; the order is not relevant. Writing 1 to the enable bit (the status is already set at 1) also triggers the interrupt in the normal order (enable first, status next). The pending interrupt event is cleared when the set status bit is overwritten by a value of 1 by a write command in the WDT_WIRQSTAT register. Reading the WDT_WIRQSTAT register and writing the value back allows a fast interrupt acknowledge process.

The watchdog timer issues an overflow interrupt if this interrupt is enabled in the watchdog interrupt enable register (WDT_WIRQENSET[0] OVF_IT_ENA = 1). When the overflow occurs, the interrupt status bit (the WDT_WIRQSTAT[0] EVENT_OVF bit) is set to 1. The output interrupt line (WDT_IRQ) is asserted (active low) when status (EVENT_OVF) and enable (OVF_IT_ENA) flags are set to 1; the order is not relevant. This interrupt can be disabled by setting the WDT_WIRQENCLR[0] OVF_IT_ENA bit to 1.

The watchdog can issue the delay interrupt if this interrupt is enabled in the interrupt enable register (WDT_WIRQENSET[1] DLY_IT_ENA = 1). When the counter is running and the counter value matches the value stored in the delay configuration register (WDT_WDLY), the corresponding interrupt status bit is set in the watchdog status register (WDT_WIRQSTAT) and the output interrupt line is asserted (active low) when the flag (EVENT_DLY) and enable (DLY_IT_ENA) bits are 1 in the WDT_WIRQSTAT and WDT_WIRQENSET registers, respectively; the order (normally enable, then flag), is not relevant. This interrupt can be disabled by setting the WDT_WIRQENCLR[1] DLY_IT_ENA bit to 1.

**NOTE:** Writing 0 to the WDT_WIRQSTAT[0] EVENT_OVF bit or the WDT_WIRQSTAT[1] EVENT_DLY bit has no effect.

The two clock domains are resynchronized because the interrupt event is generated on the functional clock domain (WDTi_FCLK) during the updating of the interrupt status register (WDT_WIRQSTAT).

The WDT_WDLY register is used to specify the value of the delay configuration register. The delay time to interrupt is the difference between the reload value stored in the counter load register (WDT_WLDR) and the programmed value in this register (WDT_WDLY).

Use the following formula to estimate the delay time:

*Delay time period = (WDT_WDLY – WDT_WLDR + 1) × Timer clock period × Clock divider*

Where:

- Timer clock period = 1/(Timer clock frequency)
- Clock divider = 2PTV

If the counter value (WDT_WCRR) reaches the programmed value (WDT_WDLY), the status bit (EVENT_DLY) gets set in the interrupt status register (WDT_WIRQSTAT), and an interrupt occurs if the corresponding enable bit is set in the interrupt enable register (WDT_WIRQENSET).

**CAUTION**

If the reload event occurs (after a triggering sequence or after a reset sequence) before reaching the programmed value (WDT_WDLY[31:0] WDLY_VALUE), no interrupt is generated.

Also, no interrupt is generated if the value programmed in the delay configuration register (WDT_WDLY) is less than the value stored in the counter load register (WDT_WLDR).


Copyright © 2011–2016, Texas Instruments Incorporated

### 26.2.12 Watchdog Timers Under Emulation

During emulation mode, the watchdog timer can/cannot continue running, according to the value of the WDT_WDSC[5] EMUFREE bit of the system configuration register (WDT_WDSC).

- When EMUFREE is 1, watchdog timer execution is not stopped and a reset pulse is still generated when overflow is reached.
- When EMUFREE is 0, the counters (prescaler/timer) are frozen and incrementation restarts after exiting from emulation mode.

### 26.2.13 Accessing Watchdog Timer Registers

Posted/nonposted selection applies only to functional registers that require synchronization on/from the timer functional clock domain (WDTi_FCLK). For write/read operation, the following registers are affected:

- WDT_WCLR
- WDT_WCRR
- WDT_WLDR
- WDT_WTGR
- WDT_WDLY
- WDT_WSPR

The timer interface clock domain synchronous registers are not affected by the posted/nonposted selection; the write/read operation is effective and acknowledged (command accepted) after one WDT_ICLK cycle from the command assertion. The timer interface clock domain synchronous registers are:

- WDT_WIDR
- WDT_WDSC
- WDT_WDST
- WDT_WIRQSTATRAW
- WDT_WIRQSTAT
- WDT_WIRQENSET
- WDT_WIRQENCLR
- WDT_WWPS

**NOTE:** Accesses to WDT2 and WDT3 are posted.


Copyright © 2011–2016, Texas Instruments Incorporated

## 26.3 Low-Level Programming Model

This section covers the low-level hardware programming sequences for configuration and use of the module.

### *26.3.1 Global Initialization*

#### 26.3.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the watchdog timer is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the watchdog timer (see Table 26-6).

**Table 26-6. Global Initialization of Surrounding Modules**

| Surrounding Modules | Comments |
|---|---|
| PRCM | The module interface and functional clocks must be enabled. |
| Control module | Module-specific pad multiplexing must be set in the control module. |
| GP Device | Enable WDT in the GP device. See the Control Module Register: CONTROL_SEC_CTRL in the *Control Module* chapter. |
| MPU INTC | The MPU INTC configuration must be performed to enable the interrupts from the watchdog timer. |

#### 26.3.1.2 Main Sequence – Watchdog Timer Module Global Initialization

Table 26-7 lists the steps for initializing the watchdog timer module when the module is to be used for the first time.

**Table 26-7. Watchdog Timer Module Global Initialization**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Execute software reset. | WDT_WDSC[1] SOFTRESET | 1 |
| Wait until reset release? | WDT_WDSC[1] SOFTRESET | 0 |
| Enable delay interrupt. | WDT_WIRQENSET[1] ENABLE_DLY | 1 |
| Enable overflow interrupt. | WDT_WIRQENSET[0] ENABLE_OVF | 1 |

### *26.3.2 Operational Mode Configuration*

#### 26.3.2.1 Main Sequence – Watchdog Timer Basic Configuration

Table 26-8 lists the steps for the basic configuration of the watchdog timer.

**Table 26-8. Watchdog Timer Basic Configuration**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Disable the watchdog timer. | See Section 26.3.2.2. | |
| Set prescaler value. | WDT_WCLR[4:2] PTV | xxx |
| Enable prescaler. | WDT_WCLR[5] PRE | 1 |
| Load delay configuration value. | WDT_WDLY | xxx |
| Load timer counter value. | WDT_WCRR | xxx |
| Enable the watchdog timer. | See Section 26.3.2.3. | |


Copyright © 2011–2016, Texas Instruments Incorporated

#### 26.3.2.2 Subsequence – Disable the Watchdog Timer

Table 26-9 lists the steps to disable the watchdog timer.

**Table 26-9. Disable the Watchdog Timer**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Write disable sequence Data1. | WDT_WSPR | XXXX AAAAh |
| Write disable sequence Data2. | WDT_WSPR | XXXX 5555h |

#### 26.3.2.3 Subsequence – Enable the Watchdog Timer

Table 26-10 lists the steps to enable the watchdog timer.

**Table 26-10. Enable the Watchdog Timer**

| Step | Register/Bit Field/Programming Model | Value |
|---|---|---|
| Write enable sequence Data1. | WDT_WSPR | XXXX BBBBh |
| Write enable sequence Data2. | WDT_WSPR | XXXX 4444h |


Copyright © 2011–2016, Texas Instruments Incorporated

## 26.4 Watchdog Timer Registers

The watchdog timer registers are listed in Table 26-11.

**CAUTION**

The watchdog timers registers are limited to 32-bit and 16-bit data accesses; 8-bit access is not allowed and can corrupt register content.

**NOTE:**

- The WDT_WISR and WDT_WIRQSTATRAW registers have the same functionality. The WDT_WISR register is used for software backward compatibility.
- The WDT_WIER and WDT_WIRQENSET/WDT_WIRQENCLR registers have the same functionality. The WDT_WIER register is used for software backward compatibility.
- The WDT_WIRQSTATRAW and WDT_WIRQSTAT registers give the same information when read. The WDT_WIRQSTATRAW register is used for debug.

**Table 26-11. Watchdog Timer Registers**

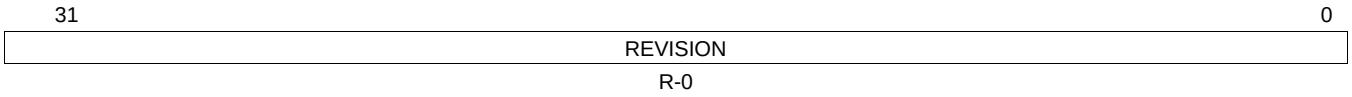
| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 0h | WDT_WIDR | WDT_WIDR | Section 26.4.1 |
| 10h | WDT_WDSC | WDT_WDSC | Section 26.4.2 |
| 14h | WDT_WDST | WDT_WDST | Section 26.4.3 |
| 18h | WDT_WISR | WDT_WISR | Section 26.4.4 |
| 1Ch | WDT_WIER | WDT_WIER | Section 26.4.5 |
| 24h | WDT_WCLR | WDT_WCLR | Section 26.4.6 |
| 28h | WDT_WCRR | WDT_WCRR | Section 26.4.7 |
| 2Ch | WDT_WLDR | WDT_WLDR | Section 26.4.8 |
| 30h | WDT_WTGR | WDT_WTGR | Section 26.4.9 |
| 34h | WDT_WWPS | WDT_WWPS | Section 26.4.10 |
| 44h | WDT_WDLY | WDT_WDLY | Section 26.4.11 |
| 48h | WDT_WSPR | WDT_WSPR | Section 26.4.12 |
| 54h | WDT_WIRQSTATRAW | WDT_WIRQSTATRAW | Section 26.4.13 |
| 58h | WDT_WIRQSTAT | WDT_WIRQSTAT | Section 26.4.14 |
| 5Ch | WDT_WIRQENSET | WDT_WIRQENSET | Section 26.4.15 |
| 60h | WDT_WIRQENCLR | WDT_WIRQENCLR | Section 26.4.16 |


Copyright © 2011–2016, Texas Instruments Incorporated

### 26.4.1 WDT_WIDR Register

The WDT_WIDR register is shown and described in the figure and table below.

**Figure 26-4. WDT_WIDR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-12. WDT_WIDR Register Field Descriptions**

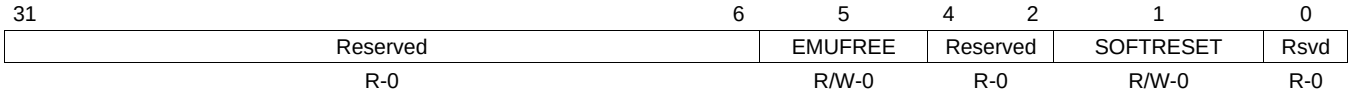
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | REVISION | R | 0 | IP Revision |

### 26.4.2 WDT_WDSC Register

The WDT_WDSC register is shown and described in the figure and table below.

This register controls the various parameters of the L4 interface.

**Figure 26-5. WDT_WDSC Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-13. WDT_WDSC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0 | Write 0s for future compatibility. Reads return 0. |
| 5 | EMUFREE | R/W | 0 | Emulation mode<br>0 = Timer counter frozen in emulation<br>1 = Timer counter free-running in emulation |
| 4-2 | Reserved | R | 0 | Write 0s for future compatibility. Reads return 0. |
| 1 | SOFTRESET | R/W | 0 | Software reset. (Optional)<br>Read 0 = Reset done, no pending action<br>Write 0 = No action<br>Write 1 = Initiate software reset.<br>Read 1 = Reset (software or other) ongoing |
| 0 | Reserved | R | 0 | Write 0s for future compatibility. Reads return 0. |


Copyright © 2011–2016, Texas Instruments Incorporated

### 26.4.3 WDT_WDST Register

The WDT_WDST register provides status information about the module.

It is shown and described in the figure and table below.

**Figure 26-6. WDT_WDST Register**

| 31 | 1 | 0 |
|---|---|---|
| Reserved | | RESETDONE |
| R-0 | | R-1 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-14. WDT_WDST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | Reserved | R | 0 | Reads return 0. |
| 0 | RESETDONE | R | 1 | Internal module reset monitoring<br>Read 0 = Internal module reset is ongoing.<br>Read 1 = Reset completed |

### 26.4.4 WDT_WISR Register

This register shows which interrupt events are pending inside the module.

It is shown and described in the figure and table below.

**Figure 26-7. WDT_WISR Register**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | DLY_IT_FLAG | OVF_IT_FLAG |
| R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-15. WDT_WISR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0 | Reads return 0. |
| 1 | DLY_IT_FLAG | R/W | 0 | Pending delay interrupt status.<br>Read 0 = No delay interrupt pending<br>Write 0 = Status unchanged<br>Write 1 = Status bit cleared<br>Read 1 = Delay interrupt pending |
| 0 | OVF_IT_FLAG | R/W | 0 | Pending overflow interrupt status.<br>Read 0 = No overflow interrupt pending<br>Write 0 = Status unchanged<br>Write 1 = Status bit cleared<br>Read 1 = Overflow interrupt pending |


Copyright © 2011–2016, Texas Instruments Incorporated

### 26.4.5 WDT_WIER Register

The WDT_WIERs register controls (enable/disable) the interrupt events.

It is shown and described in the figure and table below.

**Figure 26-8. WDT_WIER Register**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | DLY_IT_ENA | OVF_IT_ENA |
| R-0 | | R/W-0 | R/W-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-16. WDT_WIER Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0 | Reads return 0. |
| 1 | DLY_IT_ENA | R/W | 0 | Delay interrupt enable/disable<br>0 = Disable delay interrupt.<br>1 = Enable delay interrupt. |
| 0 | OVF_IT_ENA | R/W | 0 | Overflow interrupt enable/disable<br>0 = Disable overflow interrupt.<br>1 = Enable overflow interrupt. |

### 26.4.6 WDT_WCLR Register

The WDT_WCLR register controls the prescaler stage of the counter.

It is shown and descirbed in the figure and table below.

**Figure 26-9. WDT_WCLR Register**

| 31 | 6 | 5 | 4 | 2 | 0 |
|---|---|---|---|---|---|
| Reserved | | PRE | PTV | | Rsvd |
| R-0 | | R/W-1 | R/W-0 | | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-17. WDT_WCLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0 | Reads return 0. |
| 5 | PRE | R/W | 1 | Prescaler enable/disable configuration<br>0 = Prescaler disabled<br>1 = Prescaler enabled |
| 4-2 | PTV | R/W | 0 | Prescaler value<br>The timer counter is prescaled with the value: 2**PTV.<br>Example: PTV = 3 then counter increases value if started after 8 functional clock periods.<br>On reset, it is loaded from PI_PTV_RESET_VALUE input port. |
| 1-0 | Reserved | R | 0 | Write 0s for future compatibility. Reads return 0. |

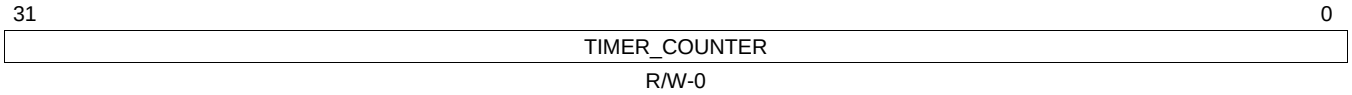

Copyright © 2011–2016, Texas Instruments Incorporated

### 26.4.7 WDT_WCRR Register

The WDT_WCRR register holds the value of the internal counter.

It is shown and described in the figure and table below.

**Figure 26-10. WDT_WCRR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-18. WDT_WCRR Register Field Descriptions**

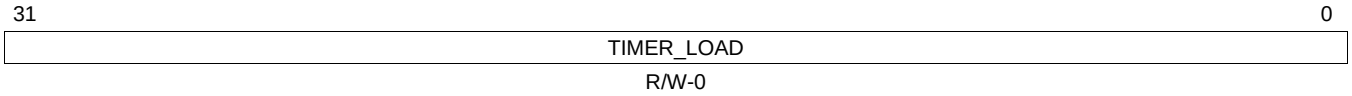
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | TIMER_COUNTER | R/W | 0 | Value of the timer counter register |

### 26.4.8 WDT_WLDR Register

The WDT_WLDR register holds the timer load value.

It is shown and described in the figure and table below.

**Figure 26-11. WDT_WLDR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

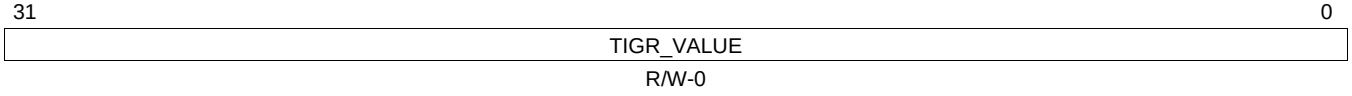
**Table 26-19. WDT_WLDR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | TIMER_LOAD | R/W | 0 | Value of the timer load register |

### 26.4.9 WDT_WTGR Register

Writing a different value than the one already written in this register does a watchdog counter reload.

**Figure 26-12. WDT_WTGR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset


Copyright © 2011–2016, Texas Instruments Incorporated

**Table 26-20. WDT_WTGR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | TTGR_VALUE | R/W | 0 | Value of the trigger register |

### 26.4.10 WDT_WWPS Register

This register contains the write posting bits for all writeable functional registers.

**Figure 26-13. WDT_WWPS Register**

| 31 | | | | | | | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| R-0 | | | | | | | |
| 15 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | W_PEND_WDLY | W_PEND_WSPR | W_PEND_WTGR | W_PEND_WLDR | W_PEND_WCRR | W_PEND_WCLR |
| R-0 | | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-21. WDT_WWPS Register Field Descriptions**

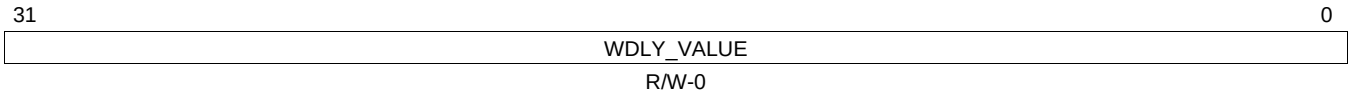
| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | Reserved | R | 0 | Write 0s for future compatibility. Reads return 0. |
| 5 | W_PEND_WDLY | R | 0 | Write pending for register WDLY<br>Read 0 = No register write pending<br>Read 1 = Register write pending |
| 4 | W_PEND_WSPR | R | 0 | Write pending for register WSPR<br>Read 0 = No register write pending<br>Read 1 = Register write pending |
| 3 | W_PEND_WTGR | R | 0 | Write pending for register WTGR<br>Read 0 = No register write pending<br>Read 1 = Register write pending |
| 2 | W_PEND_WLDR | R | 0 | Write pending for register WLDR<br>Read 0 = No register write pending<br>Read 1 = Register write pending |
| 1 | W_PEND_WCRR | R | 0 | Write pending for register WCRR<br>Read 0 = No register write pending<br>Read 1 = Register write pending |
| 0 | W_PEND_WCLR | R | 0 | Write pending for register WCLR<br>Read 0 = No register write pending<br>Read 1 = Register write pending |


Copyright © 2011–2016, Texas Instruments Incorporated

### 26.4.11 WDT_WDLY Register

ThE WDT_WDLY register holds the delay value that controls the internal pre-overflow event detection.

**Figure 26-14. WDT_WDLY**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

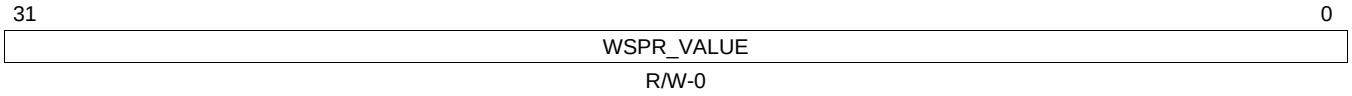
**Table 26-22. WDT_WDLY Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | WDLY_VALUE | R/W | 0 | Value of the delay register |

### 26.4.12 WDT_WSPR Register

The WDT_WSPR register holds the start-stop value that controls the internal start-stop FSM.

**Figure 26-15. WDT_WSPR Register**




LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-23. WDT_WSPR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | WSPR_VALUE | R/W | 0 | Value of the start-stop register |


Copyright © 2011–2016, Texas Instruments Incorporated

### 26.4.13 WDT_WIRQSTATRAW Register

IRQ unmasked status, status set per-event raw interrupt status vector, line 0. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.

**Figure 26-16. WDT_WIRQSTATRAW Register**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | EVENT_DLY | EVENT_OVF |
| R-0 | | R/W1S-0 | R/W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-24. WDT_WIRQSTATRAW Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0 | Write 0s for future compatibility. Reads return 0. |
| 1 | EVENT_DLY | R/W1S | 0 | Settable raw status for delay event<br>Read 0 = No event pending<br>Write 0 = No action<br>Write 1 = Set event (debug)<br>Read 1 = Event pending |
| 0 | EVENT_OVF | R/W1S | 0 | Settable raw status for overflow event<br>Read 0 = No event pending<br>Write 0 = No action<br>Write 1 = Set event (debug)<br>Read 1 = Event pending |


Copyright © 2011–2016, Texas Instruments Incorporated

### 26.4.14 WDT_WIRQSTAT Register

IRQ masked status, status clear per-event enabled interrupt status vector, line 0. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).

**Figure 26-17. WDT_WIRQSTAT Register**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | EVENT_DLY | EVENT_OVF |
| R-0 | | R/W1C-0 | R/W1C-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-25. WDT_WIRQSTAT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0 | Write 0s for future compatibility. Reads return 0. |
| 1 | EVENT_DLY | R/W1C | 0 | Clearable, enabled status for delay event<br>Read 0 = No (enabled) event pending<br>Write 0 = No action<br>Write 1 = Clear (raw) event<br>Read 1 = Event pending |
| 0 | EVENT_OVF | R/W1C | 0 | Clearable, enabled status for overflow event<br>Read 0 = No (enabled) event pending<br>Write 0 = No action<br>Write 1 = Clear (raw) event<br>Read 1 = Event pending |


Copyright © 2011–2016, Texas Instruments Incorporated

### 26.4.15 WDT_WIRQENSET Register

IRQ enable set per-event interrupt enable bit vector, line 0. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.

**Figure 26-18. WDT_WIRQENSET Register**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | ENABLE_DLY | ENABLE_OVF |
| R-0 | | R/W1S-0 | R/W1S-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-26. WDT_WIRQENSET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0 | Write 0s for future compatibility. Reads return 0. |
| 1 | ENABLE_DLY | R/W1S | 0 | Enable for delay event<br>Read 0 = Interrupt disabled (masked)<br>Write 0 = No action<br>Write 1 = Enable interrupt.<br>Read 1 = Interrupt enabled |
| 0 | ENABLE_OVF | R/W1S | 0 | Enable for overflow event<br>Read 0 = Interrupt disabled (masked)<br>Write 0 = No action<br>Write 1 = Enable interrupt.<br>Read 1 = Interrupt enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

### 26.4.16 WDT_WIRQENCLR Register

IRQ enable clear per-event interrupt enable bit vector, line 0. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.

**Figure 26-19. WDT_WIRQENCLR Register**

| 31 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | | ENABLE_DLY | ENABLE_OVF |
| R-0 | | R/W1C-0 | R/W1C-0 |

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

**Table 26-27. WDT_WIRQENCLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | Reserved | R | 0 | Write 0s for future compatibility. Reads return 0. |
| 1 | ENABLE_DLY | R/W1C | 0 | Enable for delay event<br>Read 0 = Interrupt disabled (masked)<br>Write 0 = No action<br>Write 1 = Disable interrupt.<br>Read 1 = Interrupt enabled |
| 0 | ENABLE_OVF | R/W1C | 0 | Enable for overflow event<br>Read 0 = Interrupt disabled (masked)<br>Write 0 = No action<br>Write 1 = Disable interrupt.<br>Read 1 = Interrupt enabled |


Copyright © 2011–2016, Texas Instruments Incorporated

# Revision History

| Changes from July 4, 2015 to June 15, 2016 (from F Revision (July 2015) to G Revision) | Page |
|---|---|
| • **Chapter 7: DDR2 and DDR3 Memory Controller** | 1226 |
| • Figure 7-44: Changed Reserved bits 29-21 Reset value to 10h | 1298 |
| • Figure 7-45: Changed Reserved bits 29-21 Reset value to 10h | 1299 |
| • **Chapter 9: EMAC** | 1450 |
| • Section 9.1.5.2.1: Updated third bullet | 1454 |
| • **Chapter 12: High-Definition Video Processing Subsystem (HDVPSS)** | 1776 |
| • Updated entire chapter | 1776 |


Copyright © 2011–2016, Texas Instruments Incorporated

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

| **Products** | | **Applications** | |
|---|---|---|---|
| Audio | www.ti.com/audio | Automotive and Transportation | www.ti.com/automotive |
| Amplifiers | amplifier.ti.com | Communications and Telecom | www.ti.com/communications |
| Data Converters | dataconverter.ti.com | Computers and Peripherals | www.ti.com/computers |
| DLP® Products | www.dlp.com | Consumer Electronics | www.ti.com/consumer-apps |
| DSP | dsp.ti.com | Energy and Lighting | www.ti.com/energy |
| Clocks and Timers | www.ti.com/clocks | Industrial | www.ti.com/industrial |
| Interface | interface.ti.com | Medical | www.ti.com/medical |
| Logic | logic.ti.com | Security | www.ti.com/security |
| Power Mgmt | power.ti.com | Space, Avionics and Defense | www.ti.com/space-avionics-defense |
| Microcontrollers | microcontroller.ti.com | Video and Imaging | www.ti.com/video |
| RFID | www.ti-rfid.com | | |
| OMAP Applications Processors | www.ti.com/omap | **TI E2E Community** | e2e.ti.com |
| Wireless Connectivity | www.ti.com/wirelessconnectivity | | |

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2016, Texas Instruments Incorporated